

Contents

Networking

[Windows Server supported networking scenarios](#)

[What's new in networking](#)

[Core network guidance for Windows Server](#)

[Core network components](#)

[Core network companion guidance](#)

[Deploy server certificates for 802.1X wired and wireless deployments](#)

[Server certificate deployment overview](#)

[Server certificate deployment planning](#)

[Server certificate deployment](#)

[Install the Web Server WEB1](#)

[Create an alias \(CNAME\) record in DNS for WEB1](#)

[Configure WEB1 to distribute certificate revocation lists \(CRLs\)](#)

[Prepare the CAPolicy.inf File](#)

[Install the Certification Authority](#)

[Configure the CDP and AIA Extensions on CA1](#)

[Copy the CA certificate and CRL to the virtual directory](#)

[Configure the server certificate template](#)

[Configure server certificate autoenrollment](#)

[Refresh group policy](#)

[Verify server enrollment of a server certificate](#)

[Deploy password-based 802.1X authenticated wireless access](#)

[Wireless access deployment overview](#)

[Wireless access deployment process](#)

[Wireless access deployment planning](#)

[Wireless access deployment](#)

[Deploy BranchCache hosted cache mode](#)

[BranchCache hosted cache mode deployment overview](#)

[BranchCache hosted cache mode deployment planning](#)

BranchCache hosted cache mode deployment

Install the BranchCache feature and configure the hosted cache server by Service Connection Point

Move and resize the hosted cache (Optional)

Prehash and preload content on the hosted cache server (Optional)

Configure client automatic hosted cache discovery by Service Connection Point

Additional resources

BranchCache

[BranchCache netsh and Windows PowerShell commands](#)

[BranchCache deployment guidance](#)

[Choosing a BranchCache design](#)

[Deploy BranchCache](#)

[Install and configure content servers](#)

[Install content servers that use the BranchCache feature](#)

[Install File Services content servers](#)

[Deploy hosted cache servers \(Optional\)](#)

[Prehashing and preloading content on hosted cache servers \(Optional\)](#)

[Configure BranchCache client computers](#)

[Use group policy to configure domain member client computers](#)

[Use Windows PowerShell to configure non-domain member client computers](#)

[Verify client computer settings](#)

DirectAccess

Domain Name System (DNS)

[What's new in DNS client in Windows Server](#)

[What's new in DNS server in Windows Server](#)

[DNS policy scenario guidance](#)

[DNS policies overview](#)

[Use DNS policy for geo-location traffic management with primary servers](#)

[Use DNS policy for geo-location traffic management with primary-secondary deployments](#)

[Use DNS policy for intelligent DNS responses based on time of day](#)

[DNS responses based on time of day with an Azure cloud app server](#)

[Use DNS policy for Split-Brain DNS deployment](#)

- Use DNS policy for Split-Brain DNS in Active Directory
 - Use DNS policy for applying filters on DNS queries
 - Use DNS policy for app load balancing
 - Use DNS policy for app load balancing with geo-location awareness
- Dynamic Host Configuration Protocol (DHCP)
 - What's new in DHCP
 - DHCP subnet selection options
 - DHCP logging events for DNS record registrations
 - Deploy DHCP using Windows PowerShell
- High-Performance Networking (HPN)
 - Network offload and optimization technologies
 - Software only (SO) features and technologies
 - Software and hardware (SH) integrated features and technologies
 - Hardware Only (HO) features and technologies
 - NIC advanced properties
- Insider preview
 - Receive Segment Coalescing (RSC) in the vSwitch
 - Converged NIC configuration guidance
 - Single network adapter configuration
 - Datacenter network adapter configuration
 - Physical switch configuration
 - Troubleshooting Converged NIC
 - Data Center Bridging (DCB)
 - Install DCB
 - Manage DCB
 - Virtual Receive Side Scaling (vRSS)
 - Plan the use of vRSS
 - Enable vRSS on a virtual network adapter
 - Manage vRSS
 - vRSS FAQ
 - Windows PowerShell commands for RSS and vRSS
 - Resolve vRSS issues

[Hyper-V Virtual Switch](#)

[IP Address Management \(IPAM\)](#)

[What's new in IPAM](#)

[Manage IPAM](#)

[DNS resource record management](#)

[Add a DNS resource record](#)

[Delete DNS resource records](#)

[Filter the view of DNS resource records](#)

[View DNS resource records for a specific IP address](#)

[DNS zone management](#)

[Create a DNS zone](#)

[Edit a DNS zone](#)

[View DNS resource records for a DNS zone](#)

[View DNS zones](#)

[Manage resources in multiple active directory forests](#)

[Purge utilization data](#)

[Role-based access control](#)

[Manage role-based access control with Server Manager](#)

[Create a user role for access control](#)

[Create an access policy](#)

[Set access scope for a DNS zone](#)

[Set access scope for DNS resource records](#)

[View roles and role permissions](#)

[Manage role-based access control with Windows PowerShell](#)

[Network Load Balancing](#)

[Network Policy Server \(NPS\)](#)

[NPS best practices](#)

[Getting started with NPS](#)

[Connection request processing](#)

[Connection request policies](#)

[Realm names](#)

[Remote RADIUS server groups](#)

[Network policies](#)

[Access permission](#)

[NPS templates](#)

[RADIUS clients](#)

[Plan NPS](#)

[Plan NPS as a RADIUS server](#)

[Plan NPS as a RADIUS proxy](#)

[Deploy NPS](#)

[Manage NPS](#)

[Network Policy Server Management with Administration Tools](#)

[Configure connection request policies](#)

[Configure firewalls for RADIUS traffic](#)

[Configure network policies](#)

[Configure NPS Accounting](#)

[Configure RADIUS clients](#)

[Configure remote RADIUS server groups](#)

[Manage certificates used with NPS](#)

[Configure certificate templates for PEAP and EAP requirements](#)

[Manage NPSs](#)

[Configure NPS on a multihomed computer](#)

[Configure NPS UDP port information](#)

[Disable NAS notification forwarding](#)

[Export an NPS configuration for import on another server](#)

[Increase concurrent authentications processed by NPS](#)

[Install NPS](#)

[NPS proxy server load balancing](#)

[Register an NPS in an Active Directory Domain](#)

[Unregister an NPS from an Active Directory Domain](#)

[Use regular expressions in NPS](#)

[Verify configuration after NPS changes](#)

[NPS data collection](#)

[Manage NPS templates](#)

Network Shell (Netsh)

Netsh command syntax, contexts, and formatting

Network Shell (Netsh) example batch file

Netsh http commands

Netsh interface portproxy commands

Network subsystem performance tuning

Choosing a network adapter

Configure the order of network interfaces

Performance tuning network adapters

Network-related performance counters

Performance tools for network workloads

NIC Teaming

NIC Teaming MAC address use and management

Create a New NIC Team on a host computer or VM

Troubleshooting NIC Teaming

Quality of Service (QoS) policy

Getting started with QoS policy

How QoS policy works

QoS policy architecture

QoS policy scenarios

Manage QoS policy

QoS policy events and errors

QoS policy FAQ

Software Defined Networking (SDN)

SDN in Windows Server overview

SDN technologies

Hyper-V network virtualization

Hyper-V network virtualization overview

Hyper-V network virtualization technical details

What's new in Hyper-V Network virtualization

Internal DNS service (iDNS) for SDN

Network Controller

[Network Controller high availability](#)

[Install the Network Controller server role using Server Manager](#)

[Post-deployment steps for Network Controller](#)

[Network function virtualization](#)

[Datacenter firewall overview](#)

[RAS Gateway for SDN](#)

[What's new in RAS Gateway](#)

[RAS Gateway deployment architecture](#)

[RAS Gateway high availability](#)

[Software Load Balancing \(SLB\) for SDN](#)

[Switch Embedded Teaming \(SET\) for SDN](#)

[Container networking](#)

[Plan an SDN Infrastructure](#)

[Installation and preparation requirements for deploying Network Controller](#)

[Deploy SDN](#)

[Deploy an SDN Infrastructure](#)

[Deploy an SDN infrastructure using scripts](#)

[Deploy SDN technologies using Windows PowerShell](#)

[Deploy Network Controller using Windows PowerShell](#)

[Manage SDN](#)

[Manage tenant virtual networks](#)

[Understanding usage of virtual networks and VLANs](#)

[Use Access Control lists \(ACLs\) to manage datacenter network traffic flow](#)

[Create, delete, or update tenant virtual networks](#)

[Add a virtual gateway to a tenant virtual network](#)

[Connect container endpoints to a tenant virtual network](#)

[Configure encryption for a virtual subnet](#)

[Manage tenant workloads](#)

[Create a VM and connect to a tenant virtual network or VLAN](#)

[Configure QoS for a tenant VM network adapter](#)

[Configure datacenter firewall ACLs](#)

[Configure the Software Load Balancer for load balancing and Network address Translation \(NAT\)](#)

- [Use network virtual appliances on a virtual network](#)
- [Guest clustering in a virtual network](#)
- [Update, backup, and restore an SDN infrastructure](#)
- [Security for SDN](#)
 - [Secure the Network Controller](#)
 - [Manage certificates for SDN](#)
 - [Kerberos with Service Principal Name \(SPN\)](#)
 - [SDN firewall auditing](#)
- [Virtual network peering](#)
 - [Configure virtual network peering](#)
- [Egress metering in virtual network](#)
- [Windows Server 2019 gateway performance](#)
- [Gateway bandwidth allocation](#)
- [Troubleshoot SDN](#)
 - [Troubleshoot the Windows Server Software Defined Networking Stack](#)
- [System Center Technologies for SDN](#)
- [Microsoft Azure and SDN](#)
- [Contact the Datacenter and Cloud Networking Team](#)
- [Virtual Private Networking \(VPN\)](#)
- [Windows Internet Name Service \(WINS\)](#)
- [Windows Time service](#)
 - [Insider preview - Windows Time service in Windows Server 2019](#)
 - [Accurate time for Windows Server 2016](#)
 - [Support boundary to configure the Windows Time service for high-accuracy environments](#)
 - [Configuring systems for high accuracy](#)
 - [Windows Time for traceability](#)
 - [Windows Time service technical reference](#)
 - [How the Windows Time service works](#)
 - [Windows Time service tools and settings](#)

Networking

9/21/2018 • 8 minutes to read • [Edit Online](#)

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

TIP

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



Networking is a foundational part of the Software Defined Datacenter (SDDC) platform, and Windows Server 2016 provides new and improved Software Defined Networking (SDN) technologies to help you move to a fully realized SDDC solution for your organization.

When you manage networks as a software defined resource, you can describe an application's infrastructure requirements one time, and then choose where the application runs - on premises or in the cloud.

This consistency means that your applications are now easier to scale, and you can seamlessly run applications - anywhere - with equal confidence about security, performance, quality of service, and availability.

NOTE

To download Windows Server, see [Windows Server Evaluations](#).

Windows Server 2016 adds the following new networking technologies:

- **Software Defined Networking:** Network Controller provides a centralized, programmable point of automation to manage, configure, monitor, and troubleshoot virtual and physical network infrastructure in your datacenter. Network Controller allows you to use Network Function Virtualization to easily deploy virtual machines (VMs) for Software Load Balancing (SLB) to optimize network traffic loads for your tenants, and RAS Gateways to provide tenants with the connectivity options they need between Internet, on-prem, and cloud resources. You can also use Network Controller to manage Datacenter Firewall on VMs and Hyper-V hosts.
- **Network Platform:** Using new features for existing Network Platform technologies, you can use DNS Policy to customize your DNS server responses to queries, use a converged NIC that handles combined Remote Direct Memory Access (RDMA) and Ethernet traffic, use Switch Embedded Teaming (SET) to create Hyper-V Virtual Switches connected to RDMA NICs, and use IP Address Management (IPAM) to manage DNS zones and servers as well as DHCP and IP addresses.

For more information, see [Windows Server Supported Networking Scenarios](#).

The following sections provide information about SDN technologies and Network Platform technologies.

Software Defined Networking technologies

Software Defined Networking (SDN)

You can use this topic to learn about the SDN technologies that are provided in Windows Server, System Center, and Microsoft Azure.

NOTE

For Hyper-V hosts and virtual machines (VMs) that run SDN infrastructure servers, such as Network Controller and Software Load Balancing nodes, you must install Windows Server 2016 Datacenter edition. For Hyper-V hosts that contain only tenant workload VMs that are connected to SDN-controlled networks, you can run Windows Server 2016 Standard edition.

Deploy a Software Defined Network infrastructure using scripts

This guide provides instructions on how to deploy Network Controller with virtual networks and gateways in a test lab environment.

Network Controller

Network Controller provides a centralized, programmable point of automation to manage, configure, monitor, and troubleshoot virtual and physical network infrastructure in your datacenter.

Software Load Balancing (SLB) for SDN

Cloud Service Providers (CSPs) and Enterprises that are deploying Software Defined Networking (SDN) in Windows Server 2016 can use Software Load Balancing (SLB) to evenly distribute tenant and tenant customer network traffic among virtual network resources. The Windows Server SLB enables multiple servers to host the same workload, providing high availability and scalability.

RAS Gateway for SDN

RAS Gateway, which is a software-based, multitenant, Border Gateway Protocol (BGP) capable router in Windows Server 2016, is designed for Cloud Service Providers (CSPs) and Enterprises that host multiple tenant virtual networks using Hyper-V Network Virtualization.

Network Function Virtualization

In software defined datacenters, network functions that are being performed by hardware appliances (such as load balancers, firewalls, routers, switches, and so on) are increasingly being virtualized as virtual appliances. This "network function virtualization" is a natural progression of server virtualization and network virtualization.

Datacenter Firewall Overview

Datacenter Firewall is a network layer, 5-tuple (protocol, source and destination port numbers, source and destination IP addresses), stateful, multitenant firewall.

Networking Technologies

The following table provides links to some of the networking technologies in Windows Server 2016.

What's New in Networking

You can use the following sections to discover new networking technologies and new features for existing technologies in Windows Server 2016.

BranchCache

BranchCache is a wide area network (WAN) bandwidth optimization technology. To optimize WAN bandwidth when users access content on remote servers, BranchCache fetches content from your main office or hosted cloud content servers and caches the content at branch office locations, allowing client computers at branch offices to access the content locally rather than over the WAN.

Core Network Guide for Windows Server 2016

Learn how to deploy a Windows Server network with the Core Network Guide, as well as add features to your network deployment with Core Network Companion Guides.

DirectAccess

DirectAccess allows connectivity for remote users to organization network resources.

DirectAccess documentation is now located in the [Remote access and server management](#) section of the Windows Server 2016 table of contents, under [Remote Access](#). For more information, see [DirectAccess](#).

Domain Name System (DNS)

Domain Name System (DNS) is one of the industry-standard suite of protocols that comprise TCP/IP, and together the DNS Client and DNS Server provide computer name-to-IP address mapping name resolution services to computers and users.

Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information, such as the subnet mask and default gateway.

Hyper-V Network Virtualization

Hyper-V Network Virtualization (HNV) enables virtualization of customer networks on top of a shared physical network infrastructure.

Hyper-V Virtual Switch

The 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. The switch includes programmatically managed and extensible capabilities to connect virtual machines to both virtual networks and the physical network. In addition, Hyper-V Virtual Switch provides policy enforcement for security, isolation, and service levels.

Hyper-V Virtual Switch documentation is now located in the **Virtualization** section of the Windows Server 2016 table of contents. For more information, see [Hyper-V Virtual Switch](#).

IP Address Management (IPAM)

IP Address Management (IPAM) is an integrated suite of tools to enable end-to-end planning, deploying, managing and monitoring of your IP address infrastructure, with a rich user experience. IPAM automatically discovers IP address infrastructure servers and Domain Name System (DNS) servers on your network and enables you to manage them from a central interface.

Network Load Balancing

Network Load Balancing (NLB) distributes traffic across several servers using the TCP/IP networking protocol. For non-SDN deployments, NLB ensures that stateless applications, such as Web servers running Internet Information Services (IIS), are scalable by adding more servers as the load increases.

High-Performance Networking

Network offload and optimization technologies in Windows Server 2016 include Software Only (SO) features and technologies, Software and Hardware (SH) integrated features and technologies, and Hardware Only (HO) features and technologies.

The following offload and optimization technology documentation is also available.

- [Converged Network Interface Card \(NIC\) Configuration Guide](#)
- [Data Center Bridging \(DCB\)](#)
- [Virtual Receive Side Scaling \(vRSS\)](#)

Network Policy Server

Network Policy Server (NPS) allows you to create and enforce organization-wide network access policies for connection request authentication and authorization.

Network Shell (Netsh)

You can use the Network Shell (netsh) networking utility to manage networking technologies in Windows Server 2016 and Windows 10.

Network Subsystem Performance Tuning

This topic provides information about choosing the right network adapter for your server workload, ordering network interfaces, network related performance counters, and performance tuning network adapters and related networking technologies, such as Receive Side Scaling (RSS), Receive Side Coalescing (RSC), and others.

NIC Teaming

NIC Teaming allows you to group physical Ethernet network adapters into one or more software-based virtual network adapters. These virtual network adapters provide fast performance and fault tolerance in the event of a network adapter failure.

Quality of Service (QoS) Policy

You can use QoS Policy as a central point of network bandwidth management across your entire Active Directory infrastructure by creating QoS profiles, whose settings are distributed with Group Policy.

Remote Access

You can use Remote Access technologies, such as DirectAccess and Virtual Private Networking (VPN) to provide remote workers with connectivity to internal network resources. In addition, you can use Remote Access for local area network (LAN) routing, and for Web Application Proxy, which provides reverse proxy functionality for web applications inside your corporate network to allow users on any device to access them from outside the corporate network.

Remote Access documentation is now located in the [Remote access and server management](#) section of the Windows Server 2016 table of contents. For more information, see [Remote Access](#).

For more information about Web Application Proxy, which is a role service of the Remote Access server role, see [Web Application Proxy in Windows Server 2016](#).

Virtual Private Networking (VPN)

In Windows Server 2016, **DirectAccess and VPN** is a role service of the **Remote Access** server role.

When you install Remote Access as a VPN server, you can use Virtual Private Networking (VPN) to provide your remote employees with connections to your organization network across the Internet - while also maintaining information privacy with encrypted connections.

With Windows Server 2016 Remote Access VPN - and Windows 10 client computers - you can now deploy Always On VPN. Always On VPN gives you the ability to manage remote VPN clients that are always connected, while also providing convenience for remote workers, who no longer need to manually connect to and disconnect from VPN to your organization network.

For more information, see [Remote Access Always On VPN Deployment Guide for Windows Server 2016 and Windows 10](#).

NOTE

VPN documentation is now located in the [Remote access and server management](#) section of the Windows Server 2016 table of contents, under [Remote Access](#).

For more information about VPN, see [Virtual Private Networking \(VPN\)](#).

Windows Container Networking

Windows Container Networking allows you to create and manage networks for connecting container endpoints on both Windows 10 and Windows Server hosts by using standard industry tools and workflows. Windows container networks support multiple topologies, including private, flat-L2, and routed-L3.

Also supported are overlays that you can create locally on the host by using Docker, Kubernetes, or Windows PowerShell through plugins that communicate with the Windows Host Networking Service (HNS). You can create

and manage multi-node cluster networks through higher level orchestration systems by communicating through a local agent to each node's HNS.

Windows Internet Name Service (WINS)

Windows Internet Name Service (WINS) is a legacy computer name registration and resolution service that maps computer NetBIOS names to IP addresses. Using DNS is recommended over using WINS.

Additional Resources

Networking resources for operating systems earlier than Windows Server 2016 are available at the following locations.

- Windows Server 2012 and Windows Server 2012 R2 [Networking Overview](#)
- Windows Server 2008 and Windows Server 2008 R2 [Networking](#)
- Windows Server 2003 [Windows Server 2003/2003 R2 Retired Content](#)

Windows Server supported networking scenarios

9/18/2018 • 4 minutes to read • [Edit Online](#)

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

This topic provides information about supported and unsupported scenarios that you can or cannot perform with this release of Windows Server 2016.

IMPORTANT

For all production scenarios, use the latest signed hardware drivers from your original equipment manufacturer (OEM) or independent hardware vendor (IHV).

Supported Networking Scenarios

This section includes information about the supported networking scenarios for Windows Server 2016, and includes the following scenario categories.

- [Software Defined Networking \(SDN\) scenarios](#)
- [Network Platform scenarios](#)
- [DNS Server scenarios](#)
- [IPAM scenarios with DHCP and DNS](#)
- [NIC Teaming scenarios](#)
- [Switch Embedded Teaming \(SET\) scenarios](#)

Software Defined Networking (SDN) scenarios

You can use the following documentation to deploy SDN scenarios with Windows Server 2016.

- [Deploy a Software Defined Network infrastructure using scripts](#)

For more information, see [Software Defined Networking \(SDN\)](#).

Network Controller scenarios

The Network Controller scenarios allow you to:

- Deploy and manage a multiple-node instance of Network Controller. For more information, see [Deploy Network Controller using Windows PowerShell](#).
- Use Network Controller to programmatically define network policy by using the REST Northbound API.
- Use Network Controller to create and manage virtual networks with Hyper-V Network Virtualization - using NVGRE or VXLAN encapsulation.

For more information, see [Network Controller](#).

Network Function Virtualization (NFV) scenarios

The NFV scenarios allow you to:

- Deploy and use a software load balancer to distribute both northbound and southbound traffic.

- Deploy and use a software load balancer to distribute eastbound and westbound traffic for virtual networks created with Hyper-V Network Virtualization.
- Deploy and use a NAT software load balancer for virtual networks created with Hyper-V Network Virtualization.
- Deploy and use a Layer 3 forwarding gateway
- Deploy and use a virtual private network (VPN) gateway for site-to-site IPsec (IKEv2) tunnels
- Deploy and use a Generic Routing Encapsulation (GRE) gateway.
- Deploy and configure dynamic routing and transit routing between sites using Border Gateway Protocol (BGP).
- Configure M+N redundancy for Layer 3 and site-to-site gateways, and for BGP routing.
- Use Network Controller to specify ACLs on virtual networks and network interfaces.

For more information, see [Network Function Virtualization](#).

Network Platform scenarios

For the scenarios in this section the Windows Server Networking team supports the use of any Windows Server 2016 certified driver. Please check with your network interface card (NIC) manufacturer to ensure you have the most recent driver updates.

The network platform scenarios allow you to:

- Use a converged NIC to combine both RDMA and Ethernet traffic using a single network adapter.
- Create a low-latency data path by using Packet Direct, enabled in the Hyper-V Virtual Switch, and a single network adapter.
- Configure SET to spread SMB Direct and RDMA traffic flows between up to two network adapters.

For more information, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#).

Hyper-V Virtual Switch Scenarios

The Hyper-V Virtual Switch scenarios allow you to:

- Create a Hyper-V Virtual Switch with a Remote Direct Memory Access (RDMA) vNIC
- Create a Hyper-V Virtual Switch with Switch Embedded Teaming (SET) and RDMA vNICs
- Create a SET team in Hyper-V Virtual Switch
- Manage a SET team by using Windows PowerShell commands

For more information, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#)

DNS Server scenarios

DNS Server scenarios allow you to:

- Specify Geo-Location based traffic management using DNS Policies
- Configure split-brain DNS using DNS Policies
- Apply filters on DNS queries using DNS Policies
- Configure Application Load Balancing using DNS Policies
- Specify Intelligent DNS Responses based on the time of day

- Configure DNS Zone transfer policies
- Configure DNS server policies on Active Directory Domain Services (AD DS) integrated zones
- Configure Response Rate Limiting
- Specify DNS-based Authentication of Named Entities (DANE)
- Configure support for Unknown Records in DNS

For more information, see the topics [What's New in DNS Client in Windows Server 2016](#) and [What's New in DNS Server in Windows Server 2016](#).

IPAM scenarios with DHCP and DNS

The IPAM scenarios allow you to:

- Discover and administer DNS and DHCP servers and IP addressing across multiple federated Active Directory forests
- Use IPAM for centralized management of DNS properties, including zones and resource records.
- Define granular role-based access control policies and delegate IPAM users or user groups to manage the set of DNS properties that you specify.
- Use the Windows PowerShell commands for IPAM to automate access control configuration for DHCP and DNS.

For more information, see [Manage IPAM](#).

NIC Teaming scenarios

The NIC Teaming scenarios allow you to:

- Create a NIC team in a supported configuration
- Delete a NIC team
- Add network adapters to the NIC team in a supported configuration
- Remove network adapters from the NIC team

NOTE

In Windows Server 2016, you can use NIC Teaming in Hyper-V, however in some cases Virtual Machine Queues (VMQ) might not automatically enable on the underlying network adapters when you create a NIC Team. If this occurs, you can use the following Windows PowerShell command to ensure that VMQ is enabled on the NIC team member adapters:

```
Set-NetAdapterVmq -Name <NetworkAdapterName> -Enable
```

For more information, see [NIC Teaming](#).

Switch Embedded Teaming (SET) scenarios

SET is an alternative NIC Teaming solution that you can use in environments that include Hyper-V and the Software Defined Networking (SDN) stack in Windows Server 2016. SET integrates some NIC Teaming functionality into the Hyper-V Virtual Switch.

For more information, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#)

Unsupported Networking Scenarios

The following networking scenarios are not supported in Windows Server 2016.

- VLAN-based tenant virtual networks.
- IPv6 is not supported in either the underlay or overlay.

What's new in networking

9/18/2018 • 8 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016

Following are the new or enhanced networking technologies in Windows Server 2016.

This topic contains the following sections.

- [New Networking Features and Technologies](#)
- [New Features for Additional Networking Technologies](#)

New Networking Features and Technologies

Networking is a foundational part of the Software Defined Datacenter (SDDC) platform, and Windows Server 2016 provides new and improved Software Defined Networking (SDN) technologies to help you move to a fully realized SDDC solution for your organization.

When you manage networks as a software defined resource, you can describe an application's infrastructure requirements one time, and then choose where the application runs - on premises or in the cloud. This consistency means that your applications are now easier to scale and you can seamlessly run applications , anywhere, with equal confidence around security, performance, quality of service, and availability.

The following sections contain information about these new networking features and technologies.

Software Defined Networking Infrastructure

Following are the new or improved SDN infrastructure technologies.

- **Network Controller.** New in Windows Server 2016, Network Controller provides a centralized, programmable point of automation to manage, configure, monitor, and troubleshoot virtual and physical network infrastructure in your datacenter. Using Network Controller, you can automate the configuration of network infrastructure instead of performing manual configuration of network devices and services. For more information, see [Network Controller](#) and [Deploy Software Defined Networks using scripts](#).
- **Hyper-V Virtual Switch.** The Hyper-V Virtual Switch runs on Hyper-V hosts, and allows you to create distributed switching and routing, and a policy enforcement layer that is aligned and compatible with Microsoft Azure. For more information, see [Hyper-V Virtual Switch](#).
- **Network Function Virtualization (NFV).** In today's software defined datacenters, network functions that are being performed by hardware appliances (such as load balancers, firewalls, routers, switches, and so on) are increasingly being deployed as virtual appliances. This "network function virtualization" is a natural progression of server virtualization and network virtualization. Virtual appliances are quickly emerging and creating a brand new market. They continue to generate interest and gain momentum in both virtualization platforms and cloud services. The following NFV technologies are available in Windows Server 2016.
 - **Datacenter Firewall.** This distributed firewall provides granular access control lists (ACLs), enabling you to apply firewall policies at the VM interface level or at the subnet level.
For more information, see [Datacenter Firewall Overview](#).
 - **RAS Gateway.** You can use RAS Gateway for routing traffic between virtual networks and physical networks, including site-to-site VPN connections from your cloud datacenter to your tenants' remote

sites. Specifically, you can deploy Internet Key Exchange version 2 (IKEv2) site-to-site virtual private networks (VPNs), Layer 3 (L3) VPN, and Generic Routing Encapsulation (GRE) gateways. In addition, gateway pools and M+N redundancy of gateways are now supported; and Border Gateway Protocol (BGP) with Route Reflector capabilities provides dynamic routing between networks for all gateway scenarios (IKEv2 VPN, GRE VPN, and L3 VPN).

For more information, see [What's New in RAS Gateway](#) and [RAS Gateway for SDN](#).

- **Software Load Balancer (SLB) and Network Address Translation (NAT)**. The north-south and east-west layer 4 load balancer and NAT enhances throughput by supporting Direct Server Return, with which the return network traffic can bypass the Load Balancing multiplexer.

For more information, see [Software Load Balancing \(SLB\) for SDN](#).

For more information, see [Network Function Virtualization](#).

- **Standardized Protocols**. Network Controller uses Representational State Transfer (REST) on its northbound interface with JavaScript Object Notation (JSON) payloads. The Network Controller southbound interface uses Open vSwitch Database Management Protocol (OVSDB).
- **Flexible encapsulation technologies**. These technologies operate at the data plane, and support both Virtual Extensible LAN (VxLAN) and Network Virtualization Generic Routing Encapsulation (NVGRE). For more information, see [GRE Tunneling in Windows Server 2016](#).

For more information about SDN, see [Software Defined Networking \(SDN\)](#).

Cloud Scale Fundamentals

The following cloud scale fundamentals are now available.

- **Converged Network Interface Card (NIC)**. The converged NIC allows you to use a single network adapter for management, Remote Direct Memory Access (RDMA)-enabled storage, and tenant traffic. This reduces the capital expenditures that are associated with each server in your datacenter, because you need fewer network adapters to manage different types of traffic per server.
- **Packet Direct**. Packet Direct provides a high network traffic throughput and low-latency packet processing infrastructure.
- **Switch Embedded Teaming (SET)**. SET is a NIC Teaming solution that is integrated in the Hyper-V Virtual Switch. SET allows the teaming of up to eight physical NICs into a single SET team, which improves availability and provides failover. In Windows Server 2016, you can create SET teams that are restricted to the use of Server Message Block (SMB) and RDMA. In addition, you can use SET teams to distribute network traffic for Hyper-V Network Virtualization. For more information, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#).

New Features for Additional Networking Technologies

This section contains information about new features for familiar networking technologies.

DHCP

DHCP is an Internet Engineering Task Force (IETF) standard that is designed to reduce the administrative burden and complexity of configuring hosts on a TCP/IP-based network, such as a private intranet. By using the DHCP Server service, the process of configuring TCP/IP on DHCP clients is automatic.

For more information, see [What's New in DHCP](#).

DNS

DNS is a system that is used in TCP/IP networks for naming computers and network services. DNS naming locates computers and services through user-friendly names. When a user enters a DNS name in an application, DNS services can resolve the name to other information that is associated with the name, such as an IP address.

Following is information about DNS Client and DNS Server.

DNS Client

Following are the new or improved DNS client technologies.

- **DNS Client service binding.** In Windows 10, the DNS Client service offers enhanced support for computers with more than one network interface.

For more information, see [What's New in DNS Client in Windows Server 2016](#)

DNS Server

Following are the new or improved DNS server technologies.

- **DNS Policies.** You can configure DNS policies to specify how a DNS server responds to DNS queries. DNS responses can be based on client IP address (location), time of the day, and several other parameters. DNS policies enable location-aware DNS, traffic management, load balancing, split-brain DNS, and other scenarios.
- **Nano Server support for file based DNS.** You can deploy DNS server in Windows Server 2016 on a Nano Server image. This deployment option is available to you if you are using file based DNS. By running DNS server on a Nano Server image, you can run your DNS servers with reduced footprint, quick boot up, and minimized patching.

NOTE

Active Directory integrated DNS is not supported on Nano Server.

- **Response Rate Limiting (RRL).** You can enable response rate limiting on your DNS servers. By doing this, you avoid the possibility of malicious systems using your DNS servers to initiate a denial of service attack on a DNS client.
- **DNS-based Authentication of Named Entities (DANE).** You can use TLSA (Transport Layer Security Authentication) records to provide information to DNS clients that state what certification authority (CA) they should expect a certificate from for your domain name. This prevents man-in-the-middle attacks where someone might corrupt the DNS cache to point to their own website, and provide a certificate they issued from a different CA.
- **Unknown record support.**
You can add records which are not explicitly supported by the Windows DNS server using the unknown record functionality.
- **IPv6 root hints.**
You can use the native IPv6 root hints support to perform internet name resolution using the IPv6 root servers.
- **Improved Windows PowerShell Support.**
New Windows PowerShell cmdlets are available for DNS Server.

For more information, see [What's New in DNS Server in Windows Server 2016](#)

GRE Tunneling

RAS Gateway now supports high availability Generic Routing Encapsulation (GRE) tunnels for site to site

connections and M+N redundancy of gateways. GRE is a lightweight tunneling protocol that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links over an Internet Protocol internetwork.

For more information, see [GRE Tunneling in Windows Server 2016](#).

Hyper-V Network Virtualization

Introduced in Windows Server 2012, Hyper-V Network Virtualization (HNV) enables virtualization of customer networks on top of a shared physical network infrastructure. With minimal changes necessary on the physical network fabric, HNV gives service providers the agility to deploy and migrate tenant workloads anywhere across the three clouds: the service provider cloud, the private cloud, or the Microsoft Azure public cloud.

For more information, see [What's New in Hyper-V Network Virtualization in Windows Server 2016](#)

IPAM

IPAM provides highly customizable administrative and monitoring capabilities for the IP address and DNS infrastructure on an organization network. Using IPAM, you can monitor, audit, and manage servers that are running Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS).

- **Enhanced IP address management.**

IPAM capabilities are improved for scenarios such as handling IPv4 /32 and IPv6 /128 subnets and finding free IP address subnets and ranges in an IP address block.

- **Enhanced DNS service management.**

IPAM supports DNS resource record, conditional forwarder, and DNS zone management for both domain-joined Active Directory-integrated and file-backed DNS servers.

- **Integrated DNS, DHCP, and IP address (DDI) management.**

Several new experiences and integrated lifecycle management operations are enabled, such as visualizing all DNS resource records that pertain to an IP address, automated inventory of IP addresses based on DNS resource records, and IP address lifecycle management for both DNS and DHCP operations.

- **Multiple Active Directory Forest support.**

You can use IPAM to manage the DNS and DHCP servers of multiple Active Directory forests when there is a two-way trust relationship between the forest where IPAM is installed and each of the remote forests.

- **Windows PowerShell support for Role Based Access Control.**

You can use Windows PowerShell to set access scopes on IPAM objects.

For more information, see [What's New in IPAM](#) and [Manage IPAM](#).

Core network guidance for Windows Server

9/18/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server, Windows Server 2016

This topic provides an overview of the Core network guidance for Windows Server® 2016, and contains the following sections.

- [Introduction to the Windows Server Core Network](#)
- [Core Network Guide for Windows Server](#)

Introduction to the Windows Server Core Network

A core network is a collection of network hardware, devices, and software that provides the fundamental services for your organization's information technology (IT) needs.

A Windows Server core network provides you with many benefits, including the following.

- Core protocols for network connectivity between computers and other Transmission Control Protocol/Internet Protocol (TCP/IP) compatible devices. TCP/IP is a suite of standard protocols for connecting computers and building networks. TCP/IP is network protocol software provided with Microsoft® Windows® operating systems that implements and supports the TCP/IP protocol suite.
- Dynamic Host Configuration Protocol (DHCP) server automatic IP addressing. Manual configuration of IP addresses on all computers on your network is time-consuming and less flexible than dynamically providing computers and other devices with IP address leases from a DHCP server.
- Domain Name System (DNS) name resolution service. DNS allows users, computers, applications, and services to find the IP addresses of computers and devices on the network by using the Fully Qualified Domain Name of the computer or device.
- A forest, which is one or more Active Directory domains that share the same class and attribute definitions (schema), site and replication information (configuration), and forest-wide search capabilities (global catalog).
- A forest root domain, which is the first domain created in a new forest. The Enterprise Admins and Schema Admins groups, which are forest-wide administrative groups, are located in the forest root domain. In addition, a forest root domain, as with other domains, is a collection of computer, user, and group objects that are defined by the administrator in Active Directory Domain Services (AD DS). These objects share a common directory database and security policies. They can also share security relationships with other domains if you add domains as your organization grows. The directory service also stores directory data and allows authorized computers, applications, and users to access the data.
- A user and computer account database. The directory service provides a centralized user accounts database that allows you to create user and computer accounts for people and computers that are authorized to connect to your network and access network resources, such as applications, databases, shared files and folders, and printers.

A core network also allows you to scale your network as your organization grows and IT requirements change. For example, with a core network you can add domains, IP subnets, remote access services, wireless services, and other features and server roles provided by Windows Server 2016.

Core Network Guide for Windows Server

The Windows Server 2016 Core Network Guide provides instructions on how to plan and deploy the core components required for a fully functioning network and a new Active Directory® domain in a new forest. Using this guide, you can deploy computers configured with the following Windows server components:

- The Active Directory Domain Services (AD DS) server role
- The Domain Name System (DNS) server role
- The Dynamic Host Configuration Protocol (DHCP) server role
- The Network Policy Server (NPS) role service of the Network Policy and Access Services server role
- The Web Server (IIS) server role
- Transmission Control Protocol/Internet Protocol version 4 (TCP/IP) connections on individual servers

This guide is available at the following location.

- The [Core Network Guide](#) in the Windows Server 2016 Technical Library.

Core network components

9/18/2018 • 72 minutes to read • [Edit Online](#)

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

This guide provides instructions on how to plan and deploy the core components required for a fully functioning network and a new Active Directory domain in a new forest.

NOTE

This guide is available for download in Microsoft Word format from TechNet Gallery. For more information, see [Core Network Guide for Windows Server 2016](#).

This guide contains the following sections.

- [About this guide](#)
- [Core Network Overview](#)
- [Core Network Planning](#)
- [Core Network Deployment](#)
- [Additional Technical Resources](#)
- [Appendices A through E](#)

About this guide

This guide is designed for network and system administrators who are installing a new network or who want to create a domain-based network to replace a network that consists of workgroups. The deployment scenario provided in this guide is particularly useful if you foresee the need to add more services and features to your network in the future.

It is recommended that you review design and deployment guides for each of the technologies used in this deployment scenario to assist you in determining whether this guide provides the services and configuration that you need.

A core network is a collection of network hardware, devices, and software that provides the fundamental services for your organization's information technology (IT) needs.

A Windows Server core network provides you with many benefits, including the following.

- Core protocols for network connectivity between computers and other Transmission Control Protocol/Internet Protocol (TCP/IP) compatible devices. TCP/IP is a suite of standard protocols for connecting computers and building networks. TCP/IP is network protocol software provided with Microsoft Windows operating systems that implements and supports the TCP/IP protocol suite.
- Dynamic Host Configuration Protocol (DHCP) automatic IP address assignment to computers and other devices that are configured as DHCP clients. Manual configuration of IP addresses on all computers on your network is time-consuming and less flexible than dynamically providing computers and other devices with IP address configurations using a DHCP server.
- Domain Name System (DNS) name resolution service. DNS allows users, computers, applications, and

services to find the IP addresses of computers and devices on the network by using the Fully Qualified Domain Name of the computer or device.

- A forest, which is one or more Active Directory domains that share the same class and attribute definitions (schema), site and replication information (configuration), and forest-wide search capabilities (global catalog).
- A forest root domain, which is the first domain created in a new forest. The Enterprise Admins and Schema Admins groups, which are forest-wide administrative groups, are located in the forest root domain. In addition, a forest root domain, as with other domains, is a collection of computer, user, and group objects that are defined by the administrator in Active Directory Domain Services (AD DS). These objects share a common directory database and security policies. They can also share security relationships with other domains if you add domains as your organization grows. The directory service also stores directory data and allows authorized computers, applications, and users to access the data.
- A user and computer account database. The directory service provides a centralized user accounts database that allows you to create user and computer accounts for people and computers that are authorized to connect to your network and access network resources, such as applications, databases, shared files and folders, and printers.

A core network also allows you to scale your network as your organization grows and IT requirements change. For example, with a core network you can add domains, IP subnets, remote access services, wireless services, and other features and server roles provided by Windows Server 2016.

Network hardware requirements

To successfully deploy a core network, you must deploy network hardware, including the following:

- Ethernet, Fast Ethernet, or Gigabyte Ethernet cabling
- A hub, Layer 2 or 3 switch, router, or other device that performs the function of relaying network traffic between computers and devices.
- Computers that meet the minimum hardware requirements for their respective client and server operating systems.

What this guide does not provide

This guide does not provide instructions for deploying the following:

- Network hardware, such as cabling, routers, switches, and hubs
- Additional network resources, such as printers and file servers
- Internet connectivity
- Remote access
- Wireless access
- Client computer deployment

NOTE

Computers running Windows client operating systems are configured by default to receive IP address leases from the DHCP server. Therefore, no additional DHCP or Internet Protocol version 4 (IPv4) configuration of client computers is required.

Technology Overviews

The following sections provide brief overviews of the required technologies that are deployed to create a core network.

Active Directory Domain Services

A directory is a hierarchical structure that stores information about objects on the network, such as users and computers. A directory service, such as AD DS, provides the methods for storing directory data and making this data available to network users and administrators. For example, AD DS stores information about user accounts, including names, email addresses, passwords, and phone numbers, and enables other authorized users on the same network to access this information.

DNS

DNS is a name resolution protocol for TCP/IP networks, such as the Internet or an organization network. A DNS server hosts the information that enables client computers and services to resolve easily recognized, alphanumeric DNS names to the IP addresses that computers use to communicate with each other.

DHCP

DHCP is an IP standard for simplifying the management of host IP configuration. The DHCP standard provides for the use of DHCP servers as a way to manage dynamic allocation of IP addresses and other related configuration details for DHCP-enabled clients on your network.

DHCP allows you to use a DHCP server to dynamically assign an IP address to a computer or other device, such as a printer, on your local network. Every computer on a TCP/IP network must have a unique IP address, because the IP address and its related subnet mask identify both the host computer and the subnet to which the computer is attached. By using DHCP, you can ensure that all computers that are configured as DHCP clients receive an IP address that is appropriate for their network location and subnet, and by using DHCP options, such as default gateway and DNS servers, you can automatically provide DHCP clients with the information that they need to function correctly on your network.

For TCP/IP-based networks, DHCP reduces the complexity and amount of administrative work involved in reconfiguring computers.

TCP/IP

TCP/IP in Windows Server 2016 is the following:

- Networking software based on industry-standard networking protocols.
- A routable enterprise networking protocol that supports the connection of your Windows-based computer to both local area network (LAN) and wide area network (WAN) environments.
- Core technologies and utilities for connecting your Windows-based computer with dissimilar systems for the purpose of sharing information.
- A foundation for gaining access to global Internet services, such as the World Wide Web and File Transfer Protocol (FTP) servers.
- A robust, scalable, cross-platform, client/server framework.

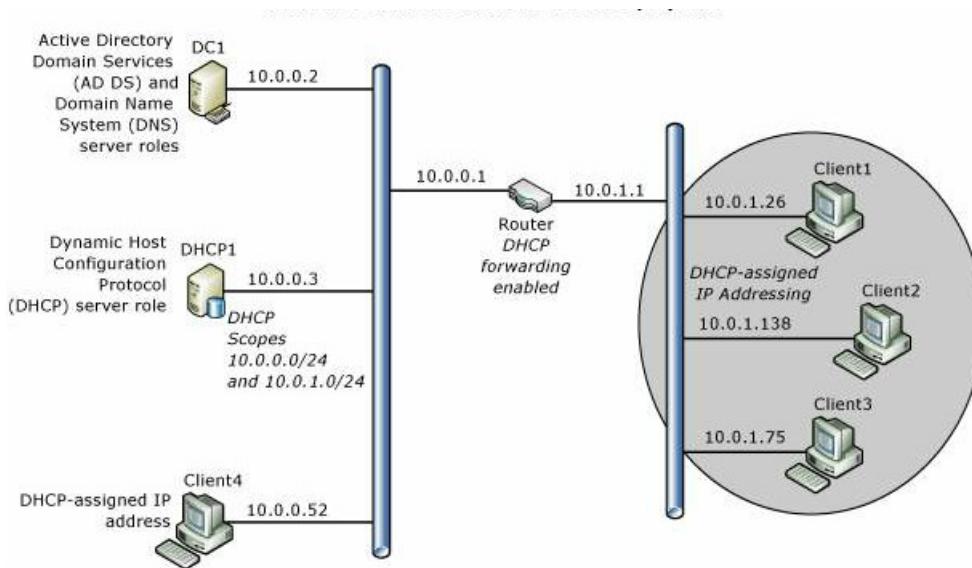
TCP/IP provides basic TCP/IP utilities that enable Windows-based computers to connect and share information with other Microsoft and non-Microsoft systems, including:

- Windows Server 2016
- Windows 10
- Windows Server 2012 R2
- Windows 8.1
- Windows Server 2012

- Windows 8
- Windows Server 2008 R2
- Windows 7
- Windows Server 2008
- Windows Vista
- Internet hosts
- Apple Macintosh systems
- IBM mainframes
- UNIX and Linux systems
- Open VMS systems
- Network-ready printers
- Tablets and cellular telephones with wired Ethernet or wireless 802.11 technology enabled

Core Network Overview

The following illustration shows the Windows Server Core Network topology.



NOTE

This guide also includes instructions for adding optional Network Policy Server (NPS) and Web Server (IIS) servers to your network topology to provide the foundation for secure network access solutions, such as 802.1X wired and wireless deployments that you can implement using Core Network Companion guides. For more information, see [Deploying optional features for network access authentication and Web services](#).

Core Network Components

Following are the components of a core network.

Router

This deployment guide provides instructions for deploying a core network with two subnets separated by a router that has DHCP forwarding enabled. You can, however, deploy a Layer 2 switch, a Layer 3 switch, or a hub, depending on your requirements and resources. If you deploy a switch, the switch must be capable of DHCP

forwarding or you must place a DHCP server on each subnet. If you deploy a hub, you are deploying a single subnet and do not need DHCP forwarding or a second scope on your DHCP server.

Static TCP/IP configurations

The servers in this deployment are configured with static IPv4 addresses. Client computers are configured by default to receive IP address leases from the DHCP server.

Active Directory Domain Services global catalog and DNS server DC1

Both Active Directory Domain Services (AD DS) and Domain Name System (DNS) are installed on this server, named DC1, which provides directory and name resolution services to all computers and devices on the network.

DHCP server DHCP1

The DHCP server, named DHCP1, is configured with a scope that provides Internet Protocol (IP) address leases to computers on the local subnet. The DHCP server can also be configured with additional scopes to provide IP address leases to computers on other subnets if DHCP forwarding is configured on routers.

Client computers

Computers running Windows client operating systems are configured by default as DHCP clients, which obtain IP addresses and DHCP options automatically from the DHCP server.

Core Network Planning

Before you deploy a core network, you must plan the following items.

- [Planning subnets](#)
- [Planning basic configuration of all servers](#)
- [Planning the deployment of DC1](#)
- [Planning domain access](#)
- [Planning the deployment of DHCP1](#)

The following sections provide more detail on each of these items.

NOTE

For assistance with planning your deployment, also see [Appendix E - Core Network Planning Preparation Sheet](#).

Planning subnets

In Transmission Control Protocol/Internet Protocol (TCP/IP) networking, routers are used to interconnect the hardware and software used on different physical network segments called subnets. Routers are also used to forward IP packets between each of the subnets. Determine the physical layout of your network, including the number of routers and subnets you need, before proceeding with the instructions in this guide.

In addition, to configure the servers on your network with static IP addresses, you must determine the IP address range that you want to use for the subnet where your core network servers are located. In this guide, the private IP address ranges 10.0.0.1 - 10.0.0.254 and 10.0.1.1 - 10.0.1.254 are used as examples, but you can use any private IP address range that you prefer.

IMPORTANT

After you select the IP address ranges that you want to use for each subnet, ensure that you configure your routers with an IP address from the same IP address range as that used on the subnet where the router is installed. For example, if your router is configured by default with an IP address of 192.168.1.1, but you are installing the router on a subnet with an IP address range of 10.0.0.0/24, you must reconfigure the router to use an IP address from the 10.0.0.0/24 IP address range.

The following recognized private IP address ranges are specified by Internet Request for Comments (RFC) 1918:

- 10.0.0.0 - 10.255.255.255
- 172.16.0.0 - 172.31.255.255
- 192.168.0.0 - 192.168.255.255

When you use the private IP address ranges as specified in RFC 1918, you cannot connect directly to the Internet using a private IP address because requests going to or from these addresses are automatically discarded by Internet service provider (ISP) routers. To add Internet connectivity to your core network later, you must contract with an ISP to obtain a public IP address.

IMPORTANT

When using private IP addresses, you must use some type of proxy or network address translation (NAT) server to convert the private IP address ranges on your local network to a public IP address that can be routed on the Internet. Most routers provide NAT services, so selecting a router that is NAT-capable should be fairly simple.

For more information, see [Planning the deployment of DHCP1](#).

Planning basic configuration of all servers

For each server in the core network, you must rename the computer and assign and configure a static IPv4 address and other TCP/IP properties for the computer.

Planning naming conventions for computers and devices

For consistency across your network, it is a good idea to use consistent names for servers, printers, and other devices. Computer names can be used to help users and administrators easily identify the purpose and location of the server, printer, or other device. For example, if you have three DNS servers, one in San Francisco, one in Los Angeles, and one in Chicago, you might use the naming convention *server function-location-number*:

- DNS-DEN-01. This name represents the DNS server in Denver, Colorado. If additional DNS servers are added in Denver, the numeric value in the name can be incremented, as in DNS-DEN-02 and DNS-DEN-03.
- DNS-SPAS-01. This name represents the DNS server in South Pasadena, California.
- DNS-ORL-01. This name represents the DNS server in Orlando, Florida.

For this guide, the server naming convention is very simple, and consists of the primary server function and a number. For example, the domain controller is named DC1 and the DHCP server is named DHCP1.

It is recommended that you choose a naming convention before you install your core network using this guide.

Planning static IP addresses

Before configuring each computer with a static IP address, you must plan your subnets and IP address ranges. In addition, you must determine the IP addresses of your DNS servers. If you plan to install a router that provides access to other networks, such as additional subnets or the Internet, you must know the IP address of the router, also called a default gateway, for static IP address configuration.

The following table provides example values for static IP address configuration.

CONFIGURATION ITEMS	EXAMPLE VALUES
IP address	10.0.0.2
Subnet mask	255.255.255.0

CONFIGURATION ITEMS	EXAMPLE VALUES
Default gateway (Router IP address)	10.0.0.1
Preferred DNS server	10.0.0.2

NOTE

If you plan on deploying more than one DNS server, you can also plan the Alternate DNS Server IP address.

Planning the deployment of DC1

Following are key planning steps before installing Active Directory Domain Services (AD DS) and DNS on DC1.

Planning the name of the forest root domain

A first step in the AD DS design process is to determine how many forests your organization requires. A forest is the top-level AD DS container, and consists of one or more domains that share a common schema and global catalog. An organization can have multiple forests, but for most organizations, a single forest design is the preferred model and the simplest to administer.

When you create the first domain controller in your organization, you are creating the first domain (also called the forest root domain) and the first forest. Before you take this action using this guide, however, you must determine the best domain name for your organization. In most cases, the organization name is used as the domain name, and in many cases this domain name is registered. If you are planning to deploy external-facing Internet based Web servers to provide information and services for your customers or partners, choose a domain name that is not already in use, and then register the domain name so that your organization owns it.

Planning the forest functional level

While installing AD DS, you must choose the forest functional level that you want to use. Domain and forest functionality, introduced in Windows Server 2003 Active Directory, provides a way to enable domain- or forest-wide Active Directory features within your network environment. Different levels of domain functionality and forest functionality are available, depending on your environment.

Forest functionality enables features across all the domains in your forest. The following forest functional levels are available:

- Windows Server 2008 . This forest functional level supports only domain controllers that are running Windows Server 2008 and later versions of the Windows Server operating system.
- Windows Server 2008 R2 . This forest functional level supports Windows Server 2008 R2 domain controllers and domain controllers that are running later versions of the Windows Server operating system.
- Windows Server 2012 . This forest functional level supports Windows Server 2012 domain controllers and domain controllers that are running later versions of the Windows Server operating system.
- Windows Server 2012 R2 . This forest functional level supports Windows Server 2012 R2 domain controllers and domain controllers that are running later versions of the Windows Server operating system.
- Windows Server 2016. This forest functional level supports only Windows Server 2016 domain controllers and domain controllers that are running later versions of the Windows Server operating system.

If you are deploying a new domain in a new forest and all of your domain controllers will be running Windows Server 2016, it is recommended that you configure AD DS with the Windows Server 2016 forest functional level during AD DS installation.

IMPORTANT

After the forest functional level is raised, domain controllers that are running earlier operating systems cannot be introduced into the forest. For example, if you raise the forest functional level to Windows Server 2016, domain controllers running Windows Server 2012 R2 or Windows Server 2008 cannot be added to the forest.

Example configuration items for AD DS are provided in the following table.

CONFIGURATION ITEMS:	EXAMPLE VALUES:
Full DNS name	Examples: - corp.contoso.com - example.com
Forest functional level	- Windows Server 2008 - Windows Server 2008 R2 - Windows Server 2012 - Windows Server 2012 R2 - Windows Server 2016
Active Directory Domain Services Database folder location	E:\Configuration\ Or accept the default location.
Active Directory Domain Services Log files folder location	E:\Configuration\ Or accept the default location.
Active Directory Domain Services SYSVOL folder location	E:\Configuration\ Or accept the default location
Directory Restore Mode Administrator Password	J*p2leO4\$F
Answer file name (optional)	AD DS_AnswerFile

Planning DNS zones

On primary, Active Directory-integrated DNS servers, a forward lookup zone is created by default during installation of the DNS Server role. A forward lookup zone allows computers and devices to query for another computer's or device's IP address based on its DNS name. In addition to a forward lookup zone, it is recommended that you create a DNS reverse lookup zone. With a DNS reverse lookup query, a computer or device can discover the name of another computer or device using its IP address. Deploying a reverse lookup zone typically improves DNS performance and greatly increases the success of DNS queries.

When you create a reverse lookup zone, the in-addr.arpa domain, which is defined in the DNS standards and reserved in the Internet DNS namespace to provide a practical and reliable way to perform reverse queries, is configured in DNS. To create the reverse namespace, subdomains within the in-addr.arpa domain are formed, using the reverse ordering of the numbers in the dotted-decimal notation of IP addresses.

The in-addr.arpa domain applies to all TCP/IP networks that are based on Internet Protocol version 4 (IPv4) addressing. The New Zone Wizard automatically assumes that you are using this domain when you create a new reverse lookup zone.

While you are running the New Zone Wizard, the following selections are recommended:

CONFIGURATION ITEMS	EXAMPLE VALUES
Zone type	Primary zone , and Store the zone in Active Directory is selected
Active Directory Zone Replication Scope	To all DNS servers in this domain
First Reverse Lookup Zone Name wizard page	IPv4 Reverse Lookup Zone
Second Reverse Lookup Zone Name wizard page	Network ID = 10.0.0.
Dynamic Updates	Allow only secure dynamic updates

Planning domain access

To log on to the domain, the computer must be a domain member computer and the user account must be created in AD DS before the logon attempt.

NOTE

Individual computers that are running Windows have a local users and groups user account database that is called the Security Accounts Manager (SAM) user accounts database. When you create a user account on the local computer in the SAM database, you can log onto the local computer, but you cannot log on to a domain. Domain user accounts are created with the Active Directory Users and Computers Microsoft Management Console (MMC) on a domain controller, not with local users and groups on the local computer.

After the first successful logon with domain logon credentials, the logon settings persist unless the computer is removed from the domain or the logon settings are manually changed.

Before you log on to the domain:

- Create user accounts in Active Directory Users and Computers. Each user must have an Active Directory Domain Services user account in Active Directory Users and Computers. For more information, see [Create a User Account in Active Directory Users and Computers](#).
- Ensure the correct IP address configuration. To join a computer to the domain, the computer must have an IP address. In this guide, servers are configured with static IP addresses and client computers receive IP address leases from the DHCP server. For this reason, the DHCP server must be deployed before you join clients to the domain. For more information, see [Deploying DHCP1](#).
- Join the computer to the domain. Any computer that provides or accesses network resources must be joined to the domain. For more information, see [Joining Server Computers to the Domain and Logging On](#) and [Joining Client Computers to the Domain and Logging On](#).

Planning the deployment of DHCP1

Following are key planning steps before installing the DHCP server role on DHCP1.

Planning DHCP servers and DHCP forwarding

Because DHCP messages are broadcast messages, they are not forwarded between subnets by routers. If you have multiple subnets and want to provide DHCP service for each subnet, you must do one of the following:

- Install a DHCP server on each subnet
- Configure routers to forward DHCP broadcast messages across subnets and configure multiple scopes on the DHCP server, one scope per subnet.

In most cases, configuring routers to forward DHCP broadcast messages is more cost effective than deploying a

DHCP server on each physical segment of the network.

Planning IP address ranges

Each subnet must have its own unique IP address range. These ranges are represented on a DHCP server with scopes.

A scope is an administrative grouping of IP addresses for computers on a subnet that use the DHCP service. The administrator first creates a scope for each physical subnet and then uses the scope to define the parameters used by clients.

A scope has the following properties:

- A range of IP addresses from which to include or exclude addresses used for DHCP service lease offerings.
- A subnet mask, which determines the subnet prefix for a given IP address.
- A scope name assigned when it is created.
- Lease duration values, which are assigned to DHCP clients that receive dynamically allocated IP addresses.
- Any DHCP scope options configured for assignment to DHCP clients, such as DNS server IP address and router/default gateway IP address.
- Reservations are optionally used to ensure that a DHCP client always receives the same IP address.

Before deploying your servers, list your subnets and the IP address range you want to use for each subnet.

Planning subnet masks

Network IDs and host IDs within an IP address are distinguished by using a subnet mask. Each subnet mask is a 32-bit number that uses consecutive bit groups of all ones (1) to identify the network ID and all zeroes (0) to identify the host ID portions of an IP address.

For example, the subnet mask normally used with the IP address 131.107.16.200 is the following 32-bit binary number:

```
11111111 11111111 00000000 00000000
```

This subnet mask number is 16 one-bits followed by 16 zero-bits, indicating that the network ID and host ID sections of this IP address are both 16 bits in length. Normally, this subnet mask is displayed in dotted decimal notation as 255.255.0.0.

The following table displays subnet masks for the Internet address classes.

ADDRESS CLASS	BITS FOR SUBNET MASK	SUBNET MASK
Class A	11111111 00000000 00000000 00000000	255.0.0.0
Class B	11111111 11111111 00000000 00000000	255.255.0.0
Class C	11111111 11111111 11111111 00000000	255.255.255.0

When you create a scope in DHCP and you enter the IP address range for the scope, DHCP provides these default subnet mask values. Typically, default subnet mask values are acceptable for most networks with no special requirements and where each IP network segment corresponds to a single physical network.

In some cases, you can use customized subnet masks to implement IP subnetting. With IP subnetting, you can

subdivide the default host ID portion of an IP address to specify subnets, which are subdivisions of the original class-based network ID.

By customizing the subnet mask length, you can reduce the number of bits that are used for the actual host ID.

To prevent addressing and routing problems, you should make sure that all TCP/IP computers on a network segment use the same subnet mask and that each computer or device has an unique IP address.

Planning exclusion ranges

When you create a scope on a DHCP server, you specify an IP address range that includes all of the IP addresses that the DHCP server is allowed to lease to DHCP clients, such as computers and other devices. If you then go and manually configure some servers and other devices with static IP addresses from the same IP address range that the DHCP server is using, you can accidentally create an IP address conflict, where you and the DHCP server have both assigned the same IP address to different devices.

To solve this problem, you can create an exclusion range for the DHCP scope. An exclusion range is a contiguous range of IP addresses within the scope's IP address range that the DHCP server is not allowed to use. If you create an exclusion range, the DHCP server does not assign the addresses in that range, allowing you to manually assign these addresses without creating an IP address conflict.

You can exclude IP addresses from distribution by the DHCP server by creating an exclusion range for each scope. You should use exclusions for all devices that are configured with a static IP address. The excluded addresses should include all IP addresses that you assigned manually to other servers, non-DHCP clients, diskless workstations, or Routing and Remote Access and PPP clients.

It is recommended that you configure your exclusion range with extra addresses to accommodate future network growth. The following table provides an example exclusion range for a scope with an IP address range of 10.0.0.1 - 10.0.0.254 and a subnet mask of 255.255.255.0.

CONFIGURATION ITEMS	EXAMPLE VALUES
Exclusion range Start IP Address	10.0.0.1
Exclusion range End IP Address	10.0.0.25

Planning TCP/IP static configuration

Certain devices, such as routers, DHCP servers, and DNS servers, must be configured with a static IP address. In addition, you might have additional devices, such as printers, that you want to ensure always have the same IP address. List the devices that you want to configure statically for each subnet, and then plan the exclusion range you want to use on the DHCP server to ensure that the DHCP server does not lease the IP address of a statically configured device. An exclusion range is a limited sequence of IP addresses within a scope, excluded from DHCP service offerings. Exclusion ranges assure that any addresses in these ranges are not offered by the server to DHCP clients on your network.

For example, if the IP address range for a subnet is 192.168.0.1 through 192.168.0.254 and you have ten devices that you want to configure with a static IP address, you can create an exclusion range for the 192.168.0.x scope that includes ten or more IP addresses: 192.168.0.1 through 192.168.0.15.

In this example, you use ten of the excluded IP addresses to configure servers and other devices with static IP addresses and five additional IP addresses are left available for static configuration of new devices that you might want to add in the future. With this exclusion range, the DHCP server is left with an address pool of 192.168.0.16 through 192.168.0.254.

Additional example configuration items for AD DS and DNS are provided in the following table.

CONFIGURATION ITEMS	EXAMPLE VALUES
Network Connect Bindings	Ethernet
DNS Server Settings	DC1.corp.contoso.com
Preferred DNS server IP address	10.0.0.2
Add Scope dialog box values 1. Scope Name 2. Starting IP Address 3. Ending IP Address 4. Subnet Mask 5. Default Gateway (optional) 6. Lease duration	1. Primary Subnet 2. 10.0.0.1 3. 10.0.0.254 4. 255.255.255.0 5. 10.0.0.1 6. 8 days
IPv6 DHCP Server Operation Mode	Not enabled

Core Network Deployment

To deploy a core network, the basic steps are as follows:

1. [Configuring All Servers](#)
2. [Deploying DC1](#)
3. [Joining Server Computers to the Domain and Logging On](#)
4. [Deploying DHCP1](#)
5. [Joining Client Computers to the Domain and Logging On](#)
6. [Deploying optional features for network access authentication and Web services](#)

NOTE

- Equivalent Windows PowerShell commands are provided for most procedures in this guide. Before running these cmdlets in Windows PowerShell, replace example values with values that are appropriate for your network deployment. In addition, you must enter each cmdlet on a single line in Windows PowerShell. In this guide, individual cmdlets might appear on several lines due to formatting constraints and the display of the document by your browser or other application.
- The procedures in this guide do not include instructions for those cases in which the **User Account Control** dialog box opens to request your permission to continue. If this dialog box opens while you are performing the procedures in this guide, and if the dialog box was opened in response to your actions, click **Continue**.

Configuring All Servers

Before installing other technologies, such as Active Directory Domain Services or DHCP, it is important to configure the following items.

- [Rename the computer](#)
- [Configure a static IP address](#)

You can use the following sections to perform these actions for each server.

Membership in **Administrators**, or equivalent, is the minimum required to perform these procedures.

Rename the computer

You can use the procedure in this section to change the name of a computer. Renaming the computer is useful for circumstances in which the operating system has automatically created a computer name that you do not want to use.

NOTE

To perform this procedure by using Windows PowerShell, open PowerShell and type the following cmdlets on separate lines, and then press ENTER. You must also replace *ComputerName* with the name that you want to use.

```
Rename-Computer ComputerName
```

```
Restart-Computer
```

To rename computers running Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012

1. In Server Manager, click **Local Server**. The computer **Properties** are displayed in the details pane.
2. In **Properties**, in **Computer name**, click the existing computer name. The **System Properties** dialog box opens. Click **Change**. The **Computer Name/Domain Changes** dialog box opens.
3. In the **Computer Name/Domain Changes** dialog box, in **Computer name**, type a new name for your computer. For example, if you want to name the computer DC1, type **DC1**.
4. Click **OK** twice, and then click **Close**. If you want to restart the computer immediately to complete the name change, click **Restart Now**. Otherwise, click **Restart Later**.

NOTE

For information on how to rename computers that are running other Microsoft operating systems, see [Appendix A - Renaming computers](#).

Configure a static IP address

You can use the procedures in this topic to configure the Internet Protocol version 4 (IPv4) properties of a network connection with a static IP address for computers running Windows Server 2016.

NOTE

To perform this procedure by using Windows PowerShell, open PowerShell and type the following cmdlets on separate lines, and then press ENTER. You must also replace interface names and IP addresses in this example with the values that you want to use to configure your computer.

```
New-NetIPAddress -IPAddress 10.0.0.2 -InterfaceAlias "Ethernet" -DefaultGateway 10.0.0.1 -AddressFamily IPv4  
-PrefixLength 24
```

```
Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses 127.0.0.1
```

To configure a static IP address on computers running Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012

1. In the task bar, right-click the Network icon, and then click **Open Network and Sharing Center**.
2. In **Network and Sharing Center**, click **Change adapter settings**. The **Network Connections** folder opens and displays the available network connections.
3. In **Network Connections**, right-click the connection that you want to configure, and then click **Properties**. The network connection **Properties** dialog box opens.
4. In the network connection **Properties** dialog box, in **This connection uses the following items**, select **Internet Protocol Version 4 (TCP/IPv4)**, and then click **Properties**. The **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box opens.
5. In **Internet Protocol Version 4 (TCP/IPv4) Properties**, on the **General** tab, click **Use the following IP**

address. In **IP address**, type the IP address that you want to use.

6. Press tab to place the cursor in **Subnet mask**. A default value for subnet mask is entered automatically. Either accept the default subnet mask, or type the subnet mask that you want to use.
7. In **Default gateway**, type the IP address of your default gateway.

NOTE

You must configure **Default gateway** with the same IP address that you use on the local area network (LAN) interface of your router. For example, if you have a router that is connected to a wide area network (WAN) such as the Internet as well as to your LAN, configure the LAN interface with the same IP address that you will then specify as the **Default gateway**. In another example, if you have a router that is connected to two LANs, where LAN A uses the address range 10.0.0.0/24 and LAN B uses the address range 192.168.0.0/24, configure the LAN A router IP address with an address from that address range, such as 10.0.0.1. In addition, in the DHCP scope for this address range, configure **Default gateway** with the IP address 10.0.0.1. For the LAN B, configure the LAN B router interface with an address from that address range, such as 192.168.0.1, and then configure the LAN B scope 192.168.0.0/24 with a **Default gateway** value of 192.168.0.1.

8. In **Preferred DNS server**, type the IP address of your DNS server. If you plan to use the local computer as the preferred DNS server, type the IP address of the local computer.
9. In **Alternate DNS Server**, type the IP address of your alternate DNS server, if any. If you plan to use the local computer as an alternate DNS server, type the IP address of the local computer.
10. Click **OK**, and then click **Close**.

NOTE

For information on how to configure a static IP address on computers that are running other Microsoft operating systems, see [Appendix B - Configuring static IP addresses](#).

Deploying DC1

To deploy DC1, which is the computer running Active Directory Domain Services (AD DS) and DNS, you must complete these steps in the following order:

- Perform the steps in the section [Configuring All Servers](#).
- [Install AD DS and DNS for a New Forest](#)
- [Create a User Account in Active Directory Users and Computers](#)
- [Assign Group Membership](#)
- [Configure a DNS Reverse Lookup Zone](#)

Administrative privileges

If you are installing a small network and are the only administrator for the network, it is recommended that you create a user account for yourself, and then add your user account as a member of both Enterprise Admins and Domain Admins. Doing so will make it easier for you to act as the administrator for all network resources. It is also recommended that you log on with this account only when you need to perform administrative tasks, and that you create a separate user account for performing non-IT related tasks.

If you have a larger organization with multiple administrators, refer to AD DS documentation to determine the best group membership for organization employees.

Differences between domain user accounts and user accounts on the local computer

One of the advantages of a domain-based infrastructure is that you do not need to create user accounts on each computer in the domain. This is true whether the computer is a client computer or a server.

Because of this, you should not create user accounts on each computer in the domain. Create all user accounts in Active Directory Users and Computers and use the preceding procedures to assign group membership. By default, all user accounts are members of the Domain Users group.

All members of the Domain Users group can log on to any client computer after it is joined to the domain.

You can configure user accounts to designate the days and times that the user is allowed to log on to the computer. You can also designate which computers each user is allowed to use. To configure these settings, open Active Directory Users and Computers, locate the user account that you want to configure, and double-click the account. In the user account **Properties**, click the **Account** tab, and then click either **Logon Hours** or **Log On To**.

Install AD DS and DNS for a New Forest

You can use one of the following procedures to install Active Directory Domain Services (AD DS) and DNS and to create a new domain in a new forest.

The first procedure provides instructions on performing these actions by using Windows PowerShell, while the second procedure shows you how to install AD DS and DNS by using Server Manager.

IMPORTANT

After you finish performing the steps in this procedure, the computer is automatically restarted.

Install AD DS and DNS Using Windows PowerShell

You can use the following commands to install and configure AD DS and DNS. You must replace the domain name in this example with the value that you want to use for your domain.

NOTE

For more information about these Windows PowerShell commands, see the following reference topics.

- [Install-WindowsFeature](#)
- [Install-ADDSForest](#)

Membership in **Administrators** is the minimum required to perform this procedure.

- Run Windows PowerShell as an Administrator, type the following command, and then press ENTER:

```
Install-WindowsFeature AD-Domain-Services -IncludeManagementTools
```

When installation has successfully completed, the following message is displayed in Windows PowerShell.

Success	Restart Needed	Exit Code	Feature Result
True	No	Success	{Active Directory Domain Services, Group P...

- In Windows PowerShell, type the following command, replacing the text **corp.contoso.com** with your domain name, and then press ENTER:

```
Install-ADDSForest -DomainName "corp.contoso.com"
```

- During the installation and configuration process, which is visible at the top of the Windows PowerShell window, the following prompt appears. After it appears, type a password and then press ENTER.

SafeModeAdministratorPassword:

- After you type a password and press ENTER, the following confirmation prompt appears. Type the same password and then press ENTER.

Confirm SafeModeAdministratorPassword:

- When the following prompt appears, type the letter **Y** and then press ENTER.

```
The target server will be configured as a domain controller and restarted when this operation is complete.  
Do you want to continue with this operation?  
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
```

- If you want to, you can read the warning messages that are displayed during normal, successful installation of AD DS and DNS. These messages are normal and are not an indication of install failure.
- After installation succeeds, a message appears stating that you are about to be logged off of the computer so that the computer can restart. If you click **Close**, you are immediately logged off the computer, and the computer restarts. If you do not click **Close**, the computer restarts after a default period of time.
- After the server is restarted, you can verify successful installation of Active Directory Domain Services and DNS. Open Windows PowerShell, type the following command, and press ENTER.

```
Get-WindowsFeature
```

The results of this command are displayed in Windows PowerShell, and should be similar to the results in the image below. For installed technologies, the brackets to the left of the technology name contain the character **X**, and the value of **Install State** is **Installed**.

Windows PowerShell Copyright (C) 2016 Microsoft Corporation. All rights reserved.		
PS D:\Users\jm> Get-WindowsFeature	Name	Install State
[] Active Directory Certificate Services	AD-Certificate	Available
[] Certification Authority	ADCS-Cert-Authority	Available
[] Certificate Enrollment Policy Web Service	ADCS-Enroll-Web-Pol	Available
[] Certificate Enrollment Web Service	ADCS-Enroll-Web-Svc	Available
[] Certification Authority Web Enrollment	ADCS-Web-Enrollment	Available
[] Network Device Enrollment Service	ADCS-Device-Enrollment	Available
[] Online Responder	ADCS-Online-Cert	Available
[X] Active Directory Domain Services	AD-Domain-Services	Installed
[] Active Directory Federation Services	ADFS-Federation	Available
[] Active Directory Lightweight Directory Services	AD LDS	Available
[] Active Directory Rights Management Services	AD RMS	Available
[] Active Directory Rights Management Server	AD RMS-Server	Available
[] Identity Federation Support	AD RMS-Identity	Available
[] Device Health Attestation	DeviceHealthAttestat...	Available
[] DHCP Server	DHCP	Available
[X] DNS Server	DNS	Installed

Install AD DS and DNS Using Server Manager

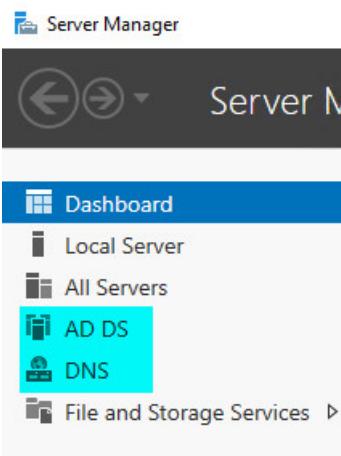
- On DC1, in **Server Manager**, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features Wizard opens.
- In **Before You Begin**, click **Next**.

NOTE

The **Before You Begin** page of the Add Roles and Features Wizard is not displayed if you have previously selected **Skip this page by default** when the Add Roles and Features Wizard was run.

- In **Select Installation Type**, ensure that **Role-Based or feature-based installation** is selected, and then click **Next**.

4. In **Select destination server**, ensure that **Select a server from the server pool** is selected. In **Server Pool**, ensure that the local computer is selected. Click **Next**.
5. In **Select server roles**, in **Roles**, click **Active Directory Domain Services**. In **Add features that are required for Active Directory Domain Services**, click **Add Features**. Click **Next**.
6. In **Select features**, click **Next**, and in **Active Directory Domain Services**, review the information that is provided, and then click **Next**.
7. In **Confirm installation selections**, click **Install**. The Installation progress page displays status during the installation process. When the process completes, in the message details, click **Promote this server to a domain controller**. The Active Directory Domain Services Configuration Wizard opens.
8. In **Deployment Configuration**, select **Add a new forest**. In **Root domain name**, type the fully qualified domain name (FQDN) for your domain. For example, if your FQDN is corp.contoso.com, type **corp.contoso.com**. Click **Next**.
9. In **Domain Controller Options**, in **Select functional level of the new forest and root domain**, select the forest functional level and domain functional level that you want to use. In **Specify domain controller capabilities**, ensure that **Domain Name System (DNS) server** and **Global Catalog (GC)** are selected. In **Password** and **Confirm password**, type the Directory Services Restore Mode (DSRM) password that you want to use. Click **Next**.
10. In **DNS Options**, click **Next**.
11. In **Additional Options**, verify the NetBIOS name that is assigned to the domain, and change it only if necessary. Click **Next**.
12. In **Paths**, in **Specify the location of the AD DS database, log files, and SYSVOL**, do one of the following:
 - Accept the default values.
 - Type folder locations that you want to use for **Database folder**, **Log files folder**, and **SYSVOL folder**.
13. Click **Next**.
14. In **Review Options**, review your selections.
15. If you want to export settings to a Windows PowerShell script, click **View script**. The script opens in Notepad, and you can save it to the folder location that you want. Click **Next**. In **Prerequisites Check**, your selections are validated. When the check completes, click **Install**. When prompted by Windows, click **Close**. The server restarts to complete installation of AD DS and DNS.
16. To verify successful installation, view the Server Manager console after the server restarts. Both AD DS and DNS should appear in the left pane, like the highlighted items in the image below.



Create a User Account in Active Directory Users and Computers

You can use this procedure to create a new domain user account in Active Directory Users and Computers Microsoft Management Console (MMC).

Membership in **Domain Admins**, or equivalent, is the minimum required to perform this procedure.

NOTE

To perform this procedure by using Windows PowerShell, open PowerShell and type the following cmdlet on one line, and then press ENTER. You must also replace the user account name in this example with the value that you want to use.

```
New-ADUser -SamAccountName User1 -AccountPassword (read-host "Set user password" -assecurestring) -name "User1" -enabled $true -PasswordNeverExpires $true -ChangePasswordAtLogon $false
```

After you press ENTER, type the password for the user account. The account is created and, by default, is granted membership to the Domain Users group.

With the following cmdlet, you can assign additional group memberships for the new user account. The example below adds User1 to the Domain Admins and Enterprise Admins groups. Ensure before running this command that you change the user account name, domain name, and groups to match your requirements.

```
Add-ADPrincipalGroupMembership -Identity "CN=User1,CN=Users,DC=corp,DC=contoso,DC=com" -MemberOf "CN=Enterprise Admins,CN=Users,DC=corp,DC=contoso,DC=com", "CN=Domain Admins,CN=Users,DC=corp,DC=contoso,DC=com"
```

To create a user account

1. On DC1, in Server Manager, click **Tools**, and then click **Active Directory Users and Computers**. The Active Directory Users and Computers MMC opens. If it is not already selected, click the node for your domain. For example, if your domain is corp.contoso.com, click **corp.contoso.com**.
2. In the details pane, right-click the folder in which you want to add a user account.

Where?

- Active Directory Users and Computers/*domain node/folder*
3. Point to **New**, and then click **User**. The **New Object - User** dialog box opens.
 4. In **First name**, type the user's first name.
 5. In **Initials**, type the user's initials.
 6. In **Last name**, type the user's last name.
 7. Modify **Full name** to add initials or reverse the order of first and last names.
 8. In **User logon name**, type the user logon name. Click **Next**.
 9. In **New Object - User**, in **Password** and **Confirm password**, type the user's password, and then select the appropriate password options.

10. Click **Next**, review the new user account settings, and then click **Finish**.

Assign Group Membership

You can use this procedure to add a user, computer, or group to a group in Active Directory Users and Computers Microsoft Management Console (MMC).

Membership in **Domain Admins**, or equivalent is the minimum required to perform this procedure.

To assign group membership

1. On DC1, in Server Manager, click **Tools**, and then click **Active Directory Users and Computers**. The Active Directory Users and Computers MMC opens. If it is not already selected, click the node for your domain. For example, if your domain is corp.contoso.com, click **corp.contoso.com**.
2. In the details pane, double-click the folder that contains the group to which you want to add a member.

Where?

- **Active Directory Users and Computers**/domain node/folder that contains the group
3. In the details pane, right-click the object that you want to add to a group, such as a user or computer, and then click **Properties**. The object's **Properties** dialog box opens. Click the **Member of** tab.
 4. On the **Member of** tab, click **Add**.
 5. In **Enter the object names to select**, type the name of the group to which you want to add the object, and then click **OK**.
 6. To assign group membership to other users, groups or computers, repeat steps 4 and 5 of this procedure.

Configure a DNS Reverse Lookup Zone

You can use this procedure to configure a reverse lookup zone in Domain Name System (DNS).

Membership in **Domain Admins** is the minimum required to perform this procedure.

NOTE

- For medium and large organizations, it's recommended that you configure and use the DNSAdmins group in Active Directory Users and Computers. For more information, see [Additional Technical Resources](#)
- To perform this procedure by using Windows PowerShell, open PowerShell and type the following cmdlet on one line, and then press ENTER. You must also replace the DNS reverse lookup zone and zonefile names in this example with the values that you want to use. Ensure that you reverse the network ID for the reverse zone name. For example, if the network ID is 192.168.0, create the reverse lookup zone name **0.168.192.in-addr.arpa**.

```
Add-DnsServerPrimaryZone 0.0.10.in-addr.arpa -ZoneFile 0.0.10.in-addr.arpa.dns
```

To configure a DNS reverse lookup zone

1. On DC1, in Server Manager, click **Tools**, and then click **DNS**. The DNS MMC opens.
2. In DNS, if it is not already expanded, double-click the server name to expand the tree. For example, if the DNS server name is DC1, double-click **DC1**.
3. Select **Reverse Lookup Zones**, right-click **Reverse Lookup Zones**, and then click **New Zone**. The New Zone Wizard opens.
4. In **Welcome to the New Zone Wizard**, click **Next**.
5. In **Zone Type**, select **Primary zone**.
6. If your DNS server is a writeable domain controller, ensure that **Store the zone in Active Directory** is selected. Click **Next**.
7. In **Active Directory Zone Replication Scope**, select **To all DNS servers running on domain**

controllers in this domain, unless you have a specific reason to choose a different option. Click **Next**.

8. In the first **Reverse Lookup Zone Name** page, select **IPv4 Reverse Lookup Zone**. Click **Next**.
9. In the second **Reverse Lookup Zone Name** page, do one of the following:
 - In **Network ID**, type the network ID of your IP address range. For example, if your IP address range is 10.0.0.1 through 10.0.0.254, type **10.0.0**.
 - In **Reverse lookup zone name**, your IPv4 reverse lookup zone name is automatically added. Click **Next**.
10. In **Dynamic Update**, select the type of dynamic updates that you want to allow. Click **Next**.
11. In **Completing the New Zone Wizard**, review your choices, and then click **Finish**.

Joining Server Computers to the Domain and Logging On

After you have installed Active Directory Domain Services (AD DS) and created one or more user accounts that have permissions to join a computer to the domain, you can join core network servers to the domain and log on to the servers in order to install additional technologies, such as Dynamic Host Configuration Protocol (DHCP).

On all servers that you are deploying, except for the server running AD DS, do the following:

1. Complete the procedures provided in [Configuring All Servers](#).
2. Use the instructions in the following two procedures to join your servers to the domain and to log on to the servers to perform additional deployment tasks:

NOTE

To perform this procedure by using Windows PowerShell, open PowerShell and type the following cmdlet, and then press ENTER. You must also replace the domain name with the name that you want to use.

```
Add-Computer -DomainName corp.contoso.com
```

When you are prompted to do so, type the user name and password for an account that has permission to join a computer to the domain. To restart the computer, type the following command and press ENTER.

```
Restart-Computer
```

To join computers running Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012 to the domain

1. In Server Manager, click **Local Server**. In the details pane, click **WORKGROUP**. The **System Properties** dialog box opens.
2. In the **System Properties** dialog box, click **Change**. The **Computer Name/Domain Changes** dialog box opens.
3. In **Computer Name**, in **Member of**, click **Domain**, and then type the name of the domain that you want to join. For example, if the domain name is corp.contoso.com, type **corp.contoso.com**.
4. Click **OK**. The **Windows Security** dialog box opens.
5. In **Computer Name/Domain Changes**, in **User name**, type the user name, and in **Password**, type the password, and then click **OK**. The **Computer Name/Domain Changes** dialog box opens, welcoming you to the domain. Click **OK**.
6. The **Computer Name/Domain Changes** dialog box displays a message indicating that you must restart the computer to apply the changes. Click **OK**.
7. On the **System Properties** dialog box, on the **Computer Name** tab, click **Close**. The **Microsoft Windows** dialog box opens, and displays a message, again indicating that you must restart the computer to apply the

changes. Click **Restart Now**.

NOTE

For information on how to join computers that are running other Microsoft operating systems to the domain, see [Appendix C - Joining computers to the domain](#).

To log on to the domain using computers running Windows Server 2016

1. Log off the computer, or restart the computer.
2. Press CTRL + ALT + DELETE. The logon screen appears.
3. In the lower left corner, click **Other User**.
4. In **User name**, type your user name.
5. In **Password**, type your domain password, and then click the arrow, or press ENTER.

NOTE

For information on how to log on to the domain using computers that are running other Microsoft operating systems, see [Appendix D - Log on to the domain](#).

Deploying DHCP1

Before deploying this component of the core network, you must do the following:

- Perform the steps in the section [Configuring All Servers](#).
- Perform the steps in the section [Joining Server Computers to the Domain and Logging On](#).

To deploy DHCP1, which is the computer running the Dynamic Host Configuration Protocol (DHCP) server role, you must complete these steps in the following order:

- [Install Dynamic Host Configuration Protocol \(DHCP\)](#)
- [Create and Activate a New DHCP Scope](#)

NOTE

To perform these procedures by using Windows PowerShell, open PowerShell and type the following cmdlets on separate lines, and then press ENTER. You must also replace the scope name, IP address start and end ranges, subnet mask, and other values in this example with the values that you want to use.

```
Install-WindowsFeature DHCP -IncludeManagementTools

Add-DhcpServerv4Scope -name "Corpnet" -StartRange 10.0.0.1 -EndRange 10.0.0.254 -SubnetMask 255.255.255.0 -State Active

Add-DhcpServerv4ExclusionRange -ScopeID 10.0.0.0 -StartRange 10.0.0.1 -EndRange 10.0.0.15

Set-DhcpServerv4OptionValue -OptionID 3 -Value 10.0.0.1 -ScopeID 10.0.0.0 -ComputerName DHCP1.corp.contoso.com

Add-DhcpServerv4Scope -name "Corpnet2" -StartRange 10.0.1.1 -EndRange 10.0.1.254 -SubnetMask 255.255.255.0 -State Active

Add-DhcpServerv4ExclusionRange -ScopeID 10.0.1.0 -StartRange 10.0.1.1 -EndRange 10.0.1.15

Set-DhcpServerv4OptionValue -OptionID 3 -Value 10.0.1.1 -ScopeID 10.0.1.0 -ComputerName DHCP1.corp.contoso.com

Set-DhcpServerv4OptionValue -DnsDomain corp.contoso.com -DnsServer 10.0.0.2

Add-DhcpServerInDC -DnsName DHCP1.corp.contoso.com
```

Install Dynamic Host Configuration Protocol (DHCP)

You can use this procedure to install and configure the DHCP Server role using the Add Roles and Features Wizard.

Membership in **Domain Admins**, or equivalent, is the minimum required to perform this procedure.

To install DHCP

1. On DHCP1, in Server Manager, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features Wizard opens.
2. In **Before You Begin**, click **Next**.

NOTE

The **Before You Begin** page of the Add Roles and Features Wizard is not displayed if you have previously selected **Skip this page by default** when the Add Roles and Features Wizard was run.

3. In **Select Installation Type**, ensure that **Role-Based or feature-based installation** is selected, and then click **Next**.
4. In **Select destination server**, ensure that **Select a server from the server pool** is selected. In **Server Pool**, ensure that the local computer is selected. Click **Next**.
5. In **Select Server Roles**, in **Roles**, select **DHCP Server**. In **Add features that are required for DHCP Server**, click **Add Features**. Click **Next**.
6. In **Select features**, click **Next**, and in **DHCP Server**, review the information that is provided, and then click **Next**.
7. In **Confirm installation selections**, click **Restart the destination server automatically if required**. When you are prompted to confirm this selection, click **Yes**, and then click **Install**. The **Installation progress** page displays status during the installation process. When the process completes, the message "Configuration required. Installation succeeded on *ComputerName*" is displayed, where *ComputerName* is the name of the computer upon which you installed DHCP Server. In the message window, click **Complete DHCP configuration**. The DHCP Post-Install configuration wizard opens. Click **Next**.

8. In **Authorization**, specify the credentials that you want to use to authorize the DHCP server in Active Directory Domain Services, and then click **Commit**. After authorization is complete, click **Close**.

Create and Activate a New DHCP Scope

You can use this procedure to create a new DHCP scope using the DHCP Microsoft Management Console (MMC). When you complete the procedure, the scope is activated and the exclusion range that you create prevents the DHCP server from leasing the IP addresses that you use to statically configure your servers and other devices that require a static IP address.

Membership in **DHCP Administrators**, or equivalent, is the minimum required to perform this procedure.

To create and activate a new DHCP Scope

1. On DHCP1, in Server Manager, click **Tools**, and then click **DHCP**. The DHCP MMC opens.
2. In **DHCP**, expand the server name. For example, if the DHCP server name is DHCP1.corp.contoso.com, click the down arrow next to **DHCP1.corp.contoso.com**.
3. Beneath the server name, right-click **IPv4**, and then click **New Scope**. The New Scope Wizard opens.
4. In **Welcome to the New Scope Wizard**, click **Next**.
5. In **Scope Name**, in **Name**, type a name for the scope. For example, type **Subnet 1**.
6. In **Description**, type a description for the new scope, and then click **Next**.
7. In **IP Address Range**, do the following:
 - a. In **Start IP address**, type the IP address that is the first IP address in the range. For example, type **10.0.0.1**.
 - b. In **End IP address**, type the IP address that is the last IP address in the range. For example, type **10.0.0.254**. Values for **Length** and **Subnet mask** are entered automatically, based on the IP address you entered for **Start IP address**.
 - c. If necessary, modify the values in **Length** or **Subnet mask**, as appropriate for your addressing scheme.
 - d. Click **Next**.
8. In **Add Exclusions**, do the following:
 - a. In **Start IP address**, type the IP address that is the first IP address in the exclusion range. For example, type **10.0.0.1**.
 - b. In **End IP address**, type the IP address that is the last IP address in the exclusion range. For example, type **10.0.0.15**.
9. Click **Add**, and then click **Next**.
10. In **Lease Duration**, modify the default values for **Days**, **Hours**, and **Minutes**, as appropriate for your network, and then click **Next**.
11. In **Configure DHCP Options**, select **Yes, I want to configure these options now**, and then click **Next**.
12. In **Router (Default Gateway)**, do one of the following:
 - If you do not have routers on your network, click **Next**.
 - In **IP address**, type the IP address of your router or default gateway. For example, type **10.0.0.1**. Click **Add**, and then click **Next**.
13. In **Domain Name and DNS Servers**, do the following:

- a. In **Parent domain**, type the name of the DNS domain that clients use for name resolution. For example, type **corp.contoso.com**.
 - b. In **Server name**, type the name of the DNS computer that clients use for name resolution. For example, type **DC1**.
 - c. Click **Resolve**. The IP address of the DNS server is added in **IP address**. Click **Add**, wait for DNS server IP address validation to complete, and then click **Next**.
14. In **WINS Servers**, because you do not have WINS servers on your network, click **Next**.
15. In **Activate Scope**, select **Yes, I want to activate this scope now**.
16. Click **Next**, and then click **Finish**.

IMPORTANT

To create new scopes for additional subnets, repeat this procedure. Use a different IP address range for each subnet that you plan to deploy, and ensure that DHCP message forwarding is enabled on all routers that lead to other subnets.

Joining Client Computers to the Domain and Logging On

NOTE

To perform this procedure by using Windows PowerShell, open PowerShell and type the following cmdlet, and then press ENTER. You must also replace the domain name with the name that you want to use.

```
Add-Computer -DomainName corp.contoso.com
```

When you are prompted to do so, type the user name and password for an account that has permission to join a computer to the domain. To restart the computer, type the following command and press ENTER.

```
Restart-Computer
```

To join computers running Windows 10 to the domain

1. Log on to the computer with the local Administrator account.
2. In **Search the web and Windows**, type **System**. In search results, click **System (Control panel)**. The **System** dialog box opens.
3. In **System**, click **Advanced system settings**. The **System Properties** dialog box opens. Click the **Computer Name** tab.
4. In **Computer Name**, click **Change**. The **Computer Name/Domain Changes** dialog box opens.
5. In **Computer Name/Domain Changes**, in **Member of**, click **Domain**, and then type the name of the domain you want to join. For example, if the domain name is corp.contoso.com, type **corp.contoso.com**.
6. Click **OK**. The **Windows Security** dialog box opens.
7. In **Computer Name/Domain Changes**, in **User name**, type the user name, and in **Password**, type the password, and then click **OK**. The **Computer Name/Domain Changes** dialog box opens, welcoming you to the domain. Click **OK**.
8. The **Computer Name/Domain Changes** dialog box displays a message indicating that you must restart the computer to apply the changes. Click **OK**.
9. On the **System Properties** dialog box, on the **Computer Name** tab, click **Close**. The **Microsoft Windows** dialog box opens, and displays a message, again indicating that you must restart the computer to apply the changes. Click **Restart Now**.

To join computers running Windows 8.1 to the domain

1. Log on to the computer with the local Administrator account.
2. Right-click **Start**, and then click **System**. The **System** dialog box opens.
3. In **System**, click **Advanced system settings**. The **System Properties** dialog box opens. Click the **Computer Name** tab.
4. In **Computer Name**, click **Change**. The **Computer Name/Domain Changes** dialog box opens.
5. In **Computer Name/Domain Changes**, In **Member of**, click **Domain**, and then type the name of the domain you want to join. For example, if the domain name is corp.contoso.com, type **corp.contoso.com**.
6. Click **OK**. The **Windows Security** dialog box opens.
7. In **Computer Name/Domain Changes**, in **User name**, type the user name, and in **Password**, type the password, and then click **OK**. The **Computer Name/Domain Changes** dialog box opens, welcoming you to the domain. Click **OK**.
8. The **Computer Name/Domain Changes** dialog box displays a message indicating that you must restart the computer to apply the changes. Click **OK**.
9. On the **System Properties** dialog box, on the **Computer Name** tab, click **Close**. The **Microsoft Windows** dialog box opens, and displays a message, again indicating that you must restart the computer to apply the changes. Click **Restart Now**.

To log on to the domain using computers running Windows 10

1. Log off the computer, or restart the computer.
2. Press CTRL + ALT + DELETE. The logon screen appears.
3. In the lower left, click **Other User**.
4. In **User name**, type your domain and user name in the format *domain\user*. For example, to log on to the domain corp.contoso.com with an account named **User-01**, type **CORP\User-01**.
5. In **Password**, type your domain password, and then click the arrow, or press ENTER.

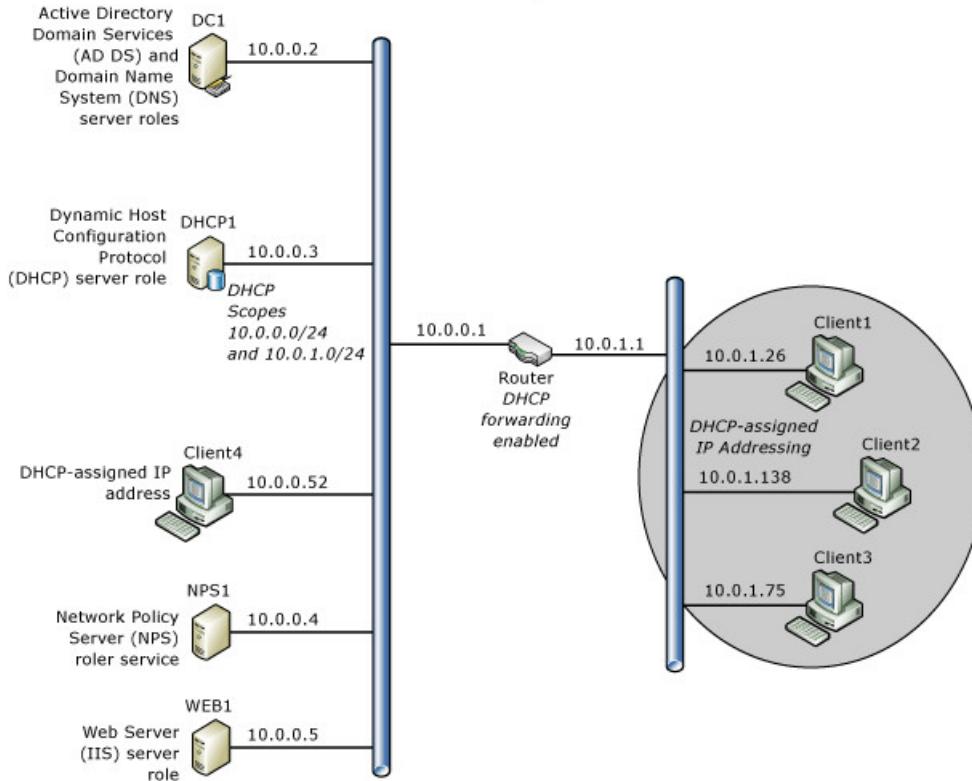
Deploying optional features for network access authentication and Web services

If you intend to deploy network access servers, such as wireless access points or VPN servers, after installing your core network, it is recommended that you deploy both an NPS and a Web server. For network access deployments, the use of secure certificate-based authentication methods is recommended. You can use NPS to manage network access policies and to deploy secure authentication methods. You can use a Web server to publish the certificate revocation list (CRL) of your certification authority (CA) that provides certificates for secure authentication.

NOTE

You can deploy server certificates and other additional features by using Core Network Companion Guides. For more information, see [Additional Technical Resources](#).

The following illustration shows the Windows Server Core Network topology with added NPS and Web servers.



The following sections provide information on adding NPS and Web servers to your network.

- [Deploying NPS1](#)
- [Deploying WEB1](#)

Deploying NPS1

The Network Policy Server (NPS) server is installed as a preparatory step for deploying other network access technologies, such as virtual private network (VPN) servers, wireless access points, and 802.1X authenticating switches.

Network Policy Server (NPS) allows you to centrally configure and manage network policies with the following features: Remote Authentication Dial-In User Service (RADIUS) server and RADIUS proxy.

NPS is an optional component of a core network, but you should install NPS if any of the following are true:

- You are planning to expand your network to include remote access servers that are compatible with the RADIUS protocol, such as a computer running Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2 or Windows Server 2008 and Routing and Remote Access service, Terminal Services Gateway, or Remote Desktop Gateway.
- You plan to deploy 802.1X authentication for wired or wireless access.

Before deploying this role service, you must perform the following steps on the computer you are configuring as an NPS.

- Perform the steps in the section [Configuring All Servers](#).
- Perform the steps in the section [Joining Server Computers to the Domain and Logging On](#)

To deploy NPS1, which is the computer running the Network Policy Server (NPS) role service of the Network Policy and Access Services server role, you must complete this step:

- [Planning the deployment of NPS1](#)
- [Install Network Policy Server \(NPS\)](#)

- Register the NPS in the Default Domain

NOTE

This guide provides instructions for deploying NPS on a standalone server or VM named NPS1. Another recommended deployment model is the installation of NPS on a domain controller. If you prefer installing NPS on a domain controller instead of on a standalone server, install NPS on DC1.

Planning the deployment of NPS1

If you intend to deploy network access servers, such as wireless access points or VPN servers, after deploying your core network, it is recommended that you deploy NPS.

When you use NPS as a Remote Authentication Dial-In User Service (RADIUS) server, NPS performs authentication and authorization for connection requests through your network access servers. NPS also allows you to centrally configure and manage network policies that determine who can access the network, how they can access the network, and when they can access the network.

Following are key planning steps before installing NPS.

- Plan the user accounts database. By default, if you join the server running NPS to an Active Directory domain, NPS performs authentication and authorization using the AD DS user accounts database. In some cases, such as with large networks that use NPS as a RADIUS proxy to forward connection requests to other RADIUS servers, you might want to install NPS on a non-domain member computer.
- Plan RADIUS accounting. NPS allows you to log accounting data to a SQL Server database or to a text file on the local computer. If you want to use SQL Server logging, plan the installation and configuration of your server running SQL Server.

Install Network Policy Server (NPS)

You can use this procedure to install Network Policy Server (NPS) by using the Add Roles and Features Wizard. NPS is a role service of the Network Policy and Access Services server role.

NOTE

By default, NPS listens for RADIUS traffic on ports 1812, 1813, 1645, and 1646 on all installed network adapters. If Windows Firewall with Advanced Security is enabled when you install NPS, firewall exceptions for these ports are automatically created during the installation process for both Internet Protocol version 6 (IPv6) and IPv4 traffic. If your network access servers are configured to send RADIUS traffic over ports other than these defaults, remove the exceptions created in Windows Firewall with Advanced Security during NPS installation, and create exceptions for the ports that you do use for RADIUS traffic.

Administrative Credentials

To complete this procedure, you must be a member of the **Domain Admins** group.

NOTE

To perform this procedure by using Windows PowerShell, open PowerShell and type the following, and then press ENTER.

```
Install-WindowsFeature NPAS -IncludeManagementTools
```

To install NPS

1. On NPS1, in Server Manager, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features Wizard opens.
2. In **Before You Begin**, click **Next**.

NOTE

The **Before You Begin** page of the Add Roles and Features Wizard is not displayed if you have previously selected **Skip this page by default** when the Add Roles and Features Wizard was run.

3. In **Select Installation Type**, ensure that **Role-Based or feature-based installation** is selected, and then click **Next**.
4. In **Select destination server**, ensure that **Select a server from the server pool** is selected. In **Server Pool**, ensure that the local computer is selected. Click **Next**.
5. In **Select Server Roles**, in **Roles**, select **Network Policy and Access Services**. A dialog box opens asking if it should add features that are required for Network Policy and Access Services. Click **Add Features**, and then click **Next**.
6. In **Select features**, click **Next**, and in **Network Policy and Access Services**, review the information that is provided, and then click **Next**.
7. In **Select role services**, click **Network Policy Server**. In **Add features that are required for Network Policy Server**, click **Add Features**. Click **Next**.
8. In **Confirm installation selections**, click **Restart the destination server automatically if required**. When you are prompted to confirm this selection, click **Yes**, and then click **Install**. The Installation progress page displays status during the installation process. When the process completes, the message "Installation succeeded on *ComputerName*" is displayed, where *ComputerName* is the name of the computer upon which you installed Network Policy Server. Click **Close**.

Register the NPS in the Default Domain

You can use this procedure to register an NPS in the domain where the server is a domain member.

NPSs must be registered in Active Directory so that they have permission to read the dial-in properties of user accounts during the authorization process. Registering an NPS adds the server to the **RAS and IAS Servers** group in Active Directory.

Administrative credentials

To complete this procedure, you must be a member of the **Domain Admins** group.

NOTE

To perform this procedure by using network shell (Netsh) commands within Windows PowerShell, open PowerShell and type the following, and then press ENTER.

```
netsh nps add registeredserver domain=corp.contoso.com server=NPS1.corp.contoso.com
```

To register an NPS in its default domain

1. On NPS1, in Server Manager, click Tools, and then click **Network Policy Server**. The Network Policy Server MMC opens.
2. Right-click **NPS (Local)**, and then click **Register server in Active Directory**. The **Network Policy Server** dialog box opens.
3. In **Network Policy Server**, click **OK**, and then click **OK** again.

For more information about Network Policy Server, see [Network Policy Server \(NPS\)](#).

Deploying WEB1

The Web Server (IIS) role in Windows Server 2016 provides a secure, easy-to-manage, modular and extensible

platform for reliably hosting web sites, services, and applications. With Internet Information Services (IIS), you can share information with users on the Internet, an intranet, or an extranet. IIS is a unified web platform that integrates IIS, ASP.NET, FTP services, PHP, and Windows Communication Foundation (WCF).

In addition to allowing you to publish a CRL for access by domain member computers, the Web Server (IIS) server role allows you to set up and manage multiple web sites, web applications, and FTP sites. IIS also provides the following benefits:

- Maximize web security through a reduced server footprint and automatic application isolation.
- Easily deploy and run ASP.NET, classic ASP, and PHP web applications on the same server.
- Achieve application isolation by giving worker processes a unique identity and sandboxed configuration by default, further reducing security risks.
- Easily add, remove, and even replace built-in IIS components with custom modules, suited for customer needs.
- Speed up your website through built-in dynamic caching and enhanced compression.

To deploy WEB1, which is the computer that is running the Web Server (IIS) server role, you must do the following:

- Perform the steps in the section [Configuring All Servers](#).
- Perform the steps in the section [Joining Server Computers to the Domain and Logging On](#)
- [Install the Web Server \(IIS\) server role](#)

Install the Web Server (IIS) server role

To complete this procedure, you must be a member of the **Administrators** group.

NOTE

To perform this procedure by using Windows PowerShell, open PowerShell and type the following, and then press ENTER.

```
Install-WindowsFeature Web-Server -IncludeManagementTools
```

1. In **Server Manager**, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features Wizard opens.
2. In **Before You Begin**, click **Next**.

NOTE

The **Before You Begin** page of the Add Roles and Features Wizard is not displayed if you have previously selected **Skip this page by default** when the Add Roles and Features Wizard was run.

3. On the **Select Installation Type** page, click **Next**.
4. On the **Select destination server** page, ensure that the local computer is selected, and then click **Next**.
5. On the **Select server roles** page, scroll to and select **Web Server (IIS)**. The **Add features that are required for Web Server (IIS)** dialog box opens. Click **Add Features**, and then click **Next**.
6. Click **Next** until you have accepted all of the default web server settings, and then click **Install**.
7. Verify that all installations were successful, and then click **Close**.

Additional Technical Resources

For more information about the technologies in this guide, see the following resources:

Windows Server 2016, Windows Server 2012 R2 , and Windows Server 2012 Technical Library Resources

- [What's new in Active Directory Domain Services \(AD DS\) in Windows Server 2016](#)
- [Active Directory Domain Services overview](#) at <https://technet.microsoft.com/library/hh831484.aspx>.
- [Domain Name System \(DNS\) overview](#) at <https://technet.microsoft.com/library/hh831667.aspx>.
- [Implementing the DNS Admins Role](#)
- [Dynamic Host Configuration Protocol \(DHCP\) overview](#) at <https://technet.microsoft.com/library/hh831825.aspx>.
- [Network Policy and Access Services overview](#) at <https://technet.microsoft.com/library/hh831683.aspx>.
- [Web Server \(IIS\) overview](#) at <https://technet.microsoft.com/library/hh831725.aspx>.

Appendices A through E

The following sections contain additional configuration information for computers that are running operating systems other than Windows Server 2016, Windows 10, Windows Server 2012 , and Windows 8. In addition, a network preparation worksheet is provided to assist you with your deployment.

1. [Appendix A - Renaming computers](#)
2. [Appendix B - Configuring static IP addresses](#)
3. [Appendix C - Joining computers to the domain](#)
4. [Appendix D - Log on to the domain](#)
5. [Appendix E - Core Network Planning Preparation Sheet](#)

Appendix A - Renaming computers

You can use the procedures in this section to provide computers running Windows Server 2008 R2, Windows 7, Windows Server 2008 , and Windows Vista with a different computer name.

- [Windows Server 2008 R2 and Windows 7](#)
- [Windows Server 2008 and Windows Vista](#)

Windows Server 2008 R2 and Windows 7

Membership in **Administrators**, or equivalent, is the minimum required to perform these procedures.

To rename computers running Windows Server 2008 R2 and Windows 7

1. Click **Start**, right-click **Computer**, and then click **Properties**. The **System** dialog box opens.
2. In **Computer name, domain, and workgroup settings**, click **Change settings**. The **System Properties** dialog box opens.

NOTE

On computers running Windows 7, before the **System Properties** dialog box opens, the **User Account Control** dialog box opens, requesting permission to continue. Click **Continue** to proceed.

3. Click **Change**. The **Computer Name/Domain Changes** dialog box opens.
4. In **Computer Name**, type the name for your computer. For example, if you want to name the computer

DC1, type **DC1**.

5. Click **OK** twice, click **Close**, and then click **Restart Now** to restart the computer.

Windows Server 2008 and Windows Vista

Membership in **Administrators**, or equivalent, is the minimum required to perform these procedures.

To rename computers running Windows Server 2008 and Windows Vista

1. Click **Start**, right-click **Computer**, and then click **Properties**. The **System** dialog box opens.
2. In **Computer name, domain, and workgroup settings**, click **Change settings**. The **System Properties** dialog box opens.

NOTE

On computers running Windows Vista, before the **System Properties** dialog box opens, the **User Account Control** dialog box opens, requesting permission to continue. Click **Continue** to proceed.

3. Click **Change**. The **Computer Name/Domain Changes** dialog box opens.
4. In **Computer Name**, type the name for your computer. For example, if you want to name the computer DC1, type **DC1**.
5. Click **OK** twice, click **Close**, and then click **Restart Now** to restart the computer.

Appendix B - Configuring static IP addresses

This topic provides procedures for configuring static IP addresses on computers running the following operating systems:

- [Windows Server 2008 R2](#)
- [Windows Server 2008](#)

Windows Server 2008 R2

Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To configure a static IP address on a computer running Windows Server 2008 R2

1. Click **Start**, and then click **Control Panel**.
2. In **Control Panel**, click **Network and Internet**. **Network and Internet** opens.
In **Network and Internet**, click **Network and Sharing Center**. **Network and Sharing Center** opens.
3. In **Network and Sharing Center**, click **Change adapter settings**. **Network Connections** opens.
4. In **Network Connections**, right-click the network connection that you want to configure, and then click **Properties**.
5. In **Local Area Connection Properties**, in **This connection uses the following items**, select **Internet Protocol Version 4 (TCP/IPv4)**, and then click **Properties**. The **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box opens.
6. In **Internet Protocol Version 4 (TCP/IPv4) Properties**, on the **General** tab, click **Use the following IP address**. In **IP address**, type the IP address that you want to use.
7. Press tab to place the cursor in **Subnet mask**. A default value for subnet mask is entered automatically. Either accept the default subnet mask, or type the subnet mask that you want to use.
8. In **Default gateway**, type the IP address of your default gateway.

9. In **Preferred DNS server**, type the IP address of your DNS server. If you plan to use the local computer as the preferred DNS server, type the IP address of the local computer.
10. In **Alternate DNS Server**, type the IP address of your alternate DNS server, if any. If you plan to use the local computer as an alternate DNS server, type the IP address of the local computer.
11. Click **OK**, and then click **Close**.

Windows Server 2008

Membership in **Administrators**, or equivalent, is the minimum required to perform these procedures.

To configure a static IP address on a computer running Windows Server 2008

1. Click **Start**, and then click **Control Panel**.
2. In **Control Panel**, verify that **Classic View** is selected, and then double-click **Network and Sharing Center**.
3. In **Network and Sharing Center**, in **Tasks**, click **Manage Network Connections**.
4. In **Network Connections**, right-click the network connection that you want to configure, and then click **Properties**.
5. In **Local Area Connection Properties**, in **This connection uses the following items**, select **Internet Protocol Version 4 (TCP/IPv4)**, and then click **Properties**. The **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box opens.
6. In **Internet Protocol Version 4 (TCP/IPv4) Properties**, on the **General** tab, click **Use the following IP address**. In **IP address**, type the IP address that you want to use.
7. Press tab to place the cursor in **Subnet mask**. A default value for subnet mask is entered automatically. Either accept the default subnet mask, or type the subnet mask that you want to use.
8. In **Default gateway**, type the IP address of your default gateway.
9. In **Preferred DNS server**, type the IP address of your DNS server. If you plan to use the local computer as the preferred DNS server, type the IP address of the local computer.
10. In **Alternate DNS Server**, type the IP address of your alternate DNS server, if any. If you plan to use the local computer as an alternate DNS server, type the IP address of the local computer.
11. Click **OK**, and then click **Close**.

Appendix C - Joining computers to the domain

You can use these procedures to join computers running Windows Server 2008 R2, Windows 7, Windows Server 2008 , and Windows Vista to the domain.

- [Windows Server 2008 R2 and Windows 7](#)
- [Windows Server 2008 and Windows Vista](#)

IMPORTANT

To join a computer to a domain, you must be logged on to the computer with the local Administrator account or, if you are logged on to the computer with a user account that does not have local computer administrative credentials, you must provide the credentials for the local Administrator account during the process of joining the computer to the domain. In addition, you must have a user account in the domain to which you want to join the computer. During the process of joining the computer to the domain, you will be prompted for your domain account credentials (user name and password).

Windows Server 2008 R2 and Windows 7

Membership in **Domain Users**, or equivalent, is the minimum required to perform this procedure.

To join computers running Windows Server 2008 R2 and Windows 7 to the domain

1. Log on to the computer with the local Administrator account.
2. Click **Start**, right-click **Computer**, and then click **Properties**. The **System** dialog box opens.
3. In **Computer name, domain, and workgroup settings**, click **Change settings**. The **System Properties** dialog box opens.

NOTE

On computers running Windows 7, before the **System Properties** dialog box opens, the **User Account Control** dialog box opens, requesting permission to continue. Click **Continue** to proceed.

4. Click **Change**. The **Computer Name/Domain Changes** dialog box opens.
5. In **Computer Name**, in **Member of**, select **Domain**, and then type the name of the domain you want to join. For example, if the domain name is corp.contoso.com, type **corp.contoso.com**.
6. Click **OK**. The **Windows Security** dialog box opens.
7. In **Computer Name/Domain Changes**, in **User name**, type the user name, and in **Password**, type the password, and then click **OK**. The **Computer Name/Domain Changes** dialog box opens, welcoming you to the domain. Click **OK**.
8. The **Computer Name/Domain Changes** dialog box displays a message indicating that you must restart the computer to apply the changes. Click **OK**.
9. On the **System Properties** dialog box, on the **Computer Name** tab, click **Close**. The **Microsoft Windows** dialog box opens, and displays a message, again indicating that you must restart the computer to apply the changes. Click **Restart Now**.

Windows Server 2008 and Windows Vista

Membership in **Domain Users**, or equivalent, is the minimum required to perform this procedure.

To join computers running Windows Server 2008 and Windows Vista to the domain

1. Log on to the computer with the local Administrator account.
2. Click **Start**, right-click **Computer**, and then click **Properties**. The **System** dialog box opens.
3. In **Computer name, domain, and workgroup settings**, click **Change settings**. The **System Properties** dialog box opens.
4. Click **Change**. The **Computer Name/Domain Changes** dialog box opens.
5. In **Computer Name**, in **Member of**, select **Domain**, and then type the name of the domain you want to join. For example, if the domain name is corp.contoso.com, type **corp.contoso.com**.
6. Click **OK**. The **Windows Security** dialog box opens.
7. In **Computer Name/Domain Changes**, in **User name**, type the user name, and in **Password**, type the password, and then click **OK**. The **Computer Name/Domain Changes** dialog box opens, welcoming you to the domain. Click **OK**.
8. The **Computer Name/Domain Changes** dialog box displays a message indicating that you must restart the computer to apply the changes. Click **OK**.
9. On the **System Properties** dialog box, on the **Computer Name** tab, click **Close**. The **Microsoft Windows**

dialog box opens, and displays a message, again indicating that you must restart the computer to apply the changes. Click **Restart Now**.

Appendix D - Log on to the domain

You can use these procedures to log on to the domain using computers running Windows Server 2008 R2, Windows 7, Windows Server 2008 , and Windows Vista.

- [Windows Server 2008 R2 and Windows 7](#)
- [Windows Server 2008 and Windows Vista](#)

Windows Server 2008 R2 and Windows 7

Membership in **Domain Users**, or equivalent, is the minimum required to perform this procedure.

Log on to the domain using computers running Windows Server 2008 R2 and Windows 7

1. Log off the computer, or restart the computer.
2. Press CTRL + ALT + DELETE. The logon screen appears.
3. Click **Switch User**, and then click **Other User**.
4. In **User name**, type your domain and user name in the format *domain\user*. For example, to log on to the domain corp.contoso.com with an account named **User-01**, type **CORP\User-01**.
5. In **Password**, type your domain password, and then click the arrow, or press ENTER.

Windows Server 2008 and Windows Vista

Membership in **Domain Users**, or equivalent, is the minimum required to perform this procedure.

Log on to the domain using computers running Windows Server 2008 and Windows Vista

1. Log off the computer, or restart the computer.
2. Press CTRL + ALT + DELETE. The logon screen appears.
3. Click **Switch User**, and then click **Other User**.
4. In **User name**, type your domain and user name in the format *domain\user*. For example, to log on to the domain corp.contoso.com with an account named **User-01**, type **CORP\User-01**.
5. In **Password**, type your domain password, and then click the arrow, or press ENTER.

Appendix E - Core Network Planning Preparation Sheet

You can use this Network Planning Preparation Sheet to gather the information required to install a core network. This topic provides tables that contain the individual configuration items for each server computer for which you must supply information or specific values during the installation or configuration process. Example values are provided for each configuration item.

For planning and tracking purposes, spaces are provided in each table for you to enter the values used for your deployment. If you log security-related values in these tables, you should store the information in a secure location.

The following links lead to the sections in this topic that provide configuration items and example values that are associated with the deployment procedures presented in this guide.

1. [Installing Active Directory Domain Services and DNS](#)
 - [Configuring a DNS Reverse Lookup Zone](#)
2. [Installing DHCP](#)
 - [Creating an exclusion range in DHCP](#)

- [Creating a new DHCP scope](#)

3. [Installing Network Policy Server \(optional\)](#)

Installing Active Directory Domain Services and DNS

The tables in this section list configuration items for pre-installation and installation of Active Directory Domain Services (AD DS) and DNS.

Pre-installation configuration items for AD DS and DNS

The following tables list pre-installation configuration items as described in [Configuring All Servers](#):

- [Configure a Static IP Address](#)

CONFIGURATION ITEMS	EXAMPLE VALUES	VALUES
IP address	10.0.0.2	
Subnet mask	255.255.255.0	
Default gateway	10.0.0.1	
Preferred DNS server	127.0.0.1	
Alternate DNS server	10.0.0.15	

- [Rename the Computer](#)

CONFIGURATION ITEM	EXAMPLE VALUE	VALUE
Computer name	DC1	

AD DS and DNS installation configuration items

Configuration items for the Windows Server Core Network deployment procedure [Install AD DS and DNS for a New Forest](#):

CONFIGURATION ITEMS	EXAMPLE VALUES	VALUES
Full DNS name	corp.contoso.com	
Forest functional level	Windows Server 2003	
Active Directory Domain Services database folder location	E:\Configuration\ Or accept the default location.	
Active Directory Domain Services log files folder location	E:\Configuration\ Or accept the default location.	
Active Directory Domain Services SYSVOL folder location	E:\Configuration\ Or accept the default location	
Directory Restore Mode Administrator password	J*p2leO4\$F	

CONFIGURATION ITEMS	EXAMPLE VALUES	VALUES
Answer file name (optional)	AD DS_AnswerFile	

Configuring a DNS Reverse Lookup Zone

CONFIGURATION ITEMS	EXAMPLE VALUES	VALUES
Zone type:	- Primary zone - Secondary zone - Stub zone	
Zone type	- Selected - Not selected	
Store the zone in Active Directory		
Active Directory zone replication scope	- To all DNS servers in this forest - To all DNS servers in this domain - To all domain controllers in this domain - To all domain controllers specified in the scope of this directory partition	
Reverse lookup zone name (IP type)	- IPv4 Reverse Lookup Zone - IPv6 Reverse Lookup Zone	
Reverse lookup zone name (network ID)	10.0.0	

Installing DHCP

The tables in this section list configuration items for pre-installation and installation of DHCP.

Pre-installation configuration items for DHCP

The following tables list pre-installation configuration items as described in [Configuring All Servers](#):

- [Configure a Static IP Address](#)

CONFIGURATION ITEMS	EXAMPLE VALUES	VALUES
IP address	10.0.0.3	
Subnet mask	255.255.255.0	
Default gateway	10.0.0.1	
Preferred DNS server	10.0.0.2	
Alternate DNS server	10.0.0.15	

- [Rename the Computer](#)

CONFIGURATION ITEM	EXAMPLE VALUE	VALUE
Computer name	DHCP1	

DHCP installation configuration items

Configuration items for the Windows Server Core Network deployment procedure [Install Dynamic Host Configuration Protocol \(DHCP\)](#):

CONFIGURATION ITEMS	EXAMPLE VALUES	VALUES
Network connect bindings	Ethernet	
DNS server settings	DC1	
Preferred DNS server IP address	10.0.0.2	
Alternate DNS server IP address	10.0.0.15	
Scope name	Corp1	
Starting IP address	10.0.0.1	
Ending IP address	10.0.0.254	
Subnet mask	255.255.255.0	
Default gateway (optional)	10.0.0.1	
Lease duration	8 days	
IPv6 DHCP server operation mode	Not enabled	

Creating an exclusion range in DHCP

Configuration items to create an exclusion range while creating a scope in DHCP.

CONFIGURATION ITEMS	EXAMPLE VALUES	VALUES
Scope name	Corp1	
Scope description	Main office subnet 1	
Exclusion range start IP address	10.0.0.1	
Exclusion range end IP address	10.0.0.15	

Creating a new DHCP scope

Configuration items for the Windows Server Core Network deployment procedure [Create and Activate a New DHCP Scope](#):

CONFIGURATION ITEMS	EXAMPLE VALUES	VALUES
New scope name	Corp2	
Scope description	Main office subnet 2	
(IP address range)	10.0.1.1	
Start IP address		

CONFIGURATION ITEMS	EXAMPLE VALUES	VALUES
(IP address range)	10.0.1.254	
End IP address		
Length	8	
Subnet mask	255.255.255.0	
(Exclusion range) Start IP address	10.0.1.1	
Exclusion range end IP address	10.0.1.15	
Lease duration	- 8 - 0 - 0	
Days		
Hours		
Minutes		
Router (default gateway)	10.0.1.1	
IP address		
DNS parent domain	corp.contoso.com	
DNS server	10.0.0.2	
IP address		

Installing Network Policy Server (optional)

The tables in this section list configuration items for pre-installation and installation of NPS.

Pre-installation configuration items

The following three tables list pre-installation configuration items as described in [Configuring All Servers](#):

- Configure a Static IP Address

CONFIGURATION ITEMS	EXAMPLE VALUES	VALUES
IP address	10.0.0.4	
Subnet mask	255.255.255.0	
Default gateway	10.0.0.1	
Preferred DNS server	10.0.0.2	
Alternate DNS server	10.0.0.15	

- Rename the Computer

CONFIGURATION ITEM	EXAMPLE VALUE	VALUE
Computer name	NPS1	

Network Policy Server installation configuration items

Configuration items for the Windows Server Core Network NPS deployment procedures [Install Network Policy Server \(NPS\)](#) and [Register the NPS in the Default Domain](#).

- No additional configuration items are required to install and register NPS.

Core network companion guidance

9/18/2018 • 3 minutes to read • [Edit Online](#)

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

While the Windows Server 2016 [Core Network Guide](#) provides instructions on how to deploy a new Active Directory® forest with a new root domain and the supporting networking infrastructure, Companion Guides provide you with the ability to add features to your network.

Each companion guide allows you to accomplish a specific goal after you have deployed your core network. In some cases, there are multiple companion guides that, when deployed together and in the correct order, allow you to accomplish very complex goals in a measured, cost-effective, reasonable manner.

If you deployed your Active Directory domain and core network before encountering the Core Network Guide, you can still use the Companion Guides to add features to your network. Simply use the Core Network Guide as a list of prerequisites, and know that to deploy additional features with the Companion Guides, your network must meet the prerequisites that are provided by the Core Network Guide.

Core Network Companion Guide: Deploy Server Certificates for 802.1X Wired and Wireless Deployments

This companion guide explains how to build upon the core network by deploying server certificates for computers that are running Network Policy Server (NPS), Remote Access Service (RAS), or both.

Server certificates are required when you deploy certificate-based authentication methods with Extensible Authentication Protocol (EAP) and Protected EAP (PEAP) for network access authentication. Deploying server certificates with Active Directory Certificate Services (AD CS) for EAP and PEAP certificate-based authentication methods provides the following benefits:

- Binding the identity of the NPS or RAS server to a private key
- A cost-efficient and secure method for automatically enrolling certificates to domain member NPS and RAS servers
- An efficient method for managing certificates and certification authorities
- Security provided by certificate-based authentication
- The ability to expand the use of certificates for additional purposes

For instructions on how to deploy server certificates, see [Deploy Server Certificates for 802.1X Wired and Wireless Deployments](#).

Core Network Companion Guide: Deploy Password-Based 802.1X Authenticated Wireless Access

This companion guide explains how to build upon the core network by providing instructions about how to deploy Institute of Electrical and Electronics Engineers (IEEE) 802.1X-authenticated IEEE 802.11 wireless access using Protected Extensible Authentication Protocol\–Microsoft Challenge Handshake Authentication Protocol version 2 (PEAP-MS-CHAP v2).

The authentication method PEAP-MS-CHAP v2 requires that authenticating servers running Network Policy Server (NPS) present wireless clients with a server certificate to prove the NPS identity to the client, however user authentication is not performed by using a certificate - instead, users provide their domain user name and

password.

Because PEAP-MS-CHAP v2 requires that users provide password-based credentials rather than a certificate during the authentication process, it is typically easier and less expensive to deploy than EAP-TLS or PEAP-TLS.

Before you use this guide to deploy wireless access with the PEAP-MS-CHAP v2 authentication method, you must do the following:

1. Follow the instructions in the Core Network Guide to deploy your core network infrastructure, or already have the technologies presented in that guide deployed on your network.
2. Follow the instructions in the Core Network Companion Guide Deploy Server Certificates for 802.1X Wired and Wireless Deployments, or already have the technologies presented in that guide deployed on your network.

For instructions on how to deploy wireless access with PEAP-MS-CHAP v2, see [Deploy Password-Based 802.1X Authenticated Wireless Access](#).

Core Network Companion Guide: Deploy BranchCache Hosted Cache Mode

This companion guide explains how to deploy BranchCache in Hosted Cache Mode in one or more branch offices.

BranchCache is a wide area network (WAN) bandwidth optimization technology that is included in some editions of the Windows Server 2016 and Windows 10 operating systems, as well as in earlier versions of Windows and Windows Server.

When you deploy BranchCache in hosted cache mode, the content cache at a branch office is hosted on one or more server computers, which are called hosted cache servers. Hosted cache servers can run workloads in addition to hosting the cache, which allows you to use the server for multiple purposes in the branch office.

BranchCache hosted cache mode increases the cache efficiency because content is available even if the client that originally requested and cached the data is offline. Because the hosted cache server is always available, more content is cached, providing greater WAN bandwidth savings, and BranchCache efficiency is improved.

When you deploy hosted cache mode, all clients in a multiple-subnet branch office can access a single cache, which is stored on the hosted cache server, even if the clients are on different subnets.

For instructions on how to deploy BranchCache in Hosted Cache Mode, see [Deploy BranchCache Hosted Cache Mode](#).

Deploy Server Certificates for 802.1X Wired and Wireless Deployments

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

You can use this guide to deploy server certificates to your Remote Access and Network Policy Server (NPS) infrastructure servers.

This guide contains the following sections.

- [Prerequisites for using this guide](#)
- [What this guide does not provide](#)
- [Technology overviews](#)
- [Server Certificate Deployment Overview](#)
- [Server Certificate Deployment Planning](#)
- [Server Certificate Deployment](#)

Digital server certificates

This guide provides instructions for using Active Directory Certificate Services (AD CS) to automatically enroll certificates to Remote Access and NPS infrastructure servers. AD CS allows you to build a public key infrastructure (PKI) and provide public key cryptography, digital certificates, and digital signature capabilities for your organization.

When you use digital server certificates for authentication between computers on your network, the certificates provide:

1. Confidentiality through encryption.
2. Integrity through digital signatures.
3. Authentication by associating certificate keys with computer, user, or device accounts on a computer network.

Server types

By using this guide, you can deploy server certificates to the following types of servers.

- Servers that are running the Remote Access service, that are DirectAccess or standard virtual private network (VPN) servers, and that are members of the **RAS and IAS Servers** group.
- Servers that are running the Network Policy Server (NPS) service that are members of the **RAS and IAS Servers** group.

Advantages of certificate autoenrollment

Automatic enrollment of server certificates, also called autoenrollment, provides the following advantages.

- The AD CS certification authority (CA) automatically enrolls a server certificate to all of your NPS and Remote Access servers.
- All computers in the domain automatically receive your CA certificate, which is installed in the Trusted Root Certification Authorities store on every domain member computer. Because of this, all computers in the domain trust the certificates that are issued by your CA. This trust allows your authentication servers to prove their

identities to each other and engage in secure communications.

- Other than refreshing Group Policy, the manual reconfiguration of every server is not required.
- Every server certificate includes both the Server Authentication purpose and the Client Authentication purpose in Enhanced Key Usage (EKU) extensions.
- Scalability. After deploying your Enterprise Root CA with this guide, you can expand your public key infrastructure (PKI) by adding Enterprise subordinate CAs.
- Manageability. You can manage AD CS by using the AD CS console or by using Windows PowerShell commands and scripts.
- Simplicity. You specify the servers that enroll server certificates by using Active Directory group accounts and group membership.
- When you deploy server certificates, the certificates are based on a template that you configure with the instructions in this guide. This means that you can customize different certificate templates for specific server types, or you can use the same template for all server certificates that you want to issue.

Prerequisites for using this guide

This guide provides instructions on how to deploy server certificates by using AD CS and the Web Server (IIS) server role in Windows Server 2016. Following are the prerequisites for performing the procedures in this guide.

- You must deploy a core network using the Windows Server 2016 Core Network Guide, or you must already have the technologies provided in the Core Network Guide installed and functioning correctly on your network. These technologies include TCP/IP v4, DHCP, Active Directory Domain Services (AD DS), DNS, and NPS.

NOTE

The Windows Server 2016 Core Network Guide is available in the Windows Server 2016 Technical Library. For more information, see [Core Network Guide](#).

- You must read the planning section of this guide to ensure that you are prepared for this deployment before you perform the deployment.
- You must perform the steps in this guide in the order in which they are presented. Do not jump ahead and deploy your CA without performing the steps that lead up to deploying the server, or your deployment will fail.
- You must be prepared to deploy two new servers on your network - one server upon which you will install AD CS as an Enterprise Root CA, and one server upon which you will install Web Server (IIS) so that your CA can publish the certificate revocation list (CRL) to the Web server.

NOTE

You are prepared to assign a static IP address to the Web and AD CS servers that you deploy with this guide, as well as to name the computers according to your organization naming conventions. In addition, you must join the computers to your domain.

What this guide does not provide

This guide does not provide comprehensive instructions for designing and deploying a public key infrastructure (PKI) by using AD CS. It is recommended that you review AD CS documentation and PKI design documentation before deploying the technologies in this guide.

Technology overviews

Following are technology overviews for AD CS and Web Server (IIS).

Active Directory Certificate Services

AD CS in Windows Server 2016 provides customizable services for creating and managing the X.509 certificates that are used in software security systems that employ public key technologies. Organizations can use AD CS to enhance security by binding the identity of a person, device, or service to a corresponding public key. AD CS also includes features that allow you to manage certificate enrollment and revocation in a variety of scalable environments.

For more information, see [Active Directory Certificate Services Overview](#) and [Public Key Infrastructure Design Guidance](#).

Web Server (IIS)

The Web Server (IIS) role in Windows Server 2016 provides a secure, easy-to-manage, modular, and extensible platform for reliably hosting websites, services, and applications. With IIS, you can share information with users on the Internet, an intranet, or an extranet. IIS is a unified web platform that integrates IIS, ASP.NET, FTP services, PHP, and Windows Communication Foundation (WCF).

For more information, see [Web Server \(IIS\) Overview](#).

Server Certificate Deployment Overview

9/1/2018 • 5 minutes to read • [Edit Online](#)

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

This topic contains the following sections.

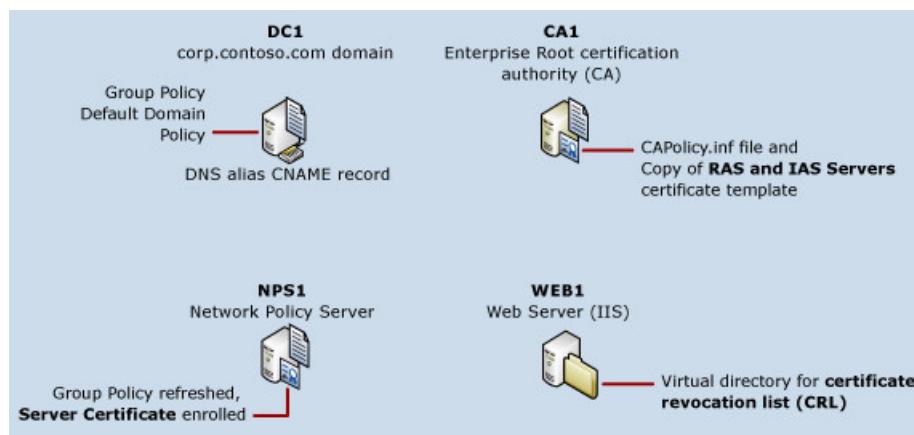
- [Server certificate deployment components](#)
- [Server certificate deployment process overview](#)

Server certificate deployment components

You can use this guide to install Active Directory Certificate Services (AD CS) as an Enterprise root certification authority (CA) and to enroll server certificates to servers that are running Network Policy Server (NPS), Routing and Remote Access service (RRAS), or both NPS and RRAS.

If you deploy SDN with certificate-based authentication, servers are required to use a server certificate to prove their identities to other servers so that they achieve secure communications.

The following illustration shows the components that are required to deploy server certificates to servers in your SDN infrastructure.



NOTE

In the illustration above, multiple servers are depicted: DC1, CA1, WEB1, and many SDN servers. This guide provides instructions for deploying and configuring CA1 and WEB1, and for configuring DC1, which this guide assumes you have already installed on your network. If you have not already installed your Active Directory domain, you can do so by using the [Core Network Guide](#) for Windows Server 2016.

For more information on each item depicted in the illustration above, see the following:

- [CA1](#)
- [WEB1](#)
- [DC1](#)
- [NPS1](#)

CA1 running the AD CS server role

In this scenario, the Enterprise Root certification authority (CA) is also an issuing CA. The CA issues certificates to server computers that have the correct security permissions to enroll a certificate. Active Directory Certificate Services (AD CS) is installed on CA1.

For larger networks or where security concerns provide justification, you can separate the roles of root CA and issuing CA, and deploy subordinate CAs that are issuing CAs.

In the most secure deployments, the Enterprise Root CA is taken offline and physically secured.

CAPolicy.inf

Before you install AD CS, you configure the CAPolicy.inf file with specific settings for your deployment.

Copy of the RAS and IAS servers certificate template

When you deploy server certificates, you make one copy of the **RAS and IAS servers** certificate template and then configure the template according to your requirements and the instructions in this guide.

You utilize a copy of the template rather than the original template so that the configuration of the original template is preserved for possible future use. You configure the copy of the **RAS and IAS servers** template so that the CA can create server certificates that it issues to the groups in Active Directory Users and Computers that you specify.

Additional CA1 configuration

The CA publishes a certificate revocation list (CRL) that computers must check to ensure that certificates that are presented to them as proof of identity are valid certificates and have not been revoked. You must configure your CA with the correct location of the CRL so that computers know where to look for the CRL during the authentication process.

WEB1 running the Web Services (IIS) server role

On the computer that is running the Web Server (IIS) server role, WEB1, you must create a folder in Windows Explorer for use as the location for the CRL and AIA.

Virtual directory for the CRL and AIA

After you create a folder in Windows Explorer, you must configure the folder as a virtual directory in Internet Information Services (IIS) Manager, as well as configuring the access control list for the virtual directory to allow computers to access the AIA and CRL after they are published there.

DC1 running the AD DS and DNS server roles

DC1 is the domain controller and DNS server on your network.

Group Policy default domain policy

After you configure the certificate template on the CA, you can configure the default domain policy in Group Policy so that certificates are autoenrolled to NPS and RAS servers. Group Policy is configured in AD DS on the server DC1.

DNS alias (CNAME) resource record

You must create an alias (CNAME) resource record for the Web server to ensure that other computers can find the server, as well as the AIA and the CRL that are stored on the server. In addition, using an alias CNAME resource record provides flexibility so that you can use the Web server for other purposes, such as hosting Web and FTP sites.

NPS1 running the Network Policy Server role service of the Network Policy and Access Services server role

The NPS is installed when you perform the tasks in the Windows Server 2016 Core Network Guide, so before you perform the tasks in this guide, you should already have one or more NPSs installed on your network.

Group Policy applied and certificate enrolled to servers

After you have configured the certificate template and autoenrollment, you can refresh Group Policy on all target servers. At this time, the servers enroll the server certificate from CA1.

Server certificate deployment process overview

NOTE

The details of how to perform these steps are provided in the section [Server Certificate Deployment](#).

The process of configuring server certificate enrollment occurs in these stages:

1. On WEB1, install the Web Server (IIS) role.
2. On DC1, create an alias (CNAME) record for your Web server, WEB1.
3. Configure your Web server to host the CRL from the CA, then publish the CRL and copy the Enterprise Root CA certificate into the new virtual directory.
4. On the computer where you are planning to install AD CS, assign the computer a static IP address, rename the computer, join the computer to the domain, and then log on to the computer with a user account that is a member of the Domain Admins and Enterprise Admins groups.
5. On the computer where you are planning to install AD CS, configure the CAPolicy.inf file with settings that are specific to your deployment.
6. Install the AD CS server role and perform additional configuration of the CA.
7. Copy the CRL and CA certificate from CA1 to the share on the Web server WEB1.
8. On the CA, configure a copy of the RAS and IAS Servers certificate template. The CA issues certificates based on a certificate template, so you must configure the template for the server certificate before the CA can issue a certificate.
9. Configure server certificate autoenrollment in Group Policy. When you configure autoenrollment, all servers that you have specified with Active Directory group memberships automatically receive a server certificate when Group Policy on each server is refreshed. If you add more servers later, they will automatically receive a server certificate, too.
10. Refresh Group Policy on servers. When Group Policy is refreshed, the servers receive the server certificate, which is based on the template that you configured in the previous step. This certificate is used by the server to prove its identity to client computers and other servers during the authentication process.

NOTE

All domain member computers automatically receive the Enterprise Root CA's certificate without the configuration of autoenrollment. This certificate is different than the server certificate that you configure and distribute by using autoenrollment. The CA's certificate is automatically installed in the Trusted Root Certification Authorities certificate store for all domain member computers so that they will trust certificates that are issued by this CA.

11. Verify that all servers have enrolled a valid server certificate.

Server Certificate Deployment Planning

9/1/2018 • 6 minutes to read • [Edit Online](#)

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

Before you deploy server certificates, you must plan the following items:

- [Plan basic server configuration](#)
- [Plan domain access](#)
- [Plan the location and name of the virtual directory on your Web server](#)
- [Plan a DNS alias \(CNAME\) record for your Web server](#)
- [Plan configuration of CAutoPolicy.inf](#)
- [Plan configuration of the CDP and AIA extensions on CA1](#)
- [Plan the copy operation between the CA and the Web server](#)
- [Plan the configuration of the server certificate template on the CA](#)

Plan basic server configuration

After you install Windows Server 2016 on the computers that you are planning to use as your certification authority and Web server, you must rename the computer and assign and configure a static IP address for the local computer.

For more information, see the Windows Server 2016 [Core Network Guide](#).

Plan domain access

To log on to the domain, the computer must be a domain member computer and the user account must be created in AD DS before the logon attempt. In addition, most procedures in this guide require that the user account is a member of the Enterprise Admins or Domain Admins groups in Active Directory Users and Computers, so you must log on to the CA with an account that has the appropriate group membership.

For more information, see the Windows Server 2016 [Core Network Guide](#).

Plan the location and name of the virtual directory on your Web server

To provide access to the CRL and the CA certificate to other computers, you must store these items in a virtual directory on your Web server. In this guide, the virtual directory is located on the Web server WEB1. This folder is on the "C:" drive and is named "pki." You can locate your virtual directory on your Web server at any folder location that is appropriate for your deployment.

Plan a DNS alias (CNAME) record for your Web server

Alias (CNAME) resource records are also sometimes called canonical name resource records. With these records, you can use more than one name to point to a single host, making it easy to do such things as host both a File Transfer Protocol (FTP) server and a Web server on the same computer. For example, the well-known server names (ftp, www) are registered using alias (CNAME) resource records that map to the Domain Name System (DNS) host

name, such as WEB1, for the server computer that hosts these services.

This guide provides instructions for configuring your Web server to host the certificate revocation list (CRL) for your certification authority (CA). Because you might also want to use your Web server for other purposes, such as to host an FTP or Web site, it's a good idea to create an alias resource record in DNS for your Web server. In this guide, the CNAME record is named "pki," but you can choose a name that is appropriate for your deployment.

Plan configuration of CAPolicy.inf

Before you install AD CS, you must configure CAPolicy.inf on the CA with information that is correct for your deployment. A CAPolicy.inf file contains the following information:

```
[Version]
Signature="$Windows NT$"
[PolicyStatementExtension]
Policies=InternalPolicy
[InternalPolicy]
OID=1.2.3.4.1455.67.89.5
Notice="Legal Policy Statement"
URL=http://pki.corp.contoso.com/pki/cps.txt
[Certsrv_Server]
RenewalKeyLength=2048
RenewalValidityPeriod=Years
RenewalValidityPeriodUnits=5
CRLPeriod=weeks
CRLPeriodUnits=1
LoadDefaultTemplates=0
AlternateSignatureAlgorithm=1
```

You must plan the following items for this file:

- **URL.** The example CAPolicy.inf file has a URL value of <http://pki.corp.contoso.com/pki/cps.txt>. This is because the Web server in this guide is named WEB1 and has a DNS CNAME resource record of pki. The Web server is also joined to the corp.contoso.com domain. In addition, there is a virtual directory on the Web server named "pki" where the certificate revocation list is stored. Ensure that the value that you provide for URL in your CAPolicy.inf file points to a virtual directory on your Web server in your domain.
- **RenewalKeyLength.** The default renewal key length for AD CS in Windows Server 2012 is 2048. The key length that you select should be as long as possible while still providing compatibility with the applications that you intend to use.
- **RenewalValidityPeriodUnits.** The example CAPolicy.inf file has a RenewalValidityPeriodUnits value of 5 years. This is because the expected lifespan of the CA is around ten years. The value of RenewalValidityPeriodUnits should reflect the overall validity period of the CA or the highest number of years for which you want to provide enrollment.
- **CRLPeriodUnits.** The example CAPolicy.inf file has a CRLPeriodUnits value of 1. This is because the example refresh interval for the certificate revocation list in this guide is 1 week. At the interval value that you specify with this setting, you must publish the CRL on the CA to the Web server virtual directory where you store the CRL and provide access to it for computers that are in the authentication process.
- **AlternateSignatureAlgorithm.** This CAPolicy.inf implements an improved security mechanism by implementing alternate signature formats. You should not implement this setting if you still have Windows XP clients that require certificates from this CA.

If you do not plan on adding any subordinate CAs to your public key infrastructure at a later time, and if you want to prevent the addition of any subordinate CAs, you can add the PathLength key to your CAPolicy.inf file with a value of 0. To add this key, copy and paste the following code into your file:

```
[BasicConstraintsExtension]
PathLength=0
Critical=Yes
```

IMPORTANT

It is not recommended that you change any other settings in the CA Policy.inf file unless you have a specific reason for doing so.

Plan configuration of the CDP and AIA extensions on CA1

When you configure the Certificate Revocation List (CRL) Distribution Point (CDP) and the Authority Information Access (AIA) settings on CA1, you need the name of your Web server and your domain name. You also need the name of the virtual directory that you create on your Web server where the certificate revocation list (CRL) and the certification authority certificate are stored.

The CDP location that you must enter during this deployment step has the format:

```
`http:///*DNSAlias\{CNAME\}RecordName*.Domain*.com/*VirtualDirectoryName*/<CaName><CRLNameSuffix>
<DeltaCRLAllowed>.crl.`
```

For example, if your Web server is named WEB1 and your DNS alias CNAME record for the Web server is "pki," your domain is corp.contoso.com, and your virtual directory is named pki, the CDP location is:

```
`http://pki.corp.contoso.com/pki/<CaName><CRLNameSuffix><DeltaCRLAllowed>.crl`
```

The AIA location that you must enter has the format:

```
`http:///*DNSAlias\{CNAME\}RecordName*.Domain*.com/*VirtualDirectoryName*/<ServerDNSName>\_<CaName>
<CertificateName>.crt.`
```

For example, if your Web server is named WEB1 and your DNS alias CNAME record for the Web server is "pki," your domain is corp.contoso.com, and your virtual directory is named pki, the AIA location is:

```
`http://pki.corp.contoso.com/pki/\_<CaName><CertificateName>.crt`
```

Plan the copy operation between the CA and the Web server

To publish the CRL and CA certificate from the CA to the Web server virtual directory, you can run the certutil -crl command after you configure the CDP and AIA locations on the CA. Ensure that you configure the correct paths on the CA Properties **Extensions** tab before you run this command using the instructions in this guide. In addition, to copy the Enterprise CA certificate to the Web server, you must have already created the virtual directory on the Web server and configured the folder as a shared folder.

Plan the configuration of the server certificate template on the CA

To deploy autoenrolled server certificates, you must copy the certificate template named **RAS and IAS Server**. By default, this copy is named **Copy of RAS and IAS Server**. If you want to rename this template copy, plan the name that you want to use during this deployment step.

NOTE

The last three deployment sections in this guide - which allow you to configure server certificate autoenrollment, refresh Group Policy on servers, and verify that the servers have received a valid server certificate from the CA - do not require additional planning steps.

Server Certificate Deployment

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

Follow these steps to install an enterprise root certification authority (CA) and to deploy server certificates for use with PEAP and EAP.

IMPORTANT

Before you install Active Directory Certificate Services, you must name the computer, configure the computer with a static IP address, and join the computer to the domain. After you install AD CS, you cannot change the computer name or the domain membership of the computer, however you can change the IP address if needed. For more information on how to accomplish these tasks, see the Windows Server® 2016 [Core Network Guide](#).

- [Install the Web Server WEB1](#)
- [Create an alias \(CNAME\) record in DNS for WEB1](#)
- [Configure WEB1 to distribute Certificate Revocation Lists \(CRLs\)](#)
- [Prepare the CAPolicy inf file](#)
- [Install the Certification Authority](#)
- [Configure the CDP and AIA extensions on CA1](#)
- [Copy the CA certificate and CRL to the virtual directory](#)
- [Configure the server certificate template](#)
- [Configure server certificate autoenrollment](#)
- [Refresh Group Policy](#)
- [Verify Server Enrollment of a Server Certificate](#)

NOTE

The procedures in this guide do not include instructions for cases in which the **User Account Control** dialog box opens to request your permission to continue. If this dialog box opens while you are performing the procedures in this guide, and if the dialog box was opened in response to your actions, click **Continue**.

Install the Web Server WEB1

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

The Web Server (IIS) role in Windows Server 2016 provides a secure, easy-to-manage, modular and extensible platform for reliably hosting websites, services, and applications. With IIS, you can share information with users on the Internet, an intranet, or an extranet. IIS is a unified web platform that integrates IIS, ASP.NET, FTP services, PHP, and Windows Communication Foundation (WCF).

When you deploy server certificates, your Web server provides you with a location where you can publish the certificate revocation list (CRL) for your certification authority (CA). After publication, the CRL is accessible to all computers on your network so that they can use this list during the authentication process to verify that certificates presented by other computers are not revoked.

If a certificate is on the CRL as revoked, the authentication effort fails and your computer is protected from trusting an entity that has a certificate that is no longer valid.

Before you install the Web Server (IIS) role, ensure that you have configured the server name and IP address and have joined the computer to the domain.

To install the Web Server (IIS) server role

To complete this procedure, you must be a member of the **Administrators** group.

NOTE

To perform this procedure by using Windows PowerShell, open PowerShell, type the following command, and then press ENTER.

```
Install-WindowsFeature Web-Server -IncludeManagementTools
```

1. In Server Manager, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features Wizard opens.

2. In **Before You Begin**, click **Next**.

Note

The **Before You Begin** page of the Add Roles and Features Wizard is not displayed if you have previously run the Add Roles and Features Wizard and you selected **Skip this page by default** at that time.

1. On the **Installation Type** page, click **Next**.
2. On the **Server selection** page, click **Next**.
3. On the **Server roles** page, select **Web Server (IIS)**, and then click **Next**.
4. Click **Next** until you have accepted all of the default web server settings, and then click **Install**.
5. Verify that all installations were successful, and then click **Close**.

Create an Alias (CNAME) Record in DNS for WEB1

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to add an Alias canonical name (CNAME) resource record for your Web server to a zone in DNS on your domain controller. With CNAME records, you can use more than one name to point to a single host, making it easy to do such things as host both a File Transfer Protocol (FTP) server and a Web server on the same computer.

Because of this, you are free to use your Web server to host the certificate revocation list (CRL) for your certification authority (CA) as well as to perform additional services, such as FTP or Web server.

When you perform this procedure, replace **Alias name** and other variables with values that are appropriate for your deployment.

To perform this procedure, you must be a member of **Domain Admins**.

To add an alias (CNAME) resource record to a zone

NOTE

To perform this procedure by using Windows PowerShell, see [Add-DnsServerResourceRecordCName](#).

1. On DC1, in Server Manager, click **Tools** and then click **DNS**. The DNS Manager Microsoft Management Console (MMC) opens.
2. In the console tree, double-click **Forward Lookup Zones**, right-click the forward lookup zone where you want to add the Alias resource record, and then click **New Alias (CNAME)**. The **New Resource Record** dialog box opens.
3. In **Alias name**, type the alias name **pki**.
4. When you type a value for **Alias name**, the **Fully qualified domain name (FQDN)** auto-fills in the dialog box. For example, if your alias name is "pki" and your domain is corp.contoso.com, the value **pki.corp.contoso.com** is auto-filled for you.
5. In **Fully qualified domain name (FQDN) for target host**, type the FQDN of your Web server. For example, if your Web server is named WEB1 and your domain is corp.contoso.com, type **WEB1.corp.contoso.com**.
6. Click **OK** to add the new record to the zone.

Configure WEB1 to Distribute Certificate Revocation Lists (CRLs)

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to configure the web server WEB1 to distribute CRLs.

In the extensions of the root CA, it was stated that the CRL from the root CA would be available via <http://pki.corp.contoso.com/pki>. Currently, there is not a PKI virtual directory on WEB1, so one must be created.

To perform this procedure, you must be a member of **Domain Admins**.

NOTE

In the procedure below, replace the user account name, the Web server name, folder names and locations, and other values with those that are appropriate for your deployment.

To configure WEB1 to distribute certificates and CRLs

1. On WEB1, run Windows PowerShell as an administrator, type `explorer c:\`, and then press ENTER. Windows Explorer opens to drive C.
2. Create a new folder named PKI on the C: drive. To do so, click **Home**, and then click **New Folder**. A new folder is created with the temporary name highlighted. Type **pki** and then press ENTER.
3. In Windows Explorer, right-click the folder you just created, hover the mouse cursor over **Share with**, and then click **Specific people**. The **File Sharing** dialog box opens.
4. In **File Sharing**, type **Cert Publishers**, and then click **Add**. The Cert Publishers group is added to the list. In the list, in **Permission Level**, click the arrow next to **Cert Publishers**, and then click **Read/Write**. Click **Share**, and then click **Done**.
5. Close Windows Explorer.
6. Open the IIS console. In Server Manager, click **Tools**, and then click **Internet Information Services (IIS) Manager**.
7. In the Internet Information Services (IIS) Manager console tree, expand **WEB1**. If you are invited to get started with Microsoft Web Platform, click **Cancel**.
8. Expand **Sites** and then right-click the **Default Web Site** and then click **Add Virtual Directory**.
9. In **Alias**, type **pki**. In **Physical path** type **C:\pki**, then click **OK**.
10. Enable Anonymous access to the pki virtual directory, so that any client can check the validity of the CA certificates and CRLs. To do so:
 - a. In the **Connections** pane, ensure that **pki** is selected.
 - b. On **pki Home** click **Authentication**.
 - c. In the **Actions** pane, click **Edit Permissions**.
 - d. On the **Security** tab, click **Edit**

- e. On the **Permissions for pki** dialog box, click **Add**.
 - f. In the **Select Users, Computers, Service Accounts, or Groups**, type **ANONYMOUS LOGON; Everyone** and then click **Check Names**. Click **OK**.
 - g. Click **OK** on the **Select Users, Computers, Service Accounts or Groups** dialog box.
 - h. Click **OK** on the **Permissions for pki** dialog box.
11. Click **OK** on the **pki Properties** dialog box.
12. In the **pki Home** pane, double-click **Request Filtering**.
13. The **File Name Extensions** tab is selected by default in the **Request Filtering** pane. In the **Actions** pane, click **Edit Feature Settings**.
14. In **Edit Request Filtering Settings**, select **Allow double escaping** and then click **OK**.
15. In the Internet Information Services (IIS) Manager MMC, click your Web server name. For example, if your Web server is named **WEB1**, click **WEB1**.
16. In **Actions**, click **Restart**. Internet services are stopped and then restarted.

CAPolicy.inf Syntax

8/3/2018 • 9 minutes to read • [Edit Online](#)

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

The CAPolicy.inf is a configuration file that defines the extensions, constraints, and other configuration settings that are applied to a root CA certificate and all certificates issued by the root CA. The CAPolicy.inf file must be installed on a host server before the setup routine for the root CA begins. When the security restrictions on a root CA are to be modified, the root certificate must be renewed and an updated CAPolicy.inf file must be installed on the server before the renewal process begins.

The CAPolicy.inf is:

- Created and defined manually by an administrator
- Utilized during the creation of root and subordinate CA certificates
- Defined on the signing CA where you sign and issue the certificate (not the CA where the request is granted)

Once you have created your CAPolicy.inf file, you must copy it into the **%systemroot%** folder of your server before you install ADCS or renew the CA certificate.

The CAPolicy.inf makes it possible to specify and configure a wide variety of CA attributes and options. The following section describes all the options for you to create an .inf file tailored to your specific needs.

CAPolicy.inf File Structure

The following terms are used to describe the .inf file structure:

- *Section* – is an area of the file that covers a logical group of keys. Section names in .inf files are identified by appearing in brackets. Many, but not all, sections are used to configure certificate extensions.
- *Key* – is the name of an entry and appears to the left of the equal sign.
- *Value* – is the parameter and appears to the right of the equal sign.

In the example below, **[Version]** is the section, **Signature** is the key, and **"\$Windows NT\$"** is the value.

Example:

```
[Version]          #section
Signature="$Windows NT$"    #key=value
```

Version

Identifies the file as an .inf file. Version is the only required section and must be at the beginning of your CAPolicy.inf file.

PolicyStatementExtension

Lists the policies that have been defined by the organization, and whether they are optional or mandatory. Multiple policies are separated by commas. The names have meaning in the context of a specific deployment, or in relation to custom applications that check for the presence of these policies.

For each policy defined, there must be a section that defines the settings for that particular policy. For each policy,

you need to provide a user-defined object identifier (OID) and either the text you want displayed as the policy statement or a URL pointer to the policy statement. The URL can be in the form of an HTTP, FTP, or LDAP URL.

If you are going to have descriptive text in the policy statement, then the next three lines of the CA Policy.inf would look like:

```
[InternalPolicy]  
OID=1.1.1.1.1.1.1  
Notice="Legal policy statement text"
```

If you are going to use a URL to host the CA policy statement, then next three lines would instead look like:

```
[InternalPolicy]  
OID=1.1.1.1.1.1.2  
URL=http://pki.wingtiptoys.com/policies/legalpolicy.asp
```

In addition:

- Multiple URL and Notice keys are supported.
- Notice and URL keys in the same policy section are supported.
- URLs with spaces or text with spaces must be surrounded by quotes. This is true for the **URL** key, regardless of the section in which it appears.

An example of multiple notices and URLs in a policy section would look like:

```
[InternalPolicy]  
OID=1.1.1.1.1.1.1  
URL=http://pki.wingtiptoys.com/policies/legalpolicy.asp  
URL=ftp://ftp.wingtiptoys.com/pki/policies/legalpolicy.asp  
Notice="Legal policy statement text"
```

CRLDistributionPoint

You can specify CRL Distribution Points (CDPs) for a root CA certificate in the CA Policy.inf. After installing the CA, you can configure the CDP URLs that the CA includes in each certificate issued. The root CA certificate shows the URLs specified in this section of the CA Policy.inf file.

```
[CRLDistributionPoint]  
URL=http://pki.wingtiptoys.com/cdp/WingtipToysRootCA.crl
```

Some additional information about this section:

- Supports:
 - HTTP
 - File URLs
 - LDAP URLs
 - Multiple URLs

IMPORTANT

Does not support HTTPS URLs.

- Quotes must surround URLs with spaces.

- If no URLs are specified – that is, if the **[CRLDistributionPoint]** section exists in the file but is empty – the Authority Information Access extension is omitted from the root CA certificate. This is usually preferable when setting up a root CA. Windows does not perform revocation checking on a root CA certificate, so the CDP extension is superfluous in a root CA certificate.
- CA can publish to FILE UNC, for example, to a share that represents the folder of a website where a client retrieves via HTTP.
- Only use this section if you are setting up a root CA or renewing the root CA certificate. The CA determines the subordinate CA CDP extensions.

AuthorityInformationAccess

You can specify the authority information access points in the CAPolicy.inf for the root CA certificate.

```
[AuthorityInformationAccess]
URL=http://pki.wingtiptoys.com/Public/myCA.crt
```

Some additional notes on the authority information access section:

- Multiple URLs are supported.
- HTTP, FTP, LDAP and FILE URLs are supported. HTTPS URLs are not supported.
- This section is only used if you are setting up a root CA, or renewing the root CA certificate. Subordinate CA AIA extensions are determined by the CA which issued the subordinate CA's certificate.
- URLs with spaces must be surrounded by quotes.
- If no URLs are specified – that is, if the **[AuthorityInformationAccess]** section exists in the file but is empty – the CRL Distribution Point extension is omitted from the root CA certificate. Again, this would be the preferred setting in the case of a root CA certificate as there is no authority higher than a root CA that would need to be referenced by a link to its certificate.

certsrv_Server

Another optional section of the CAPolicy.inf is [certsrv_server], which is used to specify renewal key length, the renewal validity period, and the certificate revocation list (CRL) validity period for a CA that is being renewed or installed. None of the keys in this section are required. Many of these settings have default values that are sufficient for most needs and can simply be omitted from the CAPolicy.inf file. Alternatively, many of these settings can be changed after the CA has been installed.

An example would look like:

```
[certsrv_server]
RenewalKeyLength=2048
RenewalValidityPeriod=Years
RenewalValidityPeriodUnits=5
CRLPeriod=Days
CRLPeriodUnits=2
CRLDeltaPeriod=Hours
CRLDeltaPeriodUnits=4
ClockSkewMinutes=20
LoadDefaultTemplates=True
AlternateSignatureAlgorithm=0
ForceUTF8=0
EnableKeyCounting=0
```

RenewalKeyLength sets the key size for renewal only. This is only used when a new key pair is generated during CA certificate renewal. The key size for the initial CA certificate is set when the CA is installed.

When renewing a CA certificate with a new key pair, the key length can be either increased or decreased. For example, if you have set a root CA key size of 4096 bytes or higher, and then discover that you have Java apps or network devices that can only support key sizes of 2048 bytes. Whether you increase or decrease the size, you must reissue all the certificates issued by that CA.

RenewalValidityPeriod and **RenewalValidityPeriodUnits** establish the lifetime of the new root CA certificate when renewing the old root CA certificate. It only applies to a root CA. The certificate lifetime of a subordinate CA is determined by its superior. **RenewalValidityPeriod** can have the following values: Hours, Days, Weeks, Months, and Years.

CRLPeriod and **CRLPeriodUnits** establish the validity period for the base CRL. **CRLPeriod** can have the following values: Hours, Days, Weeks, Months, and Years.

CRLDeltaPeriod and **CRLDeltaPeriodUnits** establish the validity period of the delta CRL. **CRLDeltaPeriod** can have the following values: Hours, Days, Weeks, Months, and Years.

Each of these settings can be configured after the CA has been installed:

```
Certutil -setreg CACRLPeriod Weeks  
Certutil -setreg CACRLPeriodUnits 1  
Certutil -setreg CACRLDeltaPeriod Days  
Certutil -setreg CACRLDeltaPeriodUnits 1
```

Remember to restart Active Directory Certificate Services for any changes to take effect.

LoadDefaultTemplates only applies during the install of an Enterprise CA. This setting, either True or False (or 1 or 0), dictates if the CA is configured with any of the default templates.

In a default installation of the CA, a subset of the default certificate templates is added to the Certificate Templates folder in the Certification Authority snap-in. This means that as soon as the AD CS service starts after the role has been installed a user or computer with sufficient permissions can immediately enroll for a certificate.

You may not want to issue any certificates immediately after a CA has been installed, so you can use the **LoadDefaultTemplates** setting to prevent the default templates from being added to the Enterprise CA. If there are no templates configured on the CA then it can issue no certificates.

AlternateSignatureAlgorithm configures the CA to support the PKCS#1 V2.1 signature format for both the CA certificate and certificate requests. When set to 1 on a root CA the CA certificate will include the PKCS#1 V2.1 signature format. When set on a subordinate CA, the subordinate CA will create a certificate request that includes the PKCS#1 V2.1 signature format.

ForceUTF8 changes the default encoding of relative distinguished names (RDNs) in Subject and Issuer distinguished names to UTF-8. Only those RDNs that support UTF-8, such as those that are defined as Directory String types by an RFC, are affected. For example, the RDN for Domain Component (DC) supports encoding as either IA5 or UTF-8, while the Country RDN (C) only supports encoding as a Printable String. The ForceUTF8 directive will therefore affect a DC RDN but will not affect a C RDN.

EnableKeyCounting configures the CA to increment a counter every time the CA's signing key is used. Do not enable this setting unless you have a Hardware Security Module (HSM) and associated cryptographic service provider (CSP) that supports key counting. Neither the Microsoft Strong CSP nor the Microsoft Software Key Storage Provider (KSP) support key counting.

Create the CAPolicy.inf file

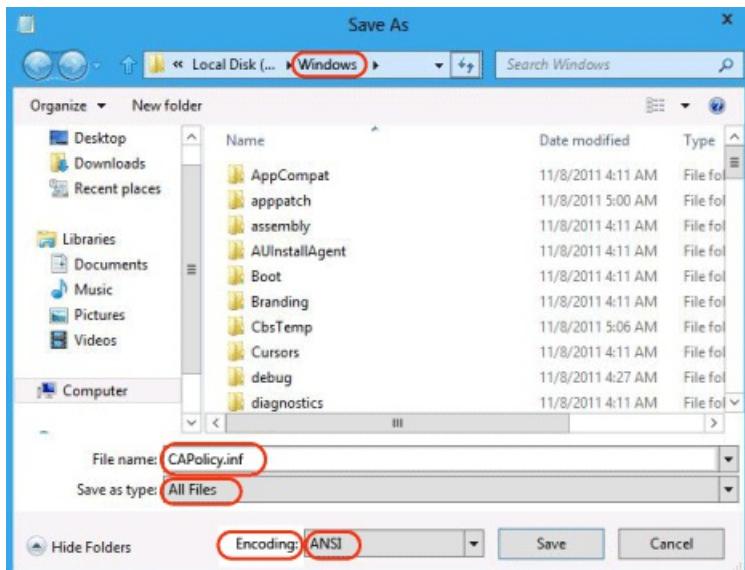
Before you install AD CS, you configure the CAPolicy.inf file with specific settings for your deployment.

Prerequisite: You must be a member of the Administrators group.

1. On the computer where you are planning to install AD CS, open Windows PowerShell, type **notepad** **c:\CAPolicy.inf** and press ENTER.
2. When prompted to create a new file, click **Yes**.
3. Enter the following as the contents of the file:

```
[Version]
Signature="$Windows NT$"
[PolicyStatementExtension]
Policies=InternalPolicy
[InternalPolicy]
OID=1.2.3.4.1455.67.89.5
Notice="Legal Policy Statement"
URL=http://pki.corp.contoso.com/pki/cps.txt
[Certsrv_Server]
RenewalKeyLength=2048
RenewalValidityPeriod=Years
RenewalValidityPeriodUnits=5
CRLPeriod=weeks
CRLPeriodUnits=1
LoadDefaultTemplates=0
AlternateSignatureAlgorithm=1
[CRLDistributionPoint]
[AuthorityInformationAccess]
```

4. Click **File**, and then click **Save As**.
5. Navigate to the %systemroot% folder.
6. Ensure the following:
 - **File name** is set to **CAPolicy.inf**
 - **Save as type** is set to **All Files**
 - **Encoding** is **ANSI**
7. Click **Save**.
8. When you are prompted to overwrite the file, click **Yes**.



Caution

Be sure to save the CAPolicy.inf with the inf extension. If you do not specifically type **.inf** at the end of the file name and select the options as described, the file will be saved as a text file and will not be used during CA

installation.

9. Close Notepad.

IMPORTANT

In the CAPolicy.inf, you can see there is a line specifying the URL <http://pki.corp.contoso.com/pki/cps.txt>. The Internal Policy section of the CAPolicy.inf is just shown as an example of how you would specify the location of a certificate practice statement (CPS). In this guide, you are not instructed to create the certificate practice statement (CPS).

Install the Certification Authority

9/1/2018 • 3 minutes to read • [Edit Online](#)

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

You can use this procedure to install Active Directory Certificate Services (AD CS) so that you can enroll a server certificate to servers that are running Network Policy Server (NPS), Routing and Remote Access Service (RRAS), or both.

IMPORTANT

- Before you install Active Directory Certificate Services, you must name the computer, configure the computer with a static IP address, and join the computer to the domain. For more information on how to accomplish these tasks, see the Windows Server 2016 [Core Network Guide](#).
- To perform this procedure, the computer on which you are installing AD CS must be joined to a domain where Active Directory Domain Services (AD DS) is installed.

Membership in both the **Enterprise Admins** and the root domain's **Domain Admins** group is the minimum required to complete this procedure.

NOTE

To perform this procedure by using Windows PowerShell, open Windows PowerShell and type the following command, and then press ENTER.

```
Add-WindowsFeature Adcs-Cert-Authority -IncludeManagementTools
```

After AD CS is installed, type the following command and press ENTER.

```
Install-AdcsCertificationAuthority -CAType EnterpriseRootCA
```

To install Active Directory Certificate Services

- Log on as a member of both the Enterprise Admins group and the root domain's Domain Admins group.
- In Server Manager, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features Wizard opens.
- In **Before You Begin**, click **Next**.

NOTE

The **Before You Begin** page of the Add Roles and Features Wizard is not displayed if you have previously selected **Skip this page by default** when the Add Roles and Features Wizard was run.

- In **Select Installation Type**, ensure that **Role-Based or feature-based installation** is selected, and then click **Next**.
- In **Select destination server**, ensure that **Select a server from the server pool** is selected. In **Server Pool**, ensure that the local computer is selected. Click **Next**.
- In **Select Server Roles**, in **Roles**, select **Active Directory Certificate Services**. When you are prompted

to add required features, click **Add Features**, and then click **Next**.

7. In **Select features**, click **Next**.
8. In **Active Directory Certificate Services**, read the provided information, and then click **Next**.
9. In **Confirm installation selections**, click **Install**. Do not close the wizard during the installation process. When installation is complete, click **Configure Active Directory Certificate Services on the destination server**. The AD CS Configuration wizard opens. Read the credentials information and, if needed, provide the credentials for an account that is a member of the Enterprise Admins group. Click **Next**.
10. In **Role Services**, click **Certification Authority**, and then click **Next**.
11. On the **Setup Type** page, verify that **Enterprise CA** is selected, and then click **Next**.
12. On the **Specify the type of the CA** page, verify that **Root CA** is selected, and then click **Next**.
13. On the **Specify the type of the private key** page, verify that **Create a new private key** is selected, and then click **Next**.
14. On the **Cryptography for CA** page, keep the default settings for CSP (**RSA#Microsoft Software Key Storage Provider**) and hash algorithm (**SHA2**), and determine the best key character length for your deployment. Large key character lengths provide optimal security; however, they can impact server performance and might not be compatible with legacy applications. It is recommended that you keep the default setting of 2048. Click **Next**.
15. On the **CA Name** page, keep the suggested common name for the CA or change the name according to your requirements. Ensure that you are certain the CA name is compatible with your naming conventions and purposes, because you cannot change the CA name after you have installed AD CS. Click **Next**.
16. On the **Validity Period** page, in **Specify the validity period**, type the number and select a time value (Years, Months, Weeks, or Days). The default setting of five years is recommended. Click **Next**.
17. On the **CA Database** page, in **Specify the database locations**, specify the folder location for the certificate database and the certificate database log. If you specify locations other than the default locations, ensure that the folders are secured with access control lists (ACLs) that prevent unauthorized users or computers from accessing the CA database and log files. Click **Next**.
18. In **Confirmation**, click **Configure** to apply your selections, and then click **Close**.

Configure the CDP and AIA Extensions on CA1

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to configure the Certificate Revocation List (CRL) Distribution Point (CDP) and the Authority Information Access (AIA) settings on CA1.

To perform this procedure, you must be a member of Domain Admins.

To configure the CDP and AIA extensions on CA1

1. In Server Manager, click **Tools** and then click **Certification Authority**.
2. In the Certification Authority console tree, right-click **corp-CA1-CA**, and then click **Properties**.

NOTE

The name of your CA is different if you did not name the computer CA1 and your domain name is different than the one in this example. The CA name is in the format *domain-CAComputerName-CA*.

3. Click the **Extensions** tab. Ensure that **Select extension** is set to **CRL Distribution Point (CDP)**, and in the **Specify locations from which users can obtain a certificate revocation list (CRL)**, do the following:

a. Select the entry `file://\\<ServerDNSName>\CertEnroll\<CaName><CRLNameSuffix><DeltaCRLAllowed>.crl`, and then click **Remove**. In **Confirm removal**, click **Yes**.

b. Select the entry `http://<ServerDNSName>/CertEnroll/<CaName><CRLNameSuffix><DeltaCRLAllowed>.crl`, and then click **Remove**. In **Confirm removal**, click **Yes**.

c. Select the entry that starts with the path

`ldap:///CN=<CATruncatedName><CRLNameSuffix>,CN=<ServerShortName>`, and then click **Remove**. In **Confirm removal**, click **Yes**.

4. In **Specify locations from which users can obtain a certificate revocation list (CRL)**, click **Add**. The **Add Location** dialog box opens.

5. In **Add Location**, in **Location**, type

`http://pki.corp.contoso.com/pki/<CaName><CRLNameSuffix><DeltaCRLAllowed>.crl`, and then click **OK**. This returns you to the CA properties dialog box.

6. On the **Extensions** tab, select the following check boxes:

- **Include in CRLs. Clients use this to find the Delta CRL locations**
- **Include in the CDP extension of issued certificates**

7. In **Specify locations from which users can obtain a certificate revocation list (CRL)**, click **Add**. The **Add Location** dialog box opens.

8. In **Add Location**, in **Location**, type

`file://\\pki.corp.contoso.com\\pki\\<CaName><CRLNameSuffix><DeltaCRLAllowed>.crl`, and then click **OK**. This returns you to the CA properties dialog box.

9. On the **Extensions** tab, select the following check boxes:

- Publish CRLs to this location
- Publish Delta CRLs to this location

10. Change **Select extension** to **Authority Information Access (AIA)**, and in the **Specify locations from which users can obtain a certificate revocation list (CRL)**, do the following:

- Select the entry that starts with the path `ldap:///CN=<CATruncatedName>,CN=AIA,CN=Public Key Services`, and then click **Remove**. In **Confirm removal**, click **Yes**.
- Select the entry `http://<ServerDNSName>/CertEnroll/<ServerDNSName>_<CaName><CertificateName>.crt`, and then click **Remove**. In **Confirm removal**, click **Yes**.
- Select the entry `file://\\<ServerDNSName>\CertEnroll\<ServerDNSName><CaName><CertificateName>.crt`, and then click **Remove**. In **Confirm removal**, click **Yes**.

11. In **Specify locations from which users can obtain a certificate revocation list (CRL)**, click **Add**. The **Add Location** dialog box opens.

12. In **Add Location**, in **Location**, type `http://pki.corp.contoso.com/pki/<ServerDNSName>_<CaName><CertificateName>.crt`, and then click **OK**. This returns you to the CA properties dialog box.
13. On the **Extensions** tab, select **Include in the AIA of issued certificates**.
14. When prompted to restart Active Directory Certificate Services, click **No**. You will restart the service later.

Copy the CA Certificate and CRL to the Virtual Directory

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to copy the Certificate Revocation List and Enterprise root CA certificate from your certification authority to a virtual directory on your Web server, and to ensure that AD CS is configured correctly. Before running the commands below, ensure that you replace directory and server names with those that are appropriate for your deployment.

To perform this procedure you must be a member of **Domain Admins**.

To copy the certificate revocation list from CA1 to WEB1

1. On CA1, run Windows PowerShell as an Administrator, and then publish the CRL with the following command:
 - Type `certutil -crl`, and then press ENTER.
 - To copy the certificate revocation lists to the file share on your Web server, type
`copy C:\Windows\system32\certsrv\cenrollment*.crl \\WEB1\pki`, and then press ENTER.
2. To verify that your CDP and AIA extension locations are correctly configured, type `pkiview.msc`, and then press ENTER. The pkiview Enterprise PKI MMC opens.
3. In the left pane, click your CA name.

For example, if your CA name is corp-CA1-CA, click **corp-CA1-CA**.

4. In the Status column of the results pane, verify that the values for the following shows **OK**:
 - **CA Certificate**
 - **AIA Location #1**
 - **CDP Location #1**

TIP

If **Status** for any item is not **OK**, do the following:

- Open the share on your Web server to verify that the certificate and certificate revocation list files were successfully copied to the share. If they were not successfully copied to the share, modify your copy commands with the correct file source and share destination and run the commands again.
- Verify that you have entered the correct locations for the CDP and AIA on the CA Extensions tab. Ensure that there are no extra spaces or other characters in the locations that you have provided.
- Verify that you copied the CRL and CA certificate to the correct location on your Web server, and that the location matches the location you provided for the CDP and AIA locations on the CA.
- Verify that you correctly configured permissions for the virtual folder where the CA certificate and CRL are stored.

Configure the Server Certificate Template

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to configure the certificate template that Active Directory® Certificate Services (AD CS) uses as the basis for server certificates that are enrolled to servers on your network.

While configuring this template, you can specify the servers by Active Directory group that should automatically receive a server certificate from AD CS.

The procedure below includes instructions for configuring the template to issue certificates to all of the following server types:

- Servers that are running the Remote Access service, including RAS Gateway servers, that are members of the **RAS and IAS Servers** group.
- Servers that are running the Network Policy Server (NPS) service that are members of the **RAS and IAS Servers** group.

Membership in both the **Enterprise Admins** and the root domain's **Domain Admins** group is the minimum required to complete this procedure.

To configure the certificate template

1. On CA1, in Server Manager, click **Tools**, and then click **Certification Authority**. The Certification Authority Microsoft Management Console (MMC) opens.
2. In the MMC, double-click the CA name, right-click **Certificate Templates**, and then click **Manage**.
3. The Certificate Templates console opens. All of the certificate templates are displayed in the details pane.
4. In the details pane, click the **RAS and IAS Server** template.
5. Click the **Action** menu, and then click **Duplicate Template**. The template **Properties** dialog box opens.
6. Click the **Security** tab.
7. On the **Security** tab, in **Group or user names**, click **RAS and IAS servers**.
8. In **Permissions for RAS and IAS servers**, under **Allow**, ensure that **Enroll** is selected, and then select the **Autoenroll** check box. Click **OK**, and close the Certificate Templates MMC.
9. In the Certification Authority MMC, click **Certificate Templates**. On the **Action** menu, point to **New**, and then click **Certificate Template to Issue**. The **Enable Certificate Templates** dialog box opens.
10. In **Enable Certificate Templates**, click the name of the certificate template that you just configured, and then click **OK**. For example, if you did not change the default certificate template name, click **Copy of RAS and IAS Server**, and then click **OK**.

Configure certificate auto-enrollment

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

NOTE

Before you perform this procedure, you must configure a server certificate template by using the Certificate Templates Microsoft Management Console snap-in on a CA that is running AD CS. Membership in both the **Enterprise Admins** and the root domain's **Domain Admins** group is the minimum required to complete this procedure.

Configure server certificate auto-enrollment

1. On the computer where AD DS is installed, open Windows PowerShell®, type **mmc**, and then press ENTER. The Microsoft Management Console opens.
2. On the **File** menu, click **Add/Remove Snap-in**. The **Add or Remove Snap-ins** dialog box opens.
3. In **Available snap-ins**, scroll down to and double-click **Group Policy Management Editor**. The **Select Group Policy Object** dialog box opens.

IMPORTANT

Ensure that you select **Group Policy Management Editor** and not **Group Policy Management**. If you select **Group Policy Management**, your configuration using these instructions will fail and a server certificate will not be autoenrolled to your NPSs.

4. In **Group Policy Object**, click **Browse**. The **Browse for a Group Policy Object** dialog box opens.
5. In **Domains, OUs, and linked Group Policy Objects**, click **Default Domain Policy**, and then click **OK**.
6. Click **Finish**, and then click **OK**.
7. Double-click **Default Domain Policy**. In the console, expand the following path: **Computer Configuration, Policies, Windows Settings, Security Settings**, and then **Public Key Policies**.
8. Click **Public Key Policies**. In the details pane, double-click **Certificate Services Client - Auto-Enrollment**. The **Properties** dialog box opens. Configure the following items, and then click **OK**:
 - a. In **Configuration Model**, select **Enabled**.
 - b. Select the **Renew expired certificates, update pending certificates, and remove revoked certificates** check box.
 - c. Select the **Update certificates that use certificate templates** check box.
9. Click **OK**.

Configure user certificate auto-enrollment

1. On the computer where AD DS is installed, open Windows PowerShell®, type **mmc**, and then press ENTER. The Microsoft Management Console opens.
2. On the **File** menu, click **Add/Remove Snap-in**. The **Add or Remove Snap-ins** dialog box opens.
3. In **Available snap-ins**, scroll down to and double-click **Group Policy Management Editor**. The **Select Group Policy Object** dialog box opens.

IMPORTANT

Ensure that you select **Group Policy Management Editor** and not **Group Policy Management**. If you select **Group Policy Management**, your configuration using these instructions will fail and a server certificate will not be autoenrolled to your NPSs.

4. In **Group Policy Object**, click **Browse**. The **Browse for a Group Policy Object** dialog box opens.
5. In **Domains, OUs, and linked Group Policy Objects**, click **Default Domain Policy**, and then click **OK**.
6. Click **Finish**, and then click **OK**.
7. Double-click **Default Domain Policy**. In the console, expand the following path: **User Configuration, Policies, Windows Settings, Security Settings**, and then **Public Key Policies**.
8. Click **Public Key Policies**. In the details pane, double-click **Certificate Services Client - Auto-Enrollment**. The **Properties** dialog box opens. Configure the following items, and then click **OK**:
 - a. In **Configuration Model**, select **Enabled**.
 - b. Select the **Renew expired certificates, update pending certificates, and remove revoked certificates** check box.
 - c. Select the **Update certificates that use certificate templates** check box.
9. Click **OK**.

Next Steps

[Refresh Group Policy](#)

Refresh Group Policy

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to manually refresh Group Policy on the local computer. When Group Policy is refreshed, if certificate autoenrollment is configured and functioning correctly, the local computer is autoenrolled a certificate by the certification authority (CA).

NOTE

Group Policy is automatically refreshed when you restart the domain member computer, or when a user logs on to a domain member computer. In addition, Group Policy is periodically refreshed. By default, this periodic refresh is performed every 90 minutes with a randomized offset of up to 30 minutes.

Membership in **Administrators**, or equivalent, is the minimum required to complete this procedure.

To refresh Group Policy on the local computer

1. On the computer where NPS is installed, open Windows PowerShell® by using the icon on the taskbar.
2. At the Windows PowerShell prompt, type **gpupdate**, and then press ENTER.

Verify Server Enrollment of a Server Certificate

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to verify that your Network Policy Server (NPS) servers have enrolled a server certificate from the certification authority (CA).

NOTE

Membership in the **Domain Admins** group is the minimum required to complete these procedures.

Verify Network Policy Server (NPS) enrollment of a server certificate

Because NPS is used to authenticate and authorize network connection requests, it is important to ensure that the server certificate you have issued to NPSs is valid when used in network policies.

To verify that a server certificate is correctly configured and is enrolled to the NPS, you must configure a test network policy and allow NPS to verify that NPS can use the certificate for authentication.

To verify NPS enrollment of a server certificate

1. In Server Manager, click **Tools**, and then click **Network Policy Server**. The Network Policy Server Microsoft Management Console (MMC) opens.
2. Double-click **Policies**, right-click **Network Policies**, and click **New**. The New Network Policy wizard opens.
3. In **Specify Network Policy Name and Connection Type**, in **Policy name**, type **Test policy**. Ensure that **Type of network access server** has the value **Unspecified**, and then click **Next**.
4. In **Specify Conditions**, click **Add**. In **Select condition**, click **Windows Groups**, and then click **Add**.
5. In **Groups**, click **Add Groups**. In **Select Group**, type **Domain Users**, and then press ENTER. Click **OK**, and then click **Next**.
6. In **Specify Access Permission**, ensure that **Access granted** is selected, and then click **Next**.
7. In **Configure Authentication Methods**, click **Add**. In **Add EAP**, click **Microsoft: Protected EAP (PEAP)**, and then click **OK**. In **EAP Types**, select **Microsoft: Protected EAP (PEAP)**, and then click **Edit**. The **Edit Protected EAP Properties** dialog box opens.
8. In the **Edit Protected EAP Properties** dialog box, in **Certificate issued to**, NPS displays the name of your server certificate in the format *ComputerName.Domain*. For example, if your NPS is named NPS-01 and your domain is example.com, NPS displays the certificate **NPS-01.example.com**. In addition, in **Issuer**, the name of your certification authority is displayed, and in **Expiration date**, the date of expiration of the server certificate is shown. This demonstrates that your NPS has enrolled a valid server certificate that it can use to prove its identity to client computers that are trying to access the network through your network access servers, such as virtual private network (VPN) servers, 802.1X-capable wireless access points, Remote Desktop Gateway servers, and 802.1X-capable Ethernet switches.

IMPORTANT

If NPS does not display a valid server certificate and if it provides the message that such a certificate cannot be found on the local computer, there are two possible reasons for this problem. It is possible that Group Policy did not refresh properly, and the NPS has not enrolled a certificate from the CA. In this circumstance, restart the NPS. When the computer restarts, Group Policy is refreshed, and you can perform this procedure again to verify that the server certificate is enrolled. If refreshing Group Policy does not resolve this issue, either the certificate template, certificate autoenrollment, or both are not configured correctly. To resolve these issues, start at the beginning of this guide and perform all steps again to ensure that the settings that you have provided are accurate.

9. When you have verified the presence of a valid server certificate, you can click **OK** and **Cancel** to exit the New Network Policy wizard.

NOTE

Because you are not completing the wizard, the test network policy is not created in NPS.

Deploy Password-Based 802.1X Authenticated Wireless Access

9/1/2018 • 21 minutes to read • [Edit Online](#)

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

This is a companion guide to the Windows Server® 2016 Core Network Guide. The Core Network Guide provides instructions for planning and deploying the components required for a fully functioning network and a new Active Directory® domain in a new forest.

This guide explains how to build upon a core network by providing instructions about how to deploy Institute of Electrical and Electronics Engineers (IEEE) 802.1X-authenticated IEEE 802.11 wireless access using Protected Extensible Authentication Protocol – Microsoft Challenge Handshake Authentication Protocol version 2 (PEAP-MS-CHAP v2).

Because PEAP-MS-CHAP v2 requires that users provide password-based credentials rather than a certificate during the authentication process, it is typically easier and less expensive to deploy than EAP-TLS or PEAP-TLS.

NOTE

In this guide, IEEE 802.1X Authenticated Wireless Access with PEAP-MS-CHAP v2 is abbreviated to “wireless access” and “WiFi access.”

About this guide

This guide, in combination with the prerequisite guides described below, provides instructions about how to deploy the following WiFi access infrastructure.

- One or more 802.1X-capable 802.11 wireless access points (APs).
- Active Directory Domain Services (AD DS) Users and Computers.
- Group Policy Management.
- One or more Network Policy Server (NPS) servers.
- Server certificates for computers running NPS.
- Wireless client computers running Windows® 10, Windows 8.1, or Windows 8.

Dependencies for this guide

To successfully deploy authenticated wireless with this guide, you must have a network and domain environment with all of the required technologies deployed. You must also have server certificates deployed to your authenticating NPSs.

The following sections provide links to documentation that shows you how to deploy these technologies.

Network and domain environment dependencies

This guide is designed for network and system administrators who have followed the instructions in the Windows Server 2016 **Core Network Guide** to deploy a core network, or for those who have previously deployed the core technologies included in the core network, including AD DS, Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), TCP/IP, NPS, and Windows Internet Name Service (WINS).

The Core Network Guide is available at the following locations:

- The Windows Server 2016 [Core Network Guide](#) is available in the Windows Server 2016 Technical Library.
- The [Core Network Guide](#) is also available in Word format at TechNet Gallery, at <https://gallery.technet.microsoft.com/Core-Network-Guide-for-9da2e683>.

Server certificate dependencies

There are two available options for enrolling authentication servers with server certificates for use with 802.1X authentication - deploy your own public key infrastructure by using Active Directory Certificate Services (AD CS) or use server certificates that are enrolled by a public certification authority (CA).

AD CS

Network and system administrators deploying authenticated wireless must follow the instructions in the Windows Server 2016 Core Network Companion Guide, **Deploy Server Certificates for 802.1X Wired and Wireless Deployments**. This guide explains how to deploy and use AD CS to autoenroll server certificates to computers running NPS.

This guide is available at the following location.

- The Windows Server 2016 Core Network Companion Guide [Deploy Server Certificates for 802.1X Wired and Wireless Deployments](#) in HTML format in the Technical Library.

Public CA

You can purchase server certificates from a public CA, such as VeriSign, that client computers already trust.

A client computer trusts a CA when the CA certificate is installed in the Trusted Root Certification Authorities certificate store. By default, computers running Windows have multiple public CA certificates installed in their Trusted Root Certification Authorities certificate store.

It is recommended that you review the design and deployment guides for each of the technologies that are used in this deployment scenario. These guides can help you determine whether this deployment scenario provides the services and configuration that you need for your organization's network.

Requirements

Following are the requirements for deploying a wireless access infrastructure by using the scenario documented in this guide:

- Before deploying this scenario, you must first purchase 802.1X-capable wireless access points to provide wireless coverage in the desired locations at your site. The planning section of this guide assists in determining the features your APs must support.
- Active Directory Domain Services (AD DS) is installed, as are the other required network technologies, according to the instructions in the Windows Server 2016 Core Network Guide.
- AD CS is deployed, and server certificates are enrolled to NPSs. These certificates are required when you deploy the PEAP-MS-CHAP v2 certificate-based authentication method that is used in this guide.
- A member of your organization is familiar with the IEEE 802.11 standards that are supported by your wireless APs and the wireless network adapters that are installed in the client computers and devices on your network. For example, someone in your organization is familiar with radio frequency types, 802.11 wireless authentication (WPA2 or WPA), and ciphers (AES or TKIP).

What this guide does not provide

Following are some items this guide does not provide:

Comprehensive guidance for selecting 802.1X-capable wireless access points

Because many differences exist between brands and models of 802.1X-capable wireless APs, this guide does not

provide detailed information about:

- Determining which brand or model of wireless AP is best suited to your needs.
- The physical deployment of wireless APs on your network.
- Advanced wireless AP configuration, such as for wireless virtual Local Area Networks (VLANs).
- Instructions on how to configure wireless AP vendor-specific attributes in NPS.

Additionally, terminology and names for settings vary between wireless AP brands and models, and might not match the generic setting names that are used in this guide. For wireless AP configuration details, you must review the product documentation provided by the manufacturer of your wireless APs.

Instructions for deploying NPS certificates

There are two alternatives for deploying NPS certificates. This guide does not provide comprehensive guidance to help you determine which alternative will best meet your needs. In general, however, the choices you face are:

- Purchasing certificates from a public CA, such as VeriSign, that are already trusted by Windows-based clients. This option is typically recommended for smaller networks.
- Deploying a Public Key Infrastructure (PKI) on your network by using AD CS. This is recommended for most networks, and the instructions for how to deploy server certificates with AD CS are available in the previously mentioned deployment guide.

NPS network policies and other NPS settings

Except for the configuration settings made when you run the **Configure 802.1X** wizard, as documented in this guide, this guide does not provide detailed information for manually configuring NPS conditions, constraints or other NPS settings.

DHCP

This deployment guide does not provide information about designing or deploying DHCP subnets for wireless LANs.

Technology overviews

Following are technology overviews for deploying wireless access:

IEEE 802.1X

The IEEE 802.1X standard defines the port-based network access control that is used to provide authenticated network access to Ethernet networks. This port-based network access control uses the physical characteristics of the switched LAN infrastructure to authenticate devices attached to a LAN port. Access to the port can be denied if the authentication process fails. Although this standard was designed for wired Ethernet networks, it has been adapted for use on 802.11 wireless LANs.

802.1X-capable wireless access points (APs)

This scenario requires the deployment of one or more 802.1X-capable wireless APs that are compatible with the Remote Authentication Dial-In User Service (RADIUS) protocol.

802.1X and RADIUS-compliant APs, when deployed in a RADIUS infrastructure with a RADIUS server such as an NPS, are called *RADIUS clients*.

Wireless clients

This guide provides comprehensive configuration details to supply 802.1X authenticated access for domain-member users who connect to the network with wireless client computers running Windows 10, Windows 8.1, and Windows 8. Computers must be joined to the domain in order to successfully establish authenticated access.

NOTE

You can also use computers that are running Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012 as wireless clients.

Support for IEEE 802.11 Standards

Supported Windows and Windows Server operating systems provide built-in support for 802.11 wireless networking. In these operating systems, an installed 802.11 wireless network adapter appears as a wireless network connection in Network and Sharing Center.

Although there is built-in support for 802.11 wireless networking, the wireless components of Windows are dependent upon the following:

- The capabilities of the wireless network adapter. The installed wireless network adapter must support the wireless LAN or wireless security standards that you require. For example, if the wireless network adapter does not support Wi-Fi Protected Access (WPA), you cannot enable or configure WPA security options.
- The capabilities of the wireless network adapter driver. To allow you to configure wireless network options, the driver for the wireless network adapter must support the reporting of all of its capabilities to Windows. Verify that the driver for your wireless network adapter is written for the capabilities of your operating system. Also ensure that the driver is the most current version by checking Microsoft Update or the Web site of the wireless network adapter vendor.

The following table shows the transmission rates and frequencies for common IEEE 802.11 wireless standards.

STANDARDS	FREQUENCIES	BIT TRANSMISSION RATES	USAGE
802.11	S-Band Industrial, Scientific, and Medical (ISM) frequency range (2.4 to 2.5 GHz)	2 megabits per second (Mbps)	Obsolete. Not commonly used.
802.11b	S-Band ISM	11 Mbps	Commonly used.
802.11a	C-Band ISM (5.725 to 5.875 GHz)	54 Mbps	Not commonly used due to expense and limited range.
802.11g	S-Band ISM	54 Mbps	Widely used. 802.11g devices are compatible with 802.11b devices.
802.11n \2.4 and 5.0 GHz	C-Band and S-Band ISM	250 Mbps	Devices based on the pre-ratification IEEE 802.11n standard became available in August 2007. Many 802.11n devices are compatible with 802.11a, b, and g devices.
802.11ac	5 GHz	6.93 Gbps	802.11ac, approved by the IEEE in 2014, is more scalable and faster than 802.11n, and is deployed where APs and wireless clients both support it.

Wireless network security methods

Wireless network security methods is an informal grouping of wireless authentication (sometimes referred to as

wireless security) and wireless security encryption. Wireless authentication and encryption are used in pairs to prevent unauthorized users from accessing the wireless network, and to protect wireless transmissions.

When configuring wireless security settings in the Wireless Network Policies of Group Policy, there are multiple combinations to choose from. However, only the WPA2-Enterprise, WPA-Enterprise, and Open with 802.1X authentication standards are supported for 802.1X Authenticated wireless deployments.

NOTE

While configuring Wireless Network Policies, you must select **WPA2-Enterprise**, **WPA-Enterprise**, or **Open with 802.1X** in order to gain access to the EAP settings that are required for 802.1X authenticated wireless deployments.

Wireless authentication

This guide recommends the use of the following wireless authentication standards for 802.1X authenticated wireless deployments.

Wi-Fi Protected Access – Enterprise (WPA-Enterprise) WPA is an interim standard developed by the WiFi Alliance to comply with the 802.11 wireless security protocol. The WPA protocol was developed in response to a number of severe flaws that were discovered in the preceding Wired Equivalent Privacy (WEP) protocol.

WPA-Enterprise provides improved security over WEP by:

1. Requiring authentication that uses the 802.1X EAP framework as part of the infrastructure that ensures centralized mutual authentication and dynamic key management
2. Enhancing the Integrity Check Value (ICV) with a Message Integrity Check (MIC), to protect the header and payload
3. Implementing a frame counter to discourage replay attacks

Wi-Fi Protected Access 2 – Enterprise (WPA2-Enterprise) Like the WPA-Enterprise standard, WPA2-Enterprise uses the 802.1X and EAP framework. WPA2-Enterprise provides stronger data protection for multiple users and large managed networks. WPA2-Enterprise is a robust protocol that is designed to prevent unauthorized network access by verifying network users through an authentication server.

Wireless security encryption

Wireless security encryption is used to protect the wireless transmissions that are sent between the wireless client and the wireless AP. Wireless security encryption is used in conjunction with the selected network security authentication method. By default, computers running Windows 10, Windows 8.1, and Windows 8 support two encryption standards:

1. **Temporal Key Integrity Protocol** (TKIP) is an older encryption protocol that was originally designed to provide more secure wireless encryption than what was provided by the inherently weak Wired Equivalent Privacy (WEP) protocol. TKIP was designed by the IEEE 802.11i task group and the Wi-Fi Alliance to replace WEP without requiring the replacement of legacy hardware. TKIP is a suite of algorithms that encapsulates the WEP payload, and allows users of legacy WiFi equipment to upgrade to TKIP without replacing hardware. Like WEP, TKIP uses the RC4 stream encryption algorithm as its basis. The new protocol, however, encrypts each data packet with a unique encryption key, and the keys are much stronger than those by WEP. Although TKIP is useful for upgrading security on older devices that were designed to use only WEP, it does not address all of the security issues facing wireless LANs, and in most cases is not sufficiently robust to protect sensitive government or corporate data transmissions.
2. **Advanced Encryption Standard** (AES) is the preferred encryption protocol for the encryption of commercial and government data. AES offers a higher level of wireless transmission security than either TKIP or WEP. Unlike TKIP and WEP, AES requires wireless hardware that supports the AES standard. AES is a symmetric-key encryption standard that uses three block ciphers, AES-128, AES-192 and AES-256.

In Windows Server 2016, the following AES-based wireless encryption methods are available for configuration in wireless profile properties when you select an authentication method of WPA2-Enterprise, which is recommended.

1. **AES-CCMP**. Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP) implements the 802.11i standard and is designed for higher security encryption than that provided by WEP, and uses 128 bit AES encryption keys.
2. **AES-GCMP**. Galois Counter Mode Protocol (GCMP) is supported by 802.11ac, is more efficient than AES-CCMP and provides better performance for wireless clients. GCMP uses 256 bit AES encryption keys.

IMPORTANT

Wired Equivalency Privacy (WEP) was the original wireless security standard that was used to encrypt network traffic. You should not deploy WEP on your network because there are well-known vulnerabilities in this outdated form of security.

Active Directory Domain Services (AD DS)

AD DS provides a distributed database that stores and manages information about network resources and application-specific data from directory-enabled applications. Administrators can use AD DS to organize elements of a network, such as users, computers, and other devices, into a hierarchical containment structure. The hierarchical containment structure includes the Active Directory forest, domains in the forest, and organizational units (OUs) in each domain. A server that is running AD DS is called a *domain controller*.

AD DS contains the user accounts, computer accounts, and account properties that are required by IEEE 802.1X and PEAP-MS-CHAP v2 to authenticate user credentials and to evaluate authorization for wireless connections.

Active Directory Users and Computers

Active Directory Users and Computers is a component of AD DS that contains accounts that represent physical entities, such as a computer, a person, or a security group. A *security group* is a collection of user or computer accounts that administrators can manage as a single unit. User and computer accounts that belong to a particular group are referred to as *group members*.

Group Policy Management

Group Policy Management enables directory-based change and configuration management of user and computer settings, including security and user information. You use Group Policy to define configurations for groups of users and computers. With Group Policy, you can specify settings for registry entries, security, software installation, scripts, folder redirection, remote installation services, and Internet Explorer maintenance. The Group Policy settings that you create are contained in a Group Policy object (GPO). By associating a GPO with selected Active Directory system containers — sites, domains, and OUs — you can apply the GPO's settings to the users and computers in those Active Directory containers. To manage Group Policy objects across an enterprise, you can use the Group Policy Management Editor Microsoft Management Console (MMC).

This guide provides detailed instructions about how to specify settings in the Wireless Network (IEEE 802.11) Policies extension of Group Policy Management. The Wireless Network (IEEE 802.11) Policies configure domain-member wireless client computers with the necessary connectivity and wireless settings for 802.1X authenticated wireless access.

Server certificates

This deployment scenario requires server certificates for each NPS that performs 802.1X authentication.

A server certificate is a digital document that is commonly used for authentication and to secure information on open networks. A certificate securely binds a public key to the entity that holds the corresponding private key. Certificates are digitally signed by the issuing CA, and they can be issued for a user, a computer, or a service.

A certification authority (CA) is an entity responsible for establishing and vouching for the authenticity of public keys belonging to subjects (usually users or computers) or other CAs. Activities of a certification authority can

include binding public keys to distinguished names through signed certificates, managing certificate serial numbers, and revoking certificates.

Active Directory Certificate Services (AD CS) is a server role that issues certificates as a network CA. An AD CS certificate infrastructure, also known as a *public key infrastructure (PKI)*, provides customizable services for issuing and managing certificates for the enterprise.

EAP, PEAP, and PEAP-MS-CHAP v2

Extensible Authentication Protocol (EAP) extends Point-to-Point Protocol (PPP) by allowing additional authentication methods that use credential and information exchanges of arbitrary lengths. With EAP authentication, both the network access client and the authenticator (such as the NPS) must support the same EAP type for successful authentication to occur. Windows Server 2016 includes an EAP infrastructure, supports two EAP types, and the ability to pass EAP messages to NPSs. By using EAP, you can support additional authentication schemes, known as *EAP types*. The EAP types that are supported by Windows Server 2016 are:

- Transport Layer Security (TLS)
- Microsoft Challenge Handshake Authentication Protocol version 2 (MS-CHAP v2)

IMPORTANT

Strong EAP types (such as those that are based on certificates) offer better security against brute-force attacks, dictionary attacks, and password guessing attacks than password-based authentication protocols (such as CHAP or MS-CHAP version 1).

Protected EAP (PEAP) uses TLS to create an encrypted channel between an authenticating PEAP client, such as a wireless computer, and a PEAP authenticator, such as an NPS or other RADIUS servers. PEAP does not specify an authentication method, but it provides additional security for other EAP authentication protocols (such as EAP-MS-CHAP v2) that can operate through the TLS encrypted channel provided by PEAP. PEAP is used as an authentication method for access clients that are connecting to your organization's network through the following types of network access servers (NASs):

- 802.1X-capable wireless access points
- 802.1X-capable authenticating switches
- Computers running Windows Server 2016 and the Remote Access Service (RAS) that are configured as virtual private network (VPN) servers, DirectAccess Servers, or both
- Computers running Windows Server 2016 and Remote Desktop Services

PEAP-MS-CHAP v2 is easier to deploy than EAP-TLS because user authentication is performed by using password-based credentials (user name and password), instead of certificates or smart cards. Only NPS or other RADIUS servers are required to have a certificate. The NPS certificate is used by the NPS during the authentication process to prove its identity to PEAP clients.

This guide provides instructions to configure your wireless clients and your NPS(s) to use PEAP-MS-CHAP v2 for 802.1X authenticated access.

Network Policy Server

Network Policy Server (NPS) allows you to centrally configure and manage network policies by using Remote Authentication Dial-In User Service (RADIUS) server and RADIUS proxy. NPS is required when you deploy 802.1X wireless access.

When you configure your 802.1X wireless access points as RADIUS clients in NPS, NPS processes the connection requests sent by the APs. During connection request processing, NPS performs authentication and authorization. Authentication determines whether the client has presented valid credentials. If NPS successfully authenticates the

requesting client, then NPS determines whether the client is authorized to make the requested connection, and either allows or denies the connection. This is explained in more detail as follows:

Authentication

Successful mutual PEAP-MS-CHAP v2 authentication has two main parts:

1. The client authenticates the NPS. During this phase of mutual authentication, the NPS sends its server certificate to the client computer so that the client can verify the NPS's identity with the certificate. To successfully authenticate the NPS, the client computer must trust the CA that issued the NPS certificate. The client trusts this CA when the CA's certificate is present in the Trusted Root Certification Authorities certificate store on the client computer.

If you deploy your own private CA, the CA certificate is automatically installed in the Trusted Root Certification Authorities certificate store for the Current User and for the Local Computer when Group Policy is refreshed on the domain member client computer. If you decide to deploy server certificates from a public CA, ensure that the public CA certificate is already in the Trusted Root Certification Authorities certificate store.

2. The NPS authenticates the user. After the client successfully authenticates the NPS, the client sends the user's password-based credentials to the NPS, which verifies the user's credentials against the user accounts database in Active Directory Domain Services (AD DS).

If the credentials are valid and authentication succeeds, the NPS begins the authorization phase of processing the connection request. If the credentials are not valid and authentication fails, NPS sends an Access Reject message and the connection request is denied.

Authorization

The server running NPS performs authorization as follows:

1. NPS checks for restrictions in the user or computer account dial-in properties in AD DS. Every user and computer account in Active Directory Users and Computers includes multiple properties, including those found on the **Dial-in** tab. On this tab, in **Network Access Permission**, if the value is **Allow access**, the user or computer is authorized to connect to the network. If the value is **Deny access**, the user or computer is not authorized to connect to the network. If the value is **Control access through NPS Network Policy**, NPS evaluates the configured network policies to determine whether the user or computer is authorized to connect to the network.
2. NPS then processes its network policies to find a policy that matches the connection request. If a matching policy is found, NPS either grants or denies the connection based on that policy's configuration.

If both authentication and authorization are successful, and if the matching network policy grants access, NPS grants access to the network, and the user and computer can connect to network resources for which they have permissions.

NOTE

To deploy wireless access, you must configure NPS policies. This guide provides instructions to use the **Configure 802.1X wizard** in NPS to create NPS policies for 802.1X authenticated wireless access.

Bootstrap profiles

In 802.1X-authenticated wireless networks, wireless clients must provide security credentials that are authenticated by a RADIUS server in order to connect to the network. For Protected EAP [PEAP]-Microsoft Challenge Handshake Authentication Protocol version 2 [MS-CHAP v2], the security credentials are a user name and password. For EAP-Transport Layer Security [TLS] or PEAP-TLS, the security credentials are certificates, such as client user and computer certificates or smart cards.

When connecting to a network that is configured to perform PEAP-MS-CHAP v2, PEAP-TLS, or EAP-TLS authentication, by default, Windows wireless clients must also validate a computer certificate that is sent by the RADIUS server. The computer certificate that is sent by the RADIUS server for every authentication session is commonly referred to as a server certificate.

As mentioned previously, you can issue your RADIUS servers their server certificate in one of two ways: from a commercial CA (such as VeriSign, Inc.), or from a private CA that you deploy on your network. If the RADIUS server sends a computer certificate that was issued by a commercial CA that already has a root certificate installed in the client's Trusted Root Certification Authorities certificate store, then the wireless client can validate the RADIUS server's computer certificate, regardless of whether the wireless client has joined the Active Directory domain. In this case the wireless client can connect to the wireless network, and then you can join the computer to the domain.

NOTE

The behavior requiring the client to validate the server certificate can be disabled, but disabling server certificate validation is not recommended in production environments.

Wireless bootstrap profiles are temporary profiles that are configured in such a way as to enable wireless client users to connect to the 802.1X-authenticated wireless network before the computer is joined to the domain, and/or before the user has successfully logged on to the domain by using a given wireless computer for the first time. This section summarizes what problem is encountered when trying to join a wireless computer to the domain, or for a user to use a domain-joined wireless computer for the first time to log on to the domain.

For deployments in which the user or IT administrator cannot physically connect a computer to the wired Ethernet network to join the computer to the domain, and the computer does not have the necessary issuing root CA certificate installed in its **Trusted Root Certification Authorities** certificate store, you can configure wireless clients with a temporary wireless connection profile, called a *bootstrap profile*, to connect to the wireless network.

A *bootstrap profile* removes the requirement to validate the RADIUS server's computer certificate. This temporary configuration enables the wireless user to join the computer to the domain, at which time the Wireless Network (IEEE 802.11) Policies are applied and the appropriate root CA certificate is automatically installed on the computer.

When Group Policy is applied, one or more wireless connection profiles that enforce the requirement for mutual authentication are applied on the computer; the bootstrap profile is no longer required and is removed. After joining the computer to the domain and restarting the computer, the user can use a wireless connection to log on to the domain.

For an overview of the wireless access deployment process using these technologies, see [Wireless Access Deployment Overview](#).

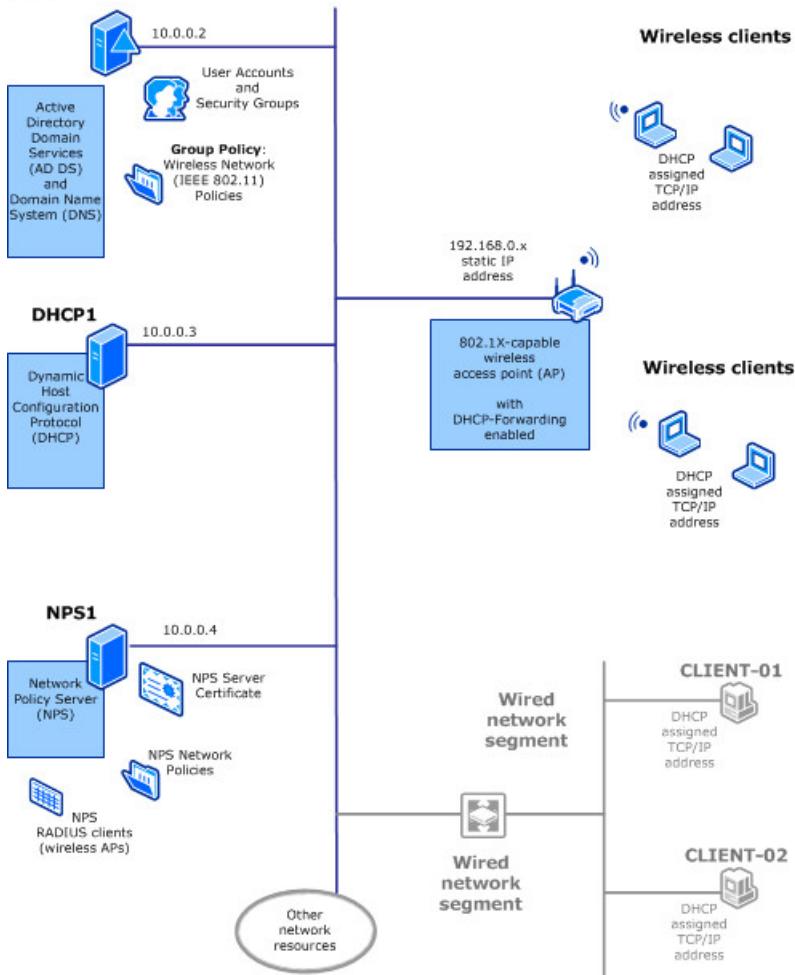
Wireless Access Deployment Overview

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

The following illustration shows the components that are required to deploy 802.1X authenticated wireless access with PEAP-MS-CHAP v2.

DC1



Wireless access deployment components

The following infrastructure is required for this wireless access deployment:

802.1X-capable Wireless access points

After the required network infrastructure services supporting your wireless local area network are in place, you can begin the design process for the location of the wireless APs. The wireless AP deployment design process involves these steps:

- Identify the areas of coverage for wireless users. While identifying the areas of coverage, be sure to identify whether you want to provide wireless service outside the building, and if so, determine specifically where those external areas are.
- Determine how many wireless APs to deploy to ensure adequate coverage.
- Determine where to place wireless APs.

- Select the channel frequencies for wireless APs.

Active Directory Domain Services

The following elements of AD DS are required for wireless access deployment.

Users and Computers

Use the Active Directory Users and Computers snap-in to create and manage user accounts, and to create a wireless security group that includes each domain member to whom you want to grant wireless access.

Wireless Network (IEEE 802.11) Policies

You can use the Wireless Network (IEEE 802.11) Policies extension of Group Policy Management to configure policies that are applied to wireless computers when they attempt to access the network.

In Group Policy Management Editor, when you right-click **Wireless Network (IEEE 802.11) Policies**, you have the following two options for the type of wireless policy that you create.

- **Create a New Wireless Network Policy for Windows Vista and Later Releases**
- **Create a New Windows XP Policy**

TIP

When configuring a new wireless network policy, you have the option to change the name and description of the policy. If you change the name of the policy, the change is reflected in the **Details** pane of Group Policy Management Editor and on the title bar of the wireless network policy dialog box. Regardless of how you rename your policies, the New XP Wireless Policy will always be listed in Group Policy Management Editor with the **Type** displaying **XP**. Other policies are listed with the **Type** showing **Vista and Later Releases**.

The Wireless Network Policy for Windows Vista and Later Releases enables you to configure, prioritize, and manage multiple wireless profiles. A wireless profile is a collection of connectivity and security settings that are used to connect to a specific wireless network. When Group Policy is updated on your wireless client computers, the profiles you create in the Wireless Network Policy are automatically added to the configuration on your wireless client computers to which the Wireless Network Policy applies.

Allowing connections to multiple wireless networks

If you have wireless clients that are moved across physical locations in your organization, such as between a main office and a branch office, you might want computers to connect to more than one wireless network. In this situation, you can configure a wireless profile that contains the specific connectivity and security settings for each network.

For example, assume your company has one wireless network for the main corporate office, with a service set identifier (SSID) WlanCorp.

Your branch office also has a wireless network to which you also want to connect. The branch office has the SSID configured as WlanBranch.

In this scenario, you can configure a profile for each network, and computers or other devices that are used at both the corporate office and branch office can connect to either of the wireless networks when they are physically in range of a network's coverage area.

Mixed-mode wireless networks

Alternately, assume your network has a mixture of wireless computers and devices that support different security standards. Perhaps some older computers have wireless adapters that can only use WPA-Enterprise, while newer devices can use the stronger WPA2-Enterprise standard.

You can create two different profiles that use the same SSID and nearly identical connectivity and security settings.

In one profile, you can set the wireless authentication to WPA2-Enterprise with AES, and in the other profile you

can specify WPA-Enterprise with TKIP.

This is commonly known as a mixed-mode deployment, and it allows computers of different types and wireless capabilities to share the same wireless network.

Network Policy Server (NPS)

NPS enables you to create and enforce network access policies for connection request authentication and authorization.

When you use NPS as a RADIUS server, you configure network access servers, such as wireless access points, as RADIUS clients in NPS. You also configure the network policies that NPS uses to authenticate access clients and authorize their connection requests.

Wireless client computers

For the purpose of this guide, wireless client computers are computers and other devices that are equipped with IEEE 802.11 wireless network adapters and that are running Windows client or Windows Server operating systems.

Server computers as wireless clients

By default, the functionality for 802.11 wireless is disabled on computers that are running Windows Server.

To enable wireless connectivity on computers running server operating systems, you must install and enable the Wireless LAN (WLAN) Service feature by using either Windows PowerShell or the Add Roles and Features Wizard in Server Manager.

When you install the **Wireless LAN Service** feature, the new service **WLAN AutoConfig** is installed in **Services**. When installation is complete, you must restart the server.

After the server is restarted, you can access WLAN AutoConfig when you click **Start**, **Windows Administrative Tools**, and **Services**.

After install and server restart, the WLAN AutoConfig service is in a stopped state with a startup type of **Automatic**. To start the service, double-click **WLAN AutoConfig**. On the **General** tab, click **Start**, and then click **OK**.

The WLAN AutoConfig service enumerates wireless adapters and manages both wireless connections and the wireless profiles that contain settings that are required to configure the server to connect to a wireless network.

For an overview of wireless access deployment, see [Wireless Access Deployment Process](#).

Wireless Access Deployment Process

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

The process that you use to deploy wireless access occurs in these stages:

Stage 1 – AP Deployment

Plan, deploy, and configure your APs for wireless client connectivity and for use with NPS. Depending on your preference and network dependencies, you can either pre-configure settings on your wireless APs prior to installing them on your network, or you can configure them remotely after installation.

Stage 2 – AD DS Group Configuration

In AD DS, you must create one or more wireless users security groups.

Next, identify the users who are allowed wireless access to the network.

Finally, add the users to the appropriate wireless users security groups that you created.

NOTE

By default, the **Network Access Permission** setting in user account dial-in properties is configured with the setting **Control access through NPS Network Policy**. Unless you have specific reasons to change this setting, it is recommended that you keep the default. This allows you to control network access through the network policies that you configure in NPS.

Stage 3 – Group Policy Configuration

Configure the Wireless Network (IEEE 802.11) Policies extension of Group Policy by using the Group Policy Management Editor Microsoft Management Console (MMC).

To configure domain-member computers using the settings in the wireless network policies, you must apply Group Policy. When a computer is first joined to the domain, Group Policy is automatically applied. If changes are made to Group Policy, the new settings are automatically applied:

- By Group Policy at pre-determined intervals
- If a domain user logs off and then back on to the network
- By restarting the client computer and logging on to the domain

You can also force Group Policy to refresh while logged on to a computer by running the command **gpupdate** at the command prompt.

Stage 4 – NPS configuration

Use a configuration wizard in NPS to add wireless access points as RADIUS clients, and to create the network policies that NPS uses when processing connection requests.

When using the wizard to create the network policies, specify PEAP as the EAP type, and the wireless users security group that was created in the second stage.

Stage 5 – Deploy wireless clients

Use client computers to connect to the network.

For domain member computers that can log on to the wired LAN, the necessary wireless configuration settings are automatically applied when Group Policy is refreshed.

If you have enabled the setting in Wireless Network (IEEE 802.11) Policies to connect automatically when the computer is within broadcast range of the wireless network, your wireless, domain-joined computers will then automatically attempt to connect to the wireless LAN.

To connect to the wireless network, users need only supply their domain user name and password credentials when prompted by Windows.

To plan your wireless access deployment, see [Wireless Access Deployment Planning](#).

Wireless Access Deployment Planning

9/1/2018 • 15 minutes to read • [Edit Online](#)

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

Before you deploy wireless access, you must plan the following items:

- Installation of wireless access points (APs) on your network
- Wireless client configuration and access

The following sections provide details on these planning steps.

Planning wireless AP installations

When you design your wireless network access solution, you must do the following:

1. Determine what standards your wireless APs must support
2. Determine the coverage areas where you want to provide wireless service
3. Determine where you want to locate wireless APs

Additionally, you must plan an IP address scheme for your wireless AP's and wireless clients. See the section **Plan the configuration of wireless AP's in NPS** below for related information.

Verify wireless AP support for standards

For the purposes of consistency and ease of deployment and AP management, it is recommended that you deploy wireless APs of the same brand and model.

The wireless APs that you deploy must support the following:

- **IEEE 802.1X**
- **RADIUS authentication**
- **Wireless Authentication and Cipher.** Listed in order of most to least preferred:
 1. WPA2-Enterprise with AES
 2. WPA2-Enterprise with TKIP
 3. WPA-Enterprise with AES
 4. WPA-Enterprise with TKIP

NOTE

To deploy WPA2, you must use wireless network adapters and wireless APs that also support WPA2. Otherwise, use WPA-Enterprise.

In addition, to provide enhanced security for the network, the wireless APs must support the following security options:

- **DHCP filtering.** The wireless AP must filter on IP ports to prevent the transmission of DHCP broadcast messages in those cases in which the wireless client is configured as a DHCP server. The wireless AP must

block the client from sending IP packets from UDP port 68 to the network.

- **DNS filtering.** The wireless AP must filter on IP ports to prevent a client from performing as a DNS server. The wireless AP must block the client from sending IP packets from TCP or UDP port 53 to the network.
- **Client isolation** If your wireless access point provides client isolation capabilities, you should enable the feature to prevent possible Address Resolution Protocol (ARP) spoofing exploits.

Identify areas of coverage for wireless users

Use architectural drawings of each floor for each building to identify the areas where you want to provide wireless coverage. For example, identify the appropriate offices, conferences rooms, lobbies, cafeterias, or courtyards.

On the drawings, indicate any devices that interfere with the wireless signals, such as medical equipment, wireless video cameras, cordless telephones that operate in the 2.4 through 2.5 GHz Industrial, Scientific and Medical (ISM) range, and Bluetooth-enabled devices.

On the drawing, mark aspects of the building that might interfere with wireless signals; metal objects used in the construction of a building can affect the wireless signal. For example, the following common objects can interfere with signal propagation: Elevators, heating and air-conditioning ducts, and concrete support girders.

Refer to your AP manufacturer for information about sources that might cause wireless AP radio frequency attenuation. Most APs provide testing software that you can use to check for signal strength, error rate, and data throughput.

Determine where to install wireless APs

On the architectural drawings, locate your wireless APs close enough together to provide ample wireless coverage but far enough apart that they do not interfere with each other.

The necessary distance between APs depends upon the type of AP and AP antenna, aspects of the building that block wireless signals, and other sources of interference. You can mark wireless AP placements so that each wireless AP is not more than 300 feet from any adjacent wireless AP. See the wireless AP manufacturer's documentation for AP specifications and guidelines for placement.

Temporarily install wireless APs in the locations specified on your architectural drawings. Then, using a laptop equipped with an 802.11 wireless adapter and the site survey software that is commonly supplied with wireless adapters, determine the signal strength within each coverage area.

In coverage areas where signal strength is low, position the AP to improve signal strength for the coverage area, install additional wireless APs to provide the necessary coverage, relocate or remove sources of signal interference.

Update your architectural drawings to indicate the final placement of all wireless APs. Having an accurate AP placement map will assist later during troubleshooting operations or when you want to upgrade or replace APs.

Plan wireless AP and NPS RADIUS Client configuration

You can use NPS to configure wireless APs individually or in groups.

If you are deploying a large wireless network that includes many APs, it is much easier to configure APs in groups. To add the APs as RADIUS client groups in NPS, you must configure the APs with these properties.

- The wireless APs are configured with IP addresses from the same IP address range.
- The wireless APs are all configured with the same shared secret.

Plan the use of PEAP Fast Reconnect

In an 802.1X infrastructure, wireless access points are configured as RADIUS clients to RADIUS servers. When PEAP fast reconnect is deployed, a wireless client that roams between two or more access points is not required to be authenticated with each new association.

PEAP fast reconnect reduces the response time for authentication between client and authenticator because the

authentication request is forwarded from the new access point to the NPS that originally performed authentication and authorization for the client connection request.

Because both the PEAP client and NPS both use previously cached Transport Layer Security (TLS) connection properties (the collection of which is named the TLS handle), the NPS can quickly determine that the client is authorized for a reconnect.

IMPORTANT

For fast reconnect to function correctly, the APs must be configured as RADIUS clients of the same NPS.

If the original NPS becomes unavailable, or if the client moves to an access point that is configured as a RADIUS client to a different RADIUS server, full authentication must occur between the client and the new authenticator.

Wireless AP configuration

The following list summarizes items commonly configured on 802.1X-capable wireless APs:

NOTE

The item names can vary by brand and model and might be different from those in the following list. See your wireless AP documentation for configuration-specific details.

- **Service set identifier (SSID).** This is the name of the wireless network (for example, ExampleWlan), and the name that is advertised to wireless clients. To reduce confusion, the SSID that you choose to advertise should not match the SSID that is broadcast by any wireless networks that are within reception range of your wireless network.

In cases in which multiple wireless APs are deployed as part of the same wireless network, configure each wireless AP with the same SSID. In cases in which multiple wireless APs are deployed as part of the same wireless network, configure each wireless AP with the same SSID.

In cases where you have a need to deploy different wireless networks to meet specific business needs, your wireless AP's on one network should broadcast a different SSID than the SSID your other network(s). For example, if you need a separate wireless network for your employees and guests, you could configure your wireless APs for the business network with the SSID set to broadcast **ExampleWLAN**. For your guest network, you could then set each wireless AP's SSID to broadcast **GuestWLAN**. In this way your employees and guests can connect to the intended network without unnecessary confusion.

TIP

Some wireless AP's have the ability to broadcast multiple SSID's to accommodate multi-network deployments. Wireless AP's that can broadcast multiple SSID's can reduce deployment and operational maintenance costs.

- **Wireless authentication and encryption.**

Wireless authentication is the security authentication that is used when the wireless client associates with a wireless access point.

Wireless encryption is the security encryption cipher that is used with wireless authentication to protect the communications that are sent between the wireless AP and the wireless client.

- **Wireless AP IP address (static).** On each wireless AP, configure a unique static IP address. If the subnet is serviced by a DHCP server, ensure that all AP IP addresses fall within a DHCP exclusion range so that the DHCP server does not try to issue the same IP address to another computer or device. Exclusion ranges are documented in the procedure "To create and activate a new DHCP Scope" in the [Core Network Guide](#). If

you are planning to configure APs as RADIUS clients by group in NPS, each AP in the group must have an IP address from the same IP address range.

- **DNS name.** Some wireless APs can be configured with a DNS name. Configure each wireless AP with a unique name. For example, if you have a deployed wireless APs in a multi-story building, you might name the first three wireless APs that are deployed on the third floor AP3-01, AP3-02, and AP3-03.
- **Wireless AP subnet mask.** Configure the mask to designate which portion of the IP address is the network ID and which portion of the IP address is the host.
- **AP DHCP service.** If your wireless AP has a built-in DHCP service, disable it.
- **RADIUS shared secret.** Use a unique RADIUS shared secret for each wireless AP unless you are planning to configure NPS RADIUS clients in groups - in which circumstance you must configure all of the APs in the group with the same shared secret. Shared secrets should be a random sequence of at least 22 characters long, with both uppercase and lowercase letters, numbers, and punctuation. To ensure randomness, you can use a random character generation program to create your shared secrets. It is recommended that you record the shared secret for each wireless AP and store it in a secure location, such as an office safe. When you configure RADIUS clients in the NPS console you will create a virtual version of each AP. The shared secret that you configure on each virtual AP in NPS must match the shared secret on the actual, physical AP.
- **RADIUS server IP address.** Type the IP address of the NPS that you want to use to authenticate and authorize connection requests to this access point.
- **UDP port(s).** By default, NPS uses UDP ports 1812 and 1645 for RADIUS authentication messages and UDP ports 1813 and 1646 for RADIUS accounting messages. It is recommended that you do not change the default RADIUS UDP ports settings.
- **VSA.** Some wireless APs require vendor-specific attributes (VSAs) to provide full wireless AP functionality.
- **DHCP filtering.** Configure wireless APs to block wireless clients from sending IP packets from UDP port 68 to the network. See the documentation for your wireless AP to configure DHCP filtering.
- **DNS filtering.** Configure wireless APs to block wireless clients from sending IP packets from TCP or UDP port 53 to the network. See the documentation for your wireless AP to configure DNS filtering.

Planning wireless client configuration and access

When planning the deployment of 802.1X-authenticated wireless access, you must consider several client-specific factors:

- **Planning support for multiple standards.**

Determine whether your wireless computers are all using the same version of Windows or whether they are a mixture of computers running different operating systems. If they are different, ensure that you understand any differences in standards supported by the operating systems.

Determine whether all of the wireless network adapters on all of the wireless client computers support the same wireless standards, or whether you need to support varying standards. For example, determine whether some network adapter hardware drivers support WPA2-Enterprise and AES, while others support only WPA-Enterprise and TKIP.

- **Planning client authentication mode.** Authentication modes define how Windows clients process domain credentials. You can select from the following three network authentication modes in the wireless network policies.

1. **User re-authentication.** This mode specifies that authentication is always performed by using

security credentials based on the computer's current state. When no users are logged on to the computer, authentication is performed by using the computer credentials. When a user is logged on to the computer, authentication is always performed by using the user credentials.

2. **Computer only.** Computer only mode specifies that authentication is always performed by using only the computer credentials.
 3. **User authentication.** User authentication mode specifies that authentication is only performed when the user is logged on to the computer. When there are no users logged on to the computer, authentication attempts are not performed.
- **Planning wireless restrictions.** Determine whether you want to provide all of your wireless users with the same level of access to your wireless network, or whether you want to restrict access for some of your wireless users. You can apply restrictions in NPS against specific groups of wireless users. For example, you can define specific days and hours that certain groups are permitted access to the wireless network.
 - **Planning methods for adding new wireless computers.** For wireless-capable computers that are joined to your domain before you deploy your wireless network, if the computer is connected to a segment of the wired network that is not protected by 802.1X, the wireless configuration settings are automatically applied after you configure Wireless Network (IEEE 802.11) Policies on the domain controller and after Group Policy is refreshed on the wireless client.

For computers that are not already joined to your domain, however, you must plan a method to apply the settings that are required for 802.1X-authenticated access. For example, determine whether you want to join the computer to the domain by using one of the following methods.

1. Connect the computer to a segment of the wired network that is not protected by 802.1X, then join the computer to the domain.
2. Provide your wireless users with the steps and settings that they require to add their own wireless bootstrap profile, which allows them to join the computer to the domain.
3. Assign IT staff to join wireless clients to the domain.

Planning support for multiple standards

The Wireless Network (IEEE 802.11) Policies extension in Group Policy provides a wide range of configuration options to support a variety of deployment options.

You can deploy wireless APs that are configured with the standards that you want to support, and then configure multiple wireless profiles in Wireless Network (IEEE 802.11) Policies, with each profile specifying one set of standards that you require.

For example, if your network has wireless computers that support WPA2-Enterprise and AES, other computers that support WPA-Enterprise and AES, and other computers that support only WPA-Enterprise and TKIP, you must determine whether you want to:

- Configure a single profile to support all of the wireless computers by using the weakest encryption method that all of your computers support - in this case, WPA-Enterprise and TKIP.
- Configure two profiles to provide the best possible security that is supported by each wireless computer. In this instance you would configure one profile that specifies the strongest encryption (WPA2-Enterprise and AES), and one profile that uses the weaker WPA-Enterprise and TKIP encryption. In this example, it is essential that you place the profile that uses WPA2-Enterprise and AES highest in the preference order. Computers that are not capable of using WPA2-Enterprise and AES will automatically skip to the next profile in the preference order and process the profile that specifies WPA-Enterprise and TKIP.

IMPORTANT

You must place the profile with the most secure standards higher in the ordered list of profiles, because connecting computers use the first profile that they are capable of using.

Planning restricted access to the wireless network

In many cases, you might want to provide wireless users with varying levels of access to the wireless network. For example, you might want to allow some users unrestricted access, any hour of the day, every day of the week. For other users, you might only want to allow access during core hours, Monday through Friday, and deny access on Saturday and Sunday.

This guide provides instructions to create an access environment that places all of your wireless users in a group with common access to wireless resources. You create one wireless users security group in the Active Directory Users and Computers snap-in, and then make every user for whom you want to grant wireless access a member of that group.

When you configure NPS network policies, you specify the wireless users security group as the object that NPS processes when determining authorization.

However, if your deployment requires support for varying levels of access you need only do the following:

1. Create more than one Wireless Users Security Group to create additional wireless security groups in Active Directory Users and Computers. For example, you can create a group that contains users who have full access, a group for those who only have access during regular working hours, and other groups that fit other criteria that match your requirements.
2. Add users to the appropriate security groups that you created.
3. Configure additional NPS network policies for each additional wireless security group, and configure the policies to apply the conditions and constraints that you require for each group.

Planning methods for adding new wireless computers

The preferred method to join new wireless computers to the domain and then log on to the domain is by using a wired connection to a segment of the LAN that has access to domain controllers, and is not protected by an 802.1X authenticating Ethernet switch.

In some cases, however, it might not be practical to use a wired connection to join computers to the domain, or, for a user to use a wired connection for their first log on attempt by using computers that are already joined to the domain.

To join a computer to the domain by using a wireless connection or for users to log on to the domain the first time by using a domain-joined computer and a wireless connection, wireless clients must first establish a connection to the wireless network on a segment that has access to the network domain controllers by using one of the following methods.

1. **A member of the IT staff joins a wireless computer to the domain, and then configures a Single Sign On bootstrap wireless profile.** With this method, an IT administrator connects the wireless computer to the wired Ethernet network, and then joins the computer to the domain. Then the administrator distributes the computer to the user. When the user starts the computer, the domain credentials that they manually specify for the user logon process are used to both establish a connection to the wireless network and log on to the domain.
2. **The user manually configures wireless computer with bootstrap wireless profile, and then joins the domain.** With this method, users manually configure their wireless computers with a bootstrap wireless profile based on instructions from an IT administrator. The bootstrap wireless profile allows users to establish a wireless connection, and then join the computer to the domain. After joining the computer to

the domain and restarting the computer, the user can log on to the domain by using a wireless connection and their domain account credentials.

To deploy wireless access, see [Wireless Access Deployment](#).

Wireless Access Deployment

9/1/2018 • 35 minutes to read • [Edit Online](#)

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

Follow these steps to deploy wireless access:

- [Deploy and Configure Wireless APs](#)
- [Create a Wireless Users Security Group](#)
- [Configure Wireless Network \(IEEE 802.11\) Policies](#)
- [Configure NPSs](#)
- [Join New Wireless Computers to the Domain](#)

Deploy and Configure Wireless APs

Follow these steps to deploy and configure your wireless APs:

- [Specify Wireless AP Channel Frequencies](#)
- [Configure Wireless APs](#)

NOTE

The procedures in this guide do not include instructions for cases in which the **User Account Control** dialog box opens to request your permission to continue. If this dialog box opens while you are performing the procedures in this guide, and if the dialog box was opened in response to your actions, click **Continue**.

Specify Wireless AP Channel Frequencies

When you deploy multiple wireless APs at a single geographical site, you must configure wireless APs that have overlapping signals to use unique channel frequencies to reduce interference between wireless APs.

You can use the following guidelines to assist you in choosing channel frequencies that do not conflict with other wireless networks at the geographical location of your wireless network.

- If there are other organizations that have offices in close proximity or in the same building as your organization, identify whether there are any wireless networks owned by those organizations. Find out both the placement and the assigned channel frequencies of their wireless AP's, because you need to assign different channel frequencies to your AP's and you need to determine the best location to install your AP's.
- Identify overlapping wireless signals on adjacent floors within your own organization. After identifying overlapping coverage areas outside and within your organization, assign channel frequencies for your wireless APs, ensuring that any two wireless APs with overlapping coverage are assigned different channel frequencies.

Configure Wireless APs

Use the following information along with the product documentation provided by the wireless AP manufacturer to configure your wireless APs.

This procedure enumerates items commonly configured on a wireless AP. The item names can vary by brand and

model and might be different from those in the following list. For specific details, see your wireless AP documentation.

To configure your wireless APs

- **SSID.** Specify the name of the wireless network(s) (for example, ExampleWLAN). This is the name that is advertised to wireless clients.
- **Encryption.** Specify WPA2-Enterprise (preferred) or WPA-Enterprise, and either AES (preferred) or TKIP encryption cipher, depending on which versions are supported by your wireless client computer network adapters.
- **Wireless AP IP address (static).** On each AP, configure a unique static IP address that falls within the exclusion range of the DHCP scope for the subnet. Using an address that is excluded from assignment by DHCP prevents the DHCP server from assigning the same IP address to a computer or other device.
- **Subnet mask.** Configure this to match the subnet mask settings of the LAN to which you have connected the wireless AP.
- **DNS name.** Some wireless APs can be configured with a DNS name. The DNS service on the network can resolve DNS names to an IP address. On each wireless AP that supports this feature, enter a unique name for DNS resolution.
- **DHCP service.** If your wireless AP has a built-in DHCP service, disable it.
- **RADIUS shared secret.** Use a unique RADIUS shared secret for each wireless AP unless you are planning to configure APs as RADIUS Clients in NPS by group. If you plan to configure APs by group in NPS, the shared secret must be the same for every member of the group. In addition, each shared secret you use should be a random sequence of at least 22 characters that mixes uppercase and lowercase letters, numbers, and punctuation. To ensure randomness, you can use a random character generator, such as the random character generator found in the NPS **Configure 802.1X** wizard, to create the shared secrets.

TIP

Record the shared secret for each wireless AP and store it in a secure location, such as an office safe. You must know the shared secret for each wireless AP when you configure RADIUS clients in the NPS.

- **RADIUS server IP address.** Type the IP address of the server running NPS.
- **UDP port(s).** By default, NPS uses UDP ports 1812 and 1645 for authentication messages and UDP ports 1813 and 1646 for accounting messages. It is recommended that you use these same UDP ports on your APs, but if you have a valid reason to use different ports, ensure that you not only configure the APs with the new port numbers but also reconfigure all of your NPSs to use the same port numbers as the APs. If the APs and the NPSs are not configured with the same UDP ports, NPS cannot receive or process connection requests from the APs, and all wireless connection attempts on the network will fail.
- **VSA.** Some wireless APs require vendor-specific attributes (VSAs) to provide full wireless AP functionality. VSAs are added in NPS network policy.
- **DHCP filtering.** Configure wireless APs to block wireless clients from sending IP packets from UDP port 68 to the network, as documented by the wireless AP manufacturer.
- **DNS filtering.** Configure wireless APs to block wireless clients from sending IP packets from TCP or UDP port 53 to the network, as documented by the wireless AP manufacturer.

Create Security Groups for Wireless Users

Follow these steps to create one or more wireless users security groups, and then add users to the appropriate

wireless users security group:

- [Create a Wireless Users Security Group](#)
- [Add Users to the Wireless Security Group](#)

Create a Wireless Users Security Group

You can use this procedure to create a wireless security group in the Active Directory Users and Computers Microsoft Management Console (MMC) snap-in.

Membership in **Domain Admins**, or equivalent, is the minimum required to perform this procedure.

To create a wireless users security group

1. Click **Start**, click **Administrative Tools**, and then click **Active Directory Users and Computers**. The Active Directory Users and Computers snap-in opens. If it is not already selected, click the node for your domain. For example, if your domain is example.com, click **example.com**.
2. In the details pane, right-click the folder in which you want to add a new group (for example, right-click **Users**), point to **New**, and then click **Group**.
3. In **New Object – Group**, in **Group name**, type the name of the new group. For example, type **Wireless Group**.
4. In **Group scope**, select one of the following options:
 - **Domain local**
 - **Global**
 - **Universal**
5. In **Group type**, select **Security**.
6. Click **OK**.

If you need more than one security group for wireless users, repeat these steps to create additional wireless users groups. Later you can create individual network policies in NPS to apply different conditions and constraints to each group, providing them with different access permissions and connectivity rules.

Add Users to the Wireless Users Security Group

You can use this procedure to add a user, computer, or group to your wireless security group in the Active Directory Users and Computers Microsoft Management Console (MMC) snap-in.

Membership in **Domain Admins**, or equivalent is the minimum required to perform this procedure.

To add users to the wireless security group

1. Click **Start**, click **Administrative Tools**, and then click **Active Directory Users and Computers**. The Active Directory Users and Computers MMC opens. If it is not already selected, click the node for your domain. For example, if your domain is example.com, click **example.com**.
2. In the details pane, double-click the folder that contains your wireless security group.
3. In the details pane, right-click the wireless security group, and then click **Properties**. The **Properties** dialog box for the security group opens.
4. On the **Members** tab, click **Add**, and then complete one of the following procedures to either add a computer or add a user or group.

To add a user or group

1. In **Enter the object names to select**, type the name of the user or group that you want to add, and then click **OK**.

2. To assign group membership to other users or groups, repeat step 1 of this procedure.

To add a computer

1. Click **Object Types**. The **Object Types** dialog box opens.
2. In **Object types**, select **Computers**, and then click **OK**.
3. In **Enter the object names to select**, type the name of the computer that you want to add, and then click **OK**.
4. To assign group membership to other computers, repeat steps 1-3 of this procedure.

Configure Wireless Network (IEEE 802.11) Policies

Follow these steps to configure Wireless Network (IEEE 802.11) Policies Group Policy extension:

- [Open or Add and Open a Group Policy Object](#)
- [Activate Default Wireless Network \(IEEE 802.11\) Policies](#)
- [Configure the New Wireless Network Policy](#)

Open or Add and Open a Group Policy Object

By default, the Group Policy Management feature is installed on computers running Windows Server 2016 when the Active Directory Domain Services (AD DS) server role is installed and the server is configured as a domain controller. The following procedure that describes how to open the Group Policy Management Console (GPMC) on your domain controller. The procedure then describes how to either open an existing domain-level Group Policy object (GPO) for editing, or create a new domain GPO and open it for editing.

Membership in **Domain Admins**, or equivalent, is the minimum required to perform this procedure.

To open or add and open a Group Policy object

1. On your domain controller, click **Start**, click **Windows Administrative Tools**, and then click **Group Policy Management**. The Group Policy Management Console opens.
2. In the left pane, double-click your forest. For example, double-click **Forest: example.com**.
3. In the left pane, double-click **Domains**, and then double-click the domain for which you want to manage a Group Policy object. For example, double-click **example.com**.
4. Do one of the following:
 - **To open an existing domain-level GPO for editing**, double click the domain that contains the Group Policy object that you want to manage, right-click the domain policy you want to manage, such as the Default Domain Policy, and then click **Edit**. **Group Policy Management Editor** opens.
 - **To create a new Group Policy object and open for editing**, right-click the domain for which you want to create a new Group Policy object, and then click **Create a GPO in this domain, and Link it here**.

In **New GPO**, in **Name**, type a name for the new Group Policy object, and then click **OK**.

Right-click your new Group Policy object, and then click **Edit**. **Group Policy Management Editor** opens.

In the next section you will use Group Policy Management Editor to create wireless policy.

Activate Default Wireless Network (IEEE 802.11) Policies

This procedure describes how to activate the default Wireless Network (IEEE 802.11) Policies by using the Group Policy Management Editor (GPME).

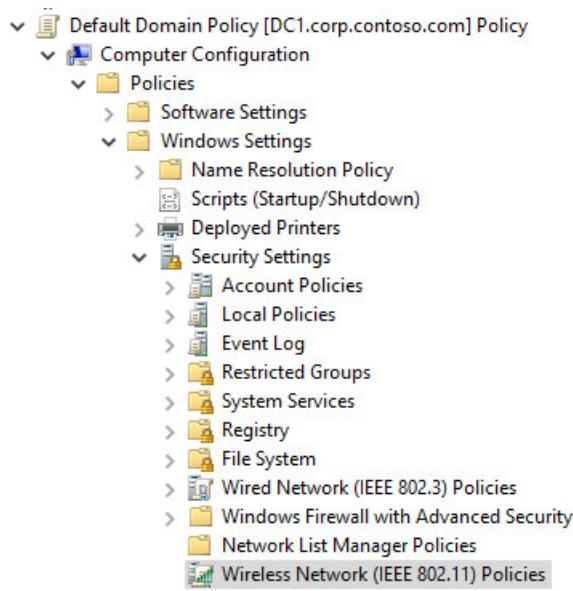
NOTE

After you activate the **Windows Vista and Later Releases** version of the Wireless Network (IEEE 802.11) Policies or the **Windows XP** version, the version option is automatically removed from the list of options when you right-click **Wireless Network (IEEE 802.11) Policies**. This occurs because after you select a policy version, the policy is added in the details pane of the GPME when you select the **Wireless Network (IEEE 802.11) Policies** node. This state remains unless you delete the wireless policy, at which time the wireless policy version returns to the right-click menu for **Wireless Network (IEEE 802.11) Policies** in the GPME. Additionally, the wireless policies are only listed in the GPME details pane when the **Wireless Network (IEEE 802.11) Policies** node is selected.

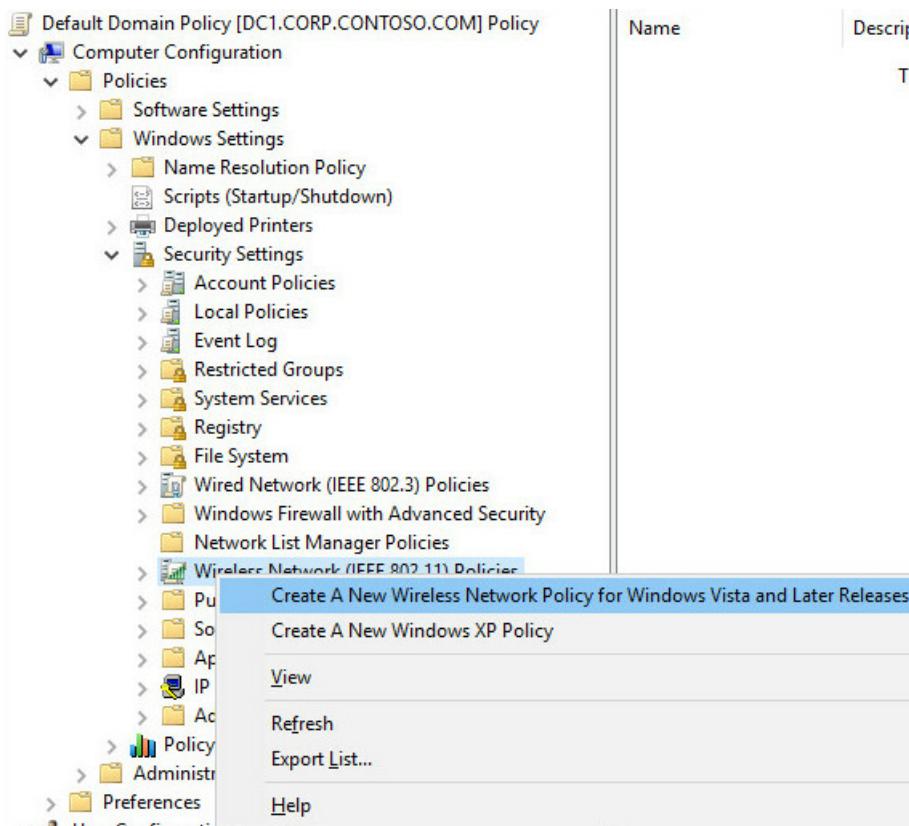
Membership in **Domain Admins**, or equivalent, is the minimum required to perform this procedure.

To activate default Wireless Network (IEEE 802.11) Policies

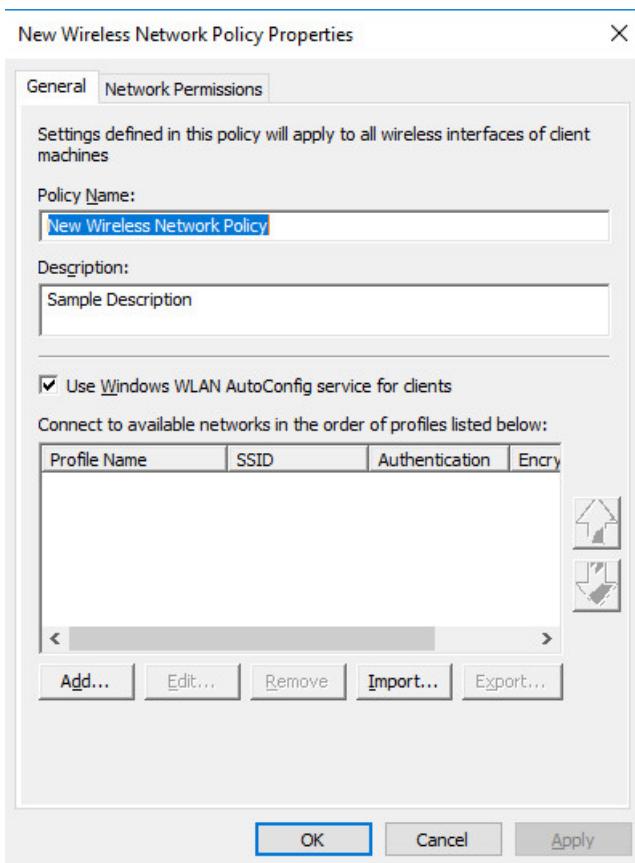
1. Follow the previous procedure, **To open or add and open a Group Policy object** to open the GPME.
2. In the GPME, in the left pane, double-click **Computer Configuration**, double-click **Policies**, double-click **Windows Settings**, and then double-click **Security Settings**.



1. In **Security Settings**, right-click **Wireless Network (IEEE 802.11) Policies**, and then click **Create a new Wireless Policy for Windows Vista and Later Releases**.



1. The **New Wireless Network Policy Properties** dialog box opens. In **Policy Name**, type a new name for the policy or keep the default name. Click **OK** to save the policy. The default policy is activated and listed in the details pane of the GPME with the new name you provided or with the default name **New Wireless Network Policy**.



1. In the details pane, double-click **New Wireless Network Policy** to open it.

In the next section you can perform policy configuration, policy processing preference order, and network permissions.

Configure the New Wireless Network Policy

You can use the procedures in this section to configure Wireless Network (IEEE 802.11) Policy. This policy enables you to configure security and authentication settings, manage wireless profiles, and specify permissions for wireless networks that are not configured as preferred networks.

- [Configure a Wireless Connection Profile for PEAP-MS-CHAP v2](#)
- [Set the Preference Order for Wireless Connection Profiles](#)
- [Define Network Permissions](#)

Configure a Wireless Connection Profile for PEAP-MS-CHAP v2

This procedure provides the steps required to configure a PEAP-MS-CHAP v2 wireless profile.

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

To configure a wireless connection profile for PEAP-MS-CHAP v2

1. In GPME, in the wireless network properties dialog box for the policy that you just created, on the **General** tab and in **Description**, type a brief description for the policy.
2. To specify that WLAN AutoConfig is used to configure wireless network adapter settings, ensure that **Use Windows WLAN AutoConfig service for clients** is selected.
3. In **Connect to available networks in the order of profiles listed below**, click **Add**, and then select **Infrastructure**. The **New Profile properties** dialog box opens.
4. In the **New Profile properties** dialog box, on the **Connection** tab, in the **Profile Name** field, type a new name for the profile. For example, type **Example.com WLAN Profile for Windows 10**.
5. In **Network Name(s) (SSID)**, type the SSID that corresponds to the SSID configured on your wireless APs, and then click **Add**.

If your deployment uses multiple SSIDs and each wireless AP uses the same wireless security settings, repeat this step to add the SSID for each wireless AP to which you want this profile to apply.

If your deployment uses multiple SSIDs and the security settings for each SSID do not match, configure a separate profile for each group of SSIDs that use the same security settings. For example, if you have one group of wireless APs configured to use WPA2-Enterprise and AES, and another group of wireless APs to use WPA-Enterprise and TKIP, configure a profile for each group of wireless APs.

6. If the default text **NEWSSID** is present, select it, and then click **Remove**.
7. If you deployed wireless access points that are configured to suppress the broadcast beacon, select **Connect even if the network is not broadcasting**.

NOTE

Enabling this option can create a security risk because wireless clients will probe for and attempt connections to any wireless network. By default, this setting is not enabled.

8. Click the **Security** tab, click **Advanced**, and then configure the following:
 - a. To configure advanced 802.1X settings, in **IEEE 802.1X**, select **Enforce advanced 802.1X settings**.
When the advanced 802.1X settings are enforced, the default values for **Max Eapol-Start Msgs**, **Held Period**, **Start Period**, and **Auth Period** are sufficient for typical wireless deployments. Because of this, you do not need to change the defaults unless you have a specific reason for doing so.
 - b. To enable Single Sign On, select **Enable Single Sign On for this network**.

- c. The remaining default values in **Single Sign On** are sufficient for typical wireless deployments.
 - d. In **Fast Roaming**, if your wireless AP is configured for pre-authentication, select **This network uses pre-authentication**.
9. To specify that wireless communications meet FIPS 140-2 standards, select **Perform cryptography in FIPS 140-2 certified mode**.
 10. Click **OK** to return to the **Security** tab. In **Select the security methods for this network**, in **Authentication**, select **WPA2-Enterprise** if it is supported by your wireless AP and wireless client network adapters. Otherwise, select **WPA-Enterprise**.
 11. In **Encryption**, if supported by your wireless AP and wireless client network adapters, select **AES-CCMP**. If you are using access points and wireless network adapters that support 802.11ac, select **AES-GCMP**. Otherwise, select **TKIP**.

NOTE

The settings for both **Authentication** and **Encryption** must match the settings configured on your wireless APs. The default settings for **Authentication Mode**, **Max Authentication Failures**, and **Cache user information for subsequent connections to this network** are sufficient for typical wireless deployments.

12. In **Select a network authentication method**, select **Protected EAP (PEAP)**, and then click **Properties**. The **Protected EAP Properties** dialog box opens.
13. In **Protected EAP Properties**, confirm that **Verify the server's identity by validating the certificate** is selected.
14. In **Trusted Root Certification Authorities**, select the trusted root certification authority (CA) that issued the server certificate to your NPS.

NOTE

This setting limits the root CAs that clients trust to the selected CAs. If no trusted root CAs are selected, then clients will trust all root CAs listed in their Trusted Root Certification Authorities certificate store.

15. In the **Select Authentication Method** list, select **Secured password (EAP-MS-CHAP v2)**.
16. Click **Configure**. In the **EAP MSCHAPv2 Properties** dialog box, verify **Automatically use my Windows logon name and password (and domain if any)** is selected, and click **OK**.
17. To enable PEAP Fast Reconnect, ensure that **Enable Fast Reconnect** is selected.
18. To require server cryptobinding TLV on connection attempts, select **Disconnect if server does not present cryptobinding TLV**.
19. To specify that user identity is masked in phase one of authentication, select **Enable Identity Privacy**, and in the textbox, type an anonymous identity name, or leave the textbox blank.

[!NOTES]

- The NPS policy for 802.1X Wireless must be created by using NPS **Connection Request Policy**. If the NPS policy is created by using NPS **Network Policy**, then identity privacy will not work.
- EAP identity privacy is provided by certain EAP methods where an empty or an anonymous identity (different from the actual identity) is sent in response to the EAP identity request. PEAP sends the identity twice during the authentication. In the first phase, the identity is sent in plain text and this identity is used for routing purposes, not for client authentication. The real identity—used for

authentication—is sent during the second phase of the authentication, within the secure tunnel that is established in the first phase. If **Enable Identity Privacy** checkbox is selected, the username is replaced with the entry specified in the textbox. For example, assume **Enable Identity Privacy** is selected and the identity privacy alias **anonymous** is specified in the textbox. For a user with a real identity alias **jdoe@example.com**, the identity sent in first phase of authentication will be changed to **anonymous@example.com**. The realm portion of the 1st phase identity is not modified as it is used for routing purposes.

20. Click **OK** to close the **Protected EAP Properties** dialog box.
21. Click **OK** to close the **Security** tab.
22. If you want to create additional profiles, click **Add**, and then repeat the previous steps, making different choices to customize each profile for the wireless clients and network to which you want the profile applied. When you are done adding profiles, click **OK** to close the Wireless Network Policy Properties dialog box.

In the next section you can order the policy profiles for optimum security.

Set the Preference Order for Wireless Connection Profiles

You can use this procedure if you have created multiple wireless profiles in your wireless network policy and you want to order the profiles for optimal effectiveness and security.

To ensure that wireless clients connect with the highest level of security that they can support, place your most restrictive policies at the top of the list.

For example, if you have two profiles, one for clients that support WPA2 and one for clients that support WPA, place the WPA2 profile higher on the list. This ensures that the clients that support WPA2 will use that method for the connection rather than the less secure WPA.

This procedure provides the steps to specify the order in which wireless connection profiles are used to connect domain member wireless clients to wireless networks.

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

To set the preference order for wireless connection profiles

1. In GPME, in the wireless network properties dialog box for the policy that you just configured, click the **General** tab.
2. On the **General** tab, in **Connect to available networks in the order of profiles listed below**, select the profile that you want to move in the list, and then click either the "up arrow" button or "down arrow" button to move the profile to the desired location in the list.
3. Repeat step 2 for each profile that you want to move in the list.
4. Click **OK** to save all changes.

In the following section, you can define network permissions for the wireless policy.

Define Network Permissions

You can configure settings on the **Network Permissions** tab for the domain members to which Wireless Network (IEEE 802.11) Policies apply.

You can only apply the following settings for wireless networks that are not configured on the **General** tab in the **Wireless Network Policy Properties** page:

- Allow or deny connections to specific wireless networks that you specify by network type and Service Set Identifier (SSID)
- Allow or deny connections to ad hoc networks
- Allow or deny connections to infrastructure networks

- Allow or deny users to view network types (ad hoc or infrastructure) to which they are denied access
- Allow or deny users to create a profile that applies to all users
- Users can only connect to allowed networks by using Group Policy profiles

Membership in **Domain Admins**, or equivalent, is the minimum required to complete these procedures.

To allow or deny connections to specific wireless networks

1. In GPME, in the wireless network properties dialog box, click the **Network Permissions** tab.
2. On the **Network Permissions** tab, click **Add**. The **New Permissions Entry** dialog box opens.
3. In the **New Permission Entry** dialog box, in the **Network Name (SSID)** field, type the network SSID of the network for which you want to define permissions.
4. In **Network Type**, select **Infrastructure** or **Ad hoc**.

NOTE

If you are uncertain whether the broadcasting network is an infrastructure or ad hoc network, you can configure two network permission entries, one for each network type.

5. In **Permission**, select **Allow** or **Deny**.
6. Click **OK**, to return to the **Network Permissions** tab.

To specify additional network permissions (Optional)

1. On the **Network Permissions** tab, configure any or all of the following:
 - To deny your domain members access to ad hoc networks, select **Prevent connections to ad-hoc networks**.
 - To deny your domain members access to infrastructure networks, select **Prevent connections to infrastructure networks**.
 - To allow your domain members to view network types (ad hoc or infrastructure) to which they are denied access, select **Allow user to view denied networks**.
 - To allow users to create profiles that apply to all users, select **Allow everyone to create all user profiles**.
 - To specify that your users can only connect to allowed networks by using Group Policy profiles, select **Only use Group Policy profiles for allowed networks**.

Configure your NPSs

Follow these steps to configure NPSs to perform 802.1X authentication for wireless access:

- [Register NPS in Active Directory Domain Services](#)
- [Configure a Wireless AP as an NPS RADIUS Client](#)
- [Create NPS Policies for 802.1X Wireless using a Wizard](#)

Register NPS in Active Directory Domain Services

You can use this procedure to register a server running Network Policy Server (NPS) in Active Directory Domain Services (AD DS) in the domain where the NPS is a member. For NPSs to be granted permission to read the dial-in properties of user accounts during the authorization process, each NPS must be registered in AD DS.

Registering an NPS adds the server to the **RAS and IAS Servers** security group in AD DS.

NOTE

You can install NPS on a domain controller or on a dedicated server. Run the following Windows PowerShell command to install NPS if you have not yet done so:

```
Install-WindowsFeature NPAS -IncludeManagementTools
```

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

To register an NPS in its default domain

1. On your NPS, in **Server Manager**, click **Tools**, and then click **Network Policy Server**. The NPS snap-in opens.
2. Right-click **NPS (Local)**, and then click **Register Server in Active Directory**. The **Network Policy Server** dialog box opens.
3. In **Network Policy Server**, click **OK**, and then click **OK** again.

Configure a Wireless AP as an NPS RADIUS Client

You can use this procedure to configure an AP, also known as a *network access server (NAS)*, as a Remote Authentication Dial-In User Service (RADIUS) client by using the NPS snap-in.

IMPORTANT

Client computers, such as wireless portable computers and other computers running client operating systems, are not RADIUS clients. RADIUS clients are network access servers—such as wireless access points, 802.1X-capable switches, virtual private network (VPN) servers, and dial-up servers—because they use the RADIUS protocol to communicate with RADIUS servers such as NPSs.

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

To add a network access server as a RADIUS client in NPS

1. On your NPS, in **Server Manager**, click **Tools**, and then click **Network Policy Server**. The NPS snap-in opens.
2. In the NPS snap-in, double-click **RADIUS Clients and Servers**. Right-click **RADIUS Clients**, and then click **New**.
3. In **New RADIUS Client**, verify that the **Enable this RADIUS client** check box is selected.
4. In **New RADIUS Client**, in **Friendly name**, type a display name for the wireless access point.

For example, if you want to add a wireless access point (AP) named AP-01, type **AP-01**.

5. In **Address (IP or DNS)**, type the IP address or fully qualified domain name (FQDN) for the NAS.

If you enter the FQDN, to verify that the name is correct and maps to a valid IP address, click **Verify**, and then in **Verify Address**, in the **Address** field, click **Resolve**. If the FQDN name maps to a valid IP address, the IP address of that NAS will automatically appear in **IP address**. If the FQDN does not resolve to an IP address you will receive a message indicating that no such host is known. If this occurs, verify that you have the correct AP name and that the AP is powered on and connected to the network.

Click **OK** to close **Verify Address**.

6. In **New RADIUS Client**, in **Shared Secret**, do one of the following:

- To manually configure a RADIUS shared secret, select **Manual**, and then in **Shared secret**, type the

strong password that is also entered on the NAS. Retype the shared secret in **Confirm shared secret**.

- To automatically generate a shared secret, select the **Generate** check box, and then click the **Generate** button. Save the generated shared secret, and then use that value to configure the NAS so that it can communicate with the NPS.

IMPORTANT

The RADIUS shared secret that you enter for your virtual AP's in NPS must exactly match the RADIUS shared secret that is configured on your actual wireless AP's. If you use the NPS option to generate a RADIUS shared secret, then you must configure the matching actual wireless AP with the RADIUS shared secret that was generated by NPS.

7. In **New RADIUS Client**, on the **Advanced** tab, in **Vendor name**, specify the NAS manufacturer name. If you are not sure of the NAS manufacturer name, select **RADIUS standard**.
8. In **Additional Options**, if you are using any authentication methods other than EAP and PEAP, and if your NAS supports the use of the message authenticator attribute, select **Access Request messages must contain the Message-Authenticator attribute**.
9. Click **OK**. Your NAS appears in the list of RADIUS clients configured on the NPS.

Create NPS Policies for 802.1X Wireless Using a Wizard

You can use this procedure to create the connection request policies and network policies required to deploy either 802.1X-capable wireless access points as Remote Authentication Dial-In User Service (RADIUS) clients to the RADIUS server running Network Policy Server (NPS).

After you run the wizard, the following policies are created:

- One connection request policy
- One network policy

NOTE

You can run the New IEEE 802.1X Secure Wired and Wireless Connections wizard every time you need to create new policies for 802.1X authenticated access.

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

Create policies for 802.1X authenticated wireless by using a wizard

1. Open the NPS snap-in. If it is not already selected, click **NPS (Local)**. If you are running the NPS MMC snap-in and want to create policies on a remote NPS, select the server.
2. In **Getting Started**, in **Standard Configuration**, select **RADIUS server for 802.1X Wireless or Wired Connections**. The text and links below the text change to reflect your selection.
3. Click **Configure 802.1X**. The Configure 802.1X wizard opens.
4. On the **Select 802.1X Connections Type** wizard page, in **Type of 802.1X connections**, select **Secure Wireless Connections**, and in **Name**, type a name for your policy, or leave the default name **Secure Wireless Connections**. Click **Next**.
5. On the **Specify 802.1X Switches** wizard page, in **RADIUS clients**, all 802.1X switches and wireless access points that you have added as RADIUS Clients in the NPS snap-in are shown. Do any of the following:
 - To add additional network access servers (NASs), such as wireless APs, in **RADIUS clients**, click

Add, and then in **New RADIUS client**, enter the information for: **Friendly name, Address (IP or DNS), and Shared Secret**.

- To modify the settings for any NAS, in **RADIUS clients**, select the AP for which you want to modify the settings, and then click **Edit**. Modify the settings as required.
- To remove a NAS from the list, in **RADIUS clients**, select the NAS, and then click **Remove**.

WARNING

Removing a RADIUS client from within the **Configure 802.1X** wizard deletes the client from the NPS configuration. All additions, modifications, and deletions that you make within the **Configure 802.1X** wizard to RADIUS clients are reflected in the NPS snap-in, in the **RADIUS Clients** node under **NPS / RADIUS Clients and Servers**. For example, if you use the wizard to remove an 802.1X switch, the switch is also removed from the NPS snap-in.

6. Click **Next**. On the **Configure an Authentication Method** wizard page, in **Type (based on method of access and network configuration)**, select **Microsoft: Protected EAP (PEAP)**, and then click **Configure**.

TIP

If you receive an error message indicating that a certificate cannot be found for use with the authentication method, and you have configured Active Directory Certificate Services to automatically issue certificates to RAS and IAS servers on your network, first ensure that you have followed the steps to Register NPS in Active Directory Domain Services, then use the following steps to update Group Policy: Click **Start**, click **Windows System**, click **Run**, and in **Open**, type **gpupdate**, and then press ENTER. When the command returns results indicating that both user and computer Group Policy have updated successfully, select **Microsoft: Protected EAP (PEAP)** again, and then click **Configure**.

If after refreshing Group Policy you continue to receive the error message indicating that a certificate cannot be found for use with the authentication method, the certificate is not being displayed because it does not meet the minimum server certificate requirements as documented in the Core Network Companion Guide: [Deploy Server Certificates for 802.1X Wired and Wireless Deployments](#). If this happens, you must discontinue NPS configuration, revoke the certificate issued to your NPS(s), and then follow the instructions to configure a new certificate by using the server certificates deployment guide.

7. On the **Edit Protected EAP Properties** wizard page, in **Certificate issued**, ensure that the correct NPS certificate is selected, and then do the following:

NOTE

Verify that the value in **Issuer** is correct for the certificate selected in **Certificate issued**. For example, the expected issuer for a certificate issued by a CA running Active Directory Certificate Services (AD CS) named corp\DC1, in the domain contoso.com, is **corp-DC1-CA**.

- To allow users to roam with their wireless computers between access points without requiring them to reauthenticate each time they associate with a new AP, select **Enable Fast Reconnect**.
- To specify that connecting wireless clients will end the network authentication process if the RADIUS server does not present cryptobinding Type-Length-Value (TLV), select **Disconnect Clients without Cryptobinding**.
- To modify the policy settings for the EAP type, in **EAP Types**, click **Edit**, in **EAP MSCHAPv2 Properties**, modify the settings as needed, and then click **OK**.

8. Click **OK**. The Edit Protected EAP Properties dialog box closes, returning you to the **Configure 802.1X** wizard. Click **Next**.
9. In **Specify User Groups**, click **Add**, and then type the name of the security group that you configured for your wireless clients in the Active Directory Users and Computers snap-in. For example, if you named your wireless security group Wireless Group, type **Wireless Group**. Click **Next**.
10. Click **Configure** to configure RADIUS standard attributes and vendor-specific attributes for virtual LAN (VLAN) as needed, and as specified by the documentation provided by your wireless AP hardware vendor. Click **Next**.
11. Review the configuration summary details, and then click **Finish**.

Your NPS policies are now created, and you can move on to joining wireless computers to the domain.

Join New Wireless Computers to the Domain

The easiest method to join new wireless computers to the domain is to physically attach the computer to a segment of the wired LAN (a segment not controlled by an 802.1X switch) before joining the computer to the domain. This is easiest because wireless group policy settings are automatically and immediately applied and, if you have deployed your own PKI, the computer receives the CA certificate and places it in the Trusted Root Certification Authorities certificate store, allowing the wireless client to trust NPSs with server certs issued by your CA.

Likewise, after a new wireless computer is joined to the domain, the preferred method for users to log on to the domain is to perform log on by using a wired connection to the network.

Other domain-join methods

In cases where it is not practical to join computers to the domain by using a wired Ethernet connection, or in cases where the user cannot log on to the domain for the first time by using a wired connection, you must use an alternate method.

- **IT Staff Computer Configuration.** A member of the IT staff joins a wireless computer to the domain and configures a Single Sign On bootstrap wireless profile. With this method, the IT administrator connects the wireless computer to the wired Ethernet network and joins the computer to the domain. Then the administrator distributes the computer to the user. When the user starts the computer without using a wired connection, the domain credentials that they manually specify for the user logon are used to both establish a connection to the wireless network and to log on to the domain.

For more information, see the section [Join the Domain and Log On by using the IT Staff Computer Configuration Method](#)

- **Bootstrap Wireless Profile Configuration by Users.** The user manually configures the wireless computer with a bootstrap wireless profile and joins the domain, based on instructions acquired from an IT administrator. The bootstrap wireless profile allows the user to establish a wireless connection and then join the domain. After joining the computer to the domain and restarting the computer, the user can log on to the domain by using a wireless connection and their domain account credentials.

For more information, see the section [Join the Domain and Log On by using Bootstrap Wireless Profile Configuration by Users](#).

Join the Domain and Log On by using the IT Staff Computer Configuration Method

Domain member users with domain-joined wireless client computers can use a temporary wireless profile to connect to an 802.1X-authenticated wireless network without first connecting to the wired LAN. This temporary wireless profile is called a *bootstrap wireless profile*.

A bootstrap wireless profile requires the user to manually specify their domain user account credentials, and does

not validate the certificate of the Remote Authentication Dial-In User Service (RADIUS) server running Network Policy Server (NPS).

After wireless connectivity is established, Group Policy is applied on the wireless client computer, and a new wireless profile is issued automatically. The new policy uses the computer and user account credentials for client authentication.

Additionally, as part of the PEAP-MS-CHAP v2 mutual authentication using the new profile instead of the bootstrap profile, the client validates the credentials of the RADIUS server.

After you join the computer to the domain, use this procedure to configure a Single Sign On bootstrap wireless profile, before distributing the wireless computer to the domain-member user.

To configure a Single Sign On bootstrap wireless profile

1. Create a bootstrap profile by using the procedure in this guide named [Configure a Wireless Connection Profile for PEAP-MS-CHAP v2](#), and use the following settings:
 - PEAP-MS-CHAP v2 authentication
 - Validate RADIUS server certificate disabled
 - Single Sign On enabled
2. In the properties of the Wireless Network Policy within which you created the new bootstrap profile, on the **General** tab, select the bootstrap profile, and then click **Export** to export the profile to a network share, USB flash drive, or other easily accessible location. The profile is saved as an *.xml file to the location that you specify.
3. Join the new wireless computer to the domain (for example, through an Ethernet connection that does not require IEEE 802.1X authentication) and add the bootstrap wireless profile to the computer by using the **netsh wlan add profile** command.

NOTE

For more information, see Netsh Commands for Wireless Local Area Network (WLAN) at <http://technet.microsoft.com/library/dd744890.aspx>.

4. Distribute the new wireless computer to the user with the procedure to "Log on to the domain using computers running Windows 10."

When the user starts the computer, Windows prompts the user to enter their domain user account name and password. Because Single Sign On is enabled, the computer uses the domain user account credentials to first establish a connection with the wireless network and then log on to the domain.

Log on to the domain using computers running Windows 10

1. Log off the computer, or restart the computer.
2. Press any key on your keyboard or click on the desktop. The logon screen appears with a local user account name displayed and a password entry field below the name. Do not log on with the local user account.
3. In the lower left corner of the screen, click **Other User**. The Other User log on screen appears with two fields, one for user name and one for password. Below the password field is the text **Sign on to:** and then the name of the domain where the computer is joined. For example, if your domain is named example.com, the text reads **Sign on to: EXAMPLE**.
4. In **User name**, type your domain user name.
5. In **Password**, type your domain password, and then click the arrow, or press ENTER.

NOTE

If the **Other User** screen does not include the text **Sign on to:** and your domain name, you should enter your user name in the format *domain\user*. For example, to log on to the domain example.com with an account named **User-01**, type **example\User-01**.

Join the Domain and Log On by using Bootstrap Wireless Profile Configuration by Users

With this method, you complete the steps in the General steps section, then you provide your domain-member users with the instructions about how to manually configure a wireless computer with a bootstrap wireless profile. The bootstrap wireless profile allows the user to establish a wireless connection and then join the domain. After the computer is joined to the domain and restarted, the user can log on to the domain through a wireless connection.

General steps

1. Configure a local computer administrator account, in **Control Panel**, for the user.

IMPORTANT

To join a computer to a domain, the user must be logged on to the computer with the local Administrator account. Alternatively, the user must provide the credentials for the local Administrator account during the process of joining the computer to the domain. In addition, the user must have a user account in the domain to which the user wants to join the computer. During the process of joining the computer to the domain, the user will be prompted for domain account credentials (user name and password).

2. Provide your domain users with the instructions for configuring a bootstrap wireless profile, as documented in the following procedure **To configure a bootstrap wireless profile**.
3. Additionally, provide users with both the local computer credentials (user name and password), and domain credentials (domain user account name and password) in the form *DomainName\UserName*, as well as the procedures to "Join the computer to the domain," and to "Log on to the domain," as documented in the Windows Server 2016 [Core Network Guide](#).

To configure a bootstrap wireless profile

1. Use the credentials provided by your network administrator or IT support professional to log on to the computer with the local computer's Administrator account.
2. Right-click the network icon on the desktop, and click **Open Network and Sharing Center**. **Network and Sharing Center** opens. In **Change your networking settings**, click **Set up a new connection or network**. The **Set Up a Connection or Network** dialog box opens.
3. Click **Manually connect to a wireless network**, and then click **Next**.
4. In **Manually connect to a wireless network**, in **Network name**, type the SSID name of the AP.
5. In **Security type**, select the setting provided by your administrator.
6. In **Encryption type** and **Security Key**, select or type the settings provided by your administrator.
7. Select **Start this connection automatically**, and then click **Next**.
8. In **Successfully added Your Network SSID**, click **Change connection settings**.
9. Click **Change connection settings**. The *Your Network SSID* Wireless Network property dialog box opens.
10. Click the **Security** tab, and then in **Choose a network authentication method**, select **Protected EAP (PEAP)**.
11. Click **Settings**. The **Protected EAP (PEAP) Properties** page opens.

12. In the **Protected EAP (PEAP) Properties** page, ensure that **Validate server certificate** is not selected, click **OK** twice, and then click **Close**.
13. Windows then attempts to connect to the wireless network. The settings of the bootstrap wireless profile specify that you must provide your domain credentials. When Windows prompts you for an account name and password, type your domain account credentials as follows: *Domain Name\User Name, Domain Password*.

To join a computer to the domain

1. Log on to the computer with the local Administrator account.
2. In the search text box, type **PowerShell**. In search results, right-click **Windows PowerShell**, and then click **Run as administrator**. Windows PowerShell opens with an elevated prompt.
3. In Windows PowerShell, type the following command, and then press ENTER. Ensure that you replace the variable `DomainName` with the name of the domain that you want to join.

`Add-Computer DomainName`

4. When prompted, type your domain user name and password, and click **OK**.
5. Restart the computer.
6. Follow the instructions in the previous section [Log on to the domain using computers running Windows 10](#).

Deploy BranchCache Hosted Cache Mode

9/1/2018 • 6 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

The Windows Server 2016 Core Network Guide provides instructions for planning and deploying the core components required for a fully functioning network and a new Active Directory® domain in a new forest.

This guide explains how to build on the core network by providing instructions for deploying BranchCache in hosted cache mode in one or more branch offices with a Read-Only Domain Controller where client computers are running Windows® 10, Windows 8.1, or Windows 8, and are joined to the domain.

IMPORTANT

Do not use this guide if you are planning to deploy or have already deployed a BranchCache hosted cache server that is running Windows Server 2008 R2. This guide provides instructions for deploying hosted cache mode with a hosted cache server that is running Windows Server® 2016, Windows Server 2012 R2, or Windows Server 2012.

This guide contains the following sections.

- [Prerequisites for using this guide](#)
- [About this guide](#)
- [What this guide does not provide](#)
- [Technology overviews](#)
- [BranchCache Hosted Cache Mode Deployment Overview](#)
- [BranchCache Hosted Cache Mode Deployment Planning](#)
- [BranchCache Hosted Cache Mode Deployment](#)
- [Additional Resources](#)

Prerequisites for using this guide

This is a companion guide to the Windows Server 2016 Core Network Guide. To deploy BranchCache in hosted cache mode with this guide, you must first do the following.

- Deploy a core network in your main office by using the Core Network Guide, or already have the technologies provided in the Core Network Guide installed and functioning correctly on your network. These technologies include TCP/IP v4, DHCP, Active Directory Domain Services (AD DS), and DNS.

NOTE

The Windows Server 2016 [Core Network Guide](#) is available in the Windows Server 2016 Technical Library.

- Deploy BranchCache content servers that are running Windows Server 2016, Windows Server 2012 R2, or Windows Server 2012 in your main office or in a cloud data center. For information on how to deploy BranchCache content servers, see [Additional Resources](#).

- Establish wide area network (WAN) connections between your branch office, your main office and, if appropriate, your Cloud resources, by using a virtual private network (VPN), DirectAccess, or other connection method.
- Deploy client computers in your branch office that are running one of the following operating systems, which provide BranchCache with support for Background Intelligent Transfer Service (BITS), Hyper Text Transfer Protocol (HTTP), and Server Message Block (SMB).
 - Windows 10 Enterprise
 - Windows 10 Education
 - Windows 8.1 Enterprise
 - Windows 8 Enterprise

NOTE

In the following operating systems, BranchCache does not support HTTP and SMB functionality, but does support BranchCache BITS functionality.

- Windows 10 Pro, BITS support only
- Windows 8.1 Pro, BITS support only
- Windows 8 Pro, BITS support only

About this guide

This guide is designed for network and system administrators who have followed the instructions in the Windows Server 2016 Core Network Guide or Windows Server 2012 Core Network Guide to deploy a core network, or for those who have previously deployed the technologies included in the Core Network Guide, including Active Directory Domain Services (AD DS), Domain Name Service (DNS), Dynamic Host Configuration Protocol (DHCP), and TCP/IP v4.

It is recommended that you review the design and deployment guides for each of the technologies that are used in this deployment scenario. These guides can help you determine whether this deployment scenario provides the services and configuration that you need for your organization's network.

What this guide does not provide

This guide does not provide conceptual information about BranchCache, including information about BranchCache modes and capabilities.

This guide does not provide information about how to deploy WAN connections or other technologies in your branch office, such as DHCP, a RODC, or a VPN server.

In addition, this guide does not provide guidance on the hardware you should use when you deploy a hosted cache server. It is possible to run other services and applications on your hosted cache server, however you must make the determination, based on workload, hardware capabilities, and branch office size, whether to install BranchCache hosted cache server on a particular computer, and how much disk space to allocate for the cache.

This guide does not provide instructions for configuring computers that are running Windows 7. If you have client computers that are running Windows 7 in your branch offices, you must configure them using procedures that are different than those provided in this guide for client computers that are running Windows 10, Windows 8.1, and Windows 8.

In addition, if you have computers running Windows 7, you must configure your hosted cache server with a server certificate that is issued by a certification authority that client computers trust. (If all of your client computers are running Windows 10, Windows 8.1, or Windows 8, you do not need to configure the hosted cache server with a

server certificate.)

IMPORTANT

If your hosted cache servers are running Windows Server 2008 R2, use the Windows Server 2008 R2 [BranchCache Deployment Guide](#) instead of this guide to deploy BranchCache in hosted cache mode. Apply the Group Policy settings that are described in that guide to all BranchCache clients that are running versions of Windows from Windows 7 to Windows 10. Computers that are running Windows Server 2008 R2 cannot be configured by using the steps in this guide.

Technology overviews

For this companion guide, BranchCache is the only technology that you need to install and configure. You must run Windows PowerShell BranchCache commands on your content servers, such as Web and file servers, however you do not need to change or reconfigure the content servers in any other way. In addition, you must configure client computers by using Group Policy on your domain controllers that are running AD DS on Windows Server 2016, Windows Server 2012 R2, or Windows Server 2012.

BranchCache

BranchCache is a wide area network (WAN) bandwidth optimization technology that is included in some editions of the Windows Server 2016 and Windows 10 operating systems, as well as in some editions of Windows Server 2012 R2, Windows 8.1, Windows Server 2012, Windows 8, Windows Server 2008 R2, and Windows 7.

To optimize WAN bandwidth when users access content on remote servers, BranchCache downloads client-requested content from your main office or hosted cloud content servers and caches the content at branch office locations, allowing other client computers at branch offices to access the same content locally rather than over the WAN.

When you deploy BranchCache in hosted cache mode, you must configure client computers in the branch office as hosted cache mode clients, and then you must deploy a hosted cache server in the branch office. This guide demonstrates how to deploy your hosted cache server with prehashed and preloaded content from your Web and file server-based content servers.

Group Policy

Group Policy in Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012 is an infrastructure used to deliver and apply one or more desired configurations or policy settings to a set of targeted users and computers within an Active Directory environment.

This infrastructure consists of a Group Policy engine and multiple client-side extensions (CSEs) that are responsible for reading policy settings on target client computers.

Group Policy is used in this scenario to configure domain member client computers with BranchCache hosted cache mode.

To continue with this guide, see [BranchCache Hosted Cache Mode Deployment Overview](#).

BranchCache Hosted Cache Mode Deployment Overview

9/1/2018 • 4 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

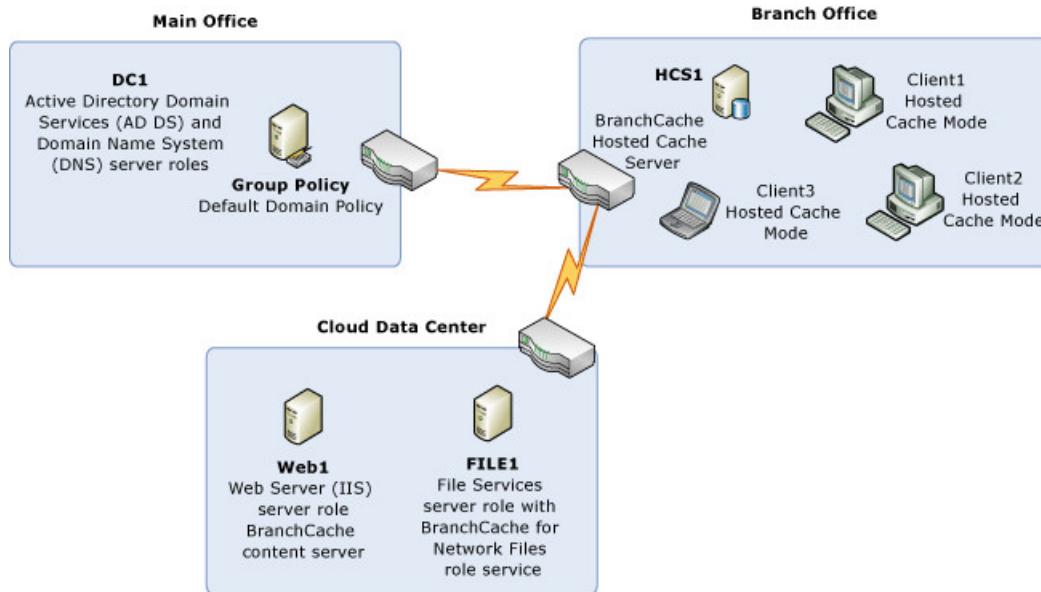
You can use this guide to deploy a BranchCache hosted cache server in a branch office where computers are joined to a domain. You can use this topic to gain an overview of the BranchCache Hosted Cache Mode deployment process.

This overview includes the BranchCache infrastructure that you need, as well as a simple step-by-step overview of deployment.

Hosted Cache Server deployment infrastructure

In this deployment, the hosted cache server is deployed by using service connection points in Active Directory Domain Services (AD DS), and you have the option with BranchCache in Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012, to prehash the shared content on Web and file based content servers, then preload the content on hosted cache servers.

The following illustration shows the infrastructure that is required to deploy a BranchCache hosted cache server.



IMPORTANT

Although this deployment depicts content servers in a cloud data center, you can use this guide to deploy a BranchCache hosted cache server regardless of where you deploy your content servers – in your main office or in a cloud location.

HCS1 in the branch office

You must configure this computer as a hosted cache server. If you choose to prehash content server data so that you can preload the content on your hosted cache servers, you can import data packages that contain the content from your Web and file servers.

WEB1 in the cloud data center

WEB1 is a BranchCache-enabled content server. If you choose to prehash content server data so that you can preload the content on your hosted cache servers, you can prehash the shared content on WEB1, then create a data package that you copy to HCS1.

FILE1 in the cloud data center

FILE1 is a BranchCache-enabled content server. If you choose to prehash content server data so that you can preload the content on your hosted cache servers, you can prehash the shared content on FILE1, then create a data package that you copy to HCS1.

DC1 in the main office

DC1 is a domain controller, and you must configure the Default Domain Policy, or another policy that is more appropriate for your deployment, with BranchCache Group Policy settings to enable Automatic Hosted Cache Discovery by Service Connection Point.

When client computers in the branch have Group Policy refreshed and this policy setting is applied, they automatically locate and begin to use the hosted cache server in the branch office.

Client computers in the branch office

You must refresh Group Policy on client computers to apply new BranchCache Group Policy settings and to allow clients to locate and use the hosted cache server.

Hosted Cache Server deployment process overview

NOTE

The details of how to perform these steps are provided in the section [BranchCache Hosted Cache Mode Deployment](#).

The process of deploying a BranchCache Hosted Cache Server occurs in these stages:

NOTE

Some of the steps below are optional, such as those steps that demonstrate how to prehash and preload content on hosted cache servers. When you deploy BranchCache in hosted cache mode, you are not required to prehash content on your Web and file content servers, to create a data package, and to import the data package in order to preload your hosted cache servers with content. The steps are noted as optional in this section and in the section [BranchCache Hosted Cache Mode Deployment](#) so that you can skip them if you prefer.

1. On HCS1, use Windows PowerShell commands to configure the computer as a hosted cache server and to register a Service Connection Point in Active Directory.
2. (Optional) On HCS1, if the BranchCache default values do not match your deployment goals for the server and the hosted cache, configure the amount of disk space that you want to allocate for the hosted cache. Also configure the disk location that you prefer for the hosted cache.
3. (Optional) Prehash content on content servers, create data packages, and preload content on the hosted cache server.

NOTE

Prehashing and preloading content on your hosted cache server is optional, however if you choose to prehash and preload, you must perform all of the steps below that are applicable to your deployment. (For example, if you do not have Web servers, you do not need to perform any of the steps related to prehashing and preloading Web server content.)

- a. On WEB1, prehash Web server content and create a data package.
 - b. On FILE1, prehash file server content and create a data package.
 - c. From WEB1 and FILE1, copy the data packages to the hosted cache server HCS1.
 - d. On HCS1, import the data packages to preload the data cache.
4. On DC1, configure domain joined branch office client computers for hosted cache mode by configuring Group Policy with BranchCache policy settings.
 5. On client computers, refresh Group Policy.

To continue with this guide, see [BranchCache Hosted Cache Mode Deployment Planning](#).

BranchCache Hosted Cache Mode Deployment Planning

9/1/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

You can use this topic to plan your deployment of BranchCache in Hosted Cache mode.

IMPORTANT

Your hosted cache server must be running Windows Server 2016, Windows Server 2012 R2, or Windows Server 2012.

Before you deploy your hosted cache server, you must plan the following items:

- [Plan basic server configuration](#)
- [Plan domain access](#)
- [Plan the location and size of the hosted cache](#)
- [Plan the share to which the content server packages are to be copied](#)
- [Plan prehashing and data package creation on content servers](#)

Plan basic server configuration

If you are planning on using an existing server in your branch office as your hosted cache server, you do not need to perform this planning step, because the computer is already named and has an IP address configuration.

After you install Windows Server 2016 on your hosted cache server, you must rename the computer and assign and configure a static IP address for the local computer.

NOTE

In this guide, the hosted cache server is named HCS1, however you should use a server name that is appropriate for your deployment.

Plan domain access

If you are planning on using an existing server in your branch office as your hosted cache server, you do not need to perform this planning step, unless the computer is not currently joined to the domain.

To log on to the domain, the computer must be a domain member computer and the user account must be created in AD DS before the logon attempt. In addition, you must join the computer to the domain with an account that has the appropriate group membership.

Plan the location and size of the hosted cache

On HCS1, determine where on your hosted cache server you want to locate the hosted cache. For example, decide

the hard disk, volume, and folder location where you plan to store the cache.

In addition, decide what percentage of disk space you want to allocate for the hosted cache.

Plan the share to which the content server packages are to be copied

After you create data packages on your content servers, you must copy them over the network to a share on your hosted cache server.

Plan the folder location and sharing permissions for the shared folder. In addition, if your content servers host a large amount of data and the packages that you create will be large files, plan to perform the copy operation during off-peak hours so that WAN bandwidth is not consumed by the copy operation during a time when others need to use the bandwidth for normal business operations.

Plan prehashing and data package creation on content servers

Before you prehash content on your content servers, you must identify the folders and files that contain content that you want to add to the data package.

In addition, you must plan on the local folder location where you can store the data packages before copying them to the hosted cache server.

To continue with this guide, see [BranchCache Hosted Cache Mode Deployment](#).

BranchCache Hosted Cache Mode Deployment

9/1/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

You can use this topic for links to detailed procedural topics that guide you through the BranchCache hosted cache mode deployment process.

Follow these steps to deploy BranchCache hosted cache mode.

- [Install the BranchCache Feature and Configure the Hosted Cache Server by Service Connection Point](#)
- [Move and Resize the Hosted Cache \(Optional\)](#)
- [Prehash and Preload Content on the Hosted Cache Server \(Optional\)](#)
- [Configure Client Automatic Hosted Cache Discovery by Service Connection Point](#)

NOTE

The procedures in this guide do not include instructions for cases in which the **User Account Control** dialog box opens to request your permission to continue. If this dialog box opens while you are performing the procedures in this guide, and if the dialog box was opened in response to your actions, click **Continue**.

To continue with this guide, see [Install the BranchCache Feature and Configure the Hosted Cache Server by Service Connection Point](#).

Install the BranchCache Feature and Configure the Hosted Cache Server by Service Connection Point

9/1/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

You can use this procedure to install the BranchCache feature on your hosted cache server, HCS1, and to configure the server to register a Service Connection Point (SCP) in Active Directory Domain Services (AD DS).

When you register hosted cache servers with an SCP in AD DS, the SCP allows client computers that are configured correctly to automatically discover hosted cache servers by querying AD DS for the SCP. Instructions on how to configure client computers to perform this action are provided later in this guide.

IMPORTANT

Before you perform this procedure, you must join the computer to the domain and configure the computer with a static IP address.

To perform this procedure, you must be a member of the Administrators group.

To install the BranchCache feature and configure the hosted cache server

1. On the server computer, run Windows PowerShell as an Administrator. Type the following command, and then press ENTER.

```
Install-WindowsFeature BranchCache
```

2. To configure the computer as a hosted cache server after the BranchCache feature is installed, and to register a Service Connection Point in AD DS, type the following command in Windows PowerShell, and then press ENTER.

```
Enable-BCHostedServer -RegisterSCP
```

3. To verify the hosted cache server configuration, type the following command and press ENTER.

```
Get-BCStatus
```

The results of the command display status for all aspects of your BranchCache installation. Following are a few of the BranchCache settings and the correct value for each item:

- BranchCachelsEnabled: True
- HostedCacheServerIsEnabled: True
- HostedCacheScpRegistrationEnabled: True

4. To prepare for the step of copying your data packages from your content servers to your hosted cache servers, either identify an existing share on the hosted cache server or create a new folder and share the folder so that it is accessible from your content servers. After you create your data packages on your content servers, you will copy the data packages to this shared folder on the hosted cache server.
5. If you are deploying more than one hosted cache server, repeat this procedure on each server.

To continue with this guide, see [Move and Resize the Hosted Cache \(Optional\)](#).

Move and Resize the Hosted Cache (Optional)

9/1/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

You can use this procedure to move the hosted cache to the drive and folder that you prefer, and to specify the amount of disk space that the hosted cache server can use for the hosted cache.

This procedure is optional. If the default cache location (%windir%\ServiceProfiles\NetworkService\AppData\Local\PeerDistPub) and size – which is 5% of the total hard disk space – are appropriate for your deployment, you do not need to change them.

You must be a member of the Administrators group to perform this procedure.

To move and resize the hosted cache

1. Open Windows PowerShell with Administrator privileges.
2. Type the following command to move the hosted cache to another location on the local computer, and then press ENTER.

IMPORTANT

Before running the following command, replace parameter values, such as –Path and –MoveTo, with values that are appropriate for your deployment.

```
Set-BCCache -Path C:\datacache -MoveTo D:\datacache
```

3. Type the following command to resize the hosted cache –specifically the datacache - on the local computer. Press ENTER.

IMPORTANT

Before running the following command, replace parameter values, such as –Percentage, with values that are appropriate for your deployment.

```
Set-BCCache -Percentage 20
```

4. To verify the hosted cache server configuration, type the following command and press ENTER.

```
Get-BCStatus
```

The results of the command display status for all aspects of your BranchCache installation. Following are a few of the BranchCache settings and the correct value for each item:

- DataCache | CacheFilePath: Displays the hard disk location that matches the value you provided with the –MoveTo parameter of the SetBCCache command. For example, if you provided the value D:\datacache, that value is displayed in the command output.

- DataCache | MaxCacheSizeAsPercentageOfDiskVolume: Displays the number that matches the value you provided with the –Percentage parameter of the SetBCCache command. For example, if you provided the value 20, that value is displayed in the command output.

To continue with this guide, see [Prehash and Preload Content on the Hosted Cache Server \(Optional\)](#).

Prehash and Preload Content on the Hosted Cache Server (Optional)

9/1/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

You can use the procedures in this section to prehash content on your content servers, add the content to data packages, and then preload the content on your hosted cache servers.

These procedures are optional because you are not required to prehash and preload content on your hosted cache servers.

If you do not preload content, data is added to the hosted cache automatically as clients download it over the WAN connection.

IMPORTANT

Although these procedures are collectively optional, if you decide to prehash and preload content on your hosted cache servers, performing both procedures is required.

- [Create Content Server Data Packages for Web and File Content \(Optional\)](#)
- [Import Data Packages on the Hosted Cache Server \(Optional\)](#)

To continue with this guide, see [Create Content Server Data Packages for Web and File Content \(Optional\)](#).

Create Content Server Data Packages for Web and File Content (Optional)

9/1/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

You can use this procedure to prehash content on Web and file servers, and then create data packages to import on your hosted cache server.

This procedure is optional because you are not required to prehash and preload content on your hosted cache servers. If you do not preload content, data is added to the hosted cache automatically as clients download it over the WAN connection.

This procedure provides instructions for prehashing content on both file servers and Web servers. If you do not have one of those types of content servers, you do not have to perform the instructions for that content server type.

IMPORTANT

Before you perform this procedure, you must install and configure BranchCache on your content servers. In addition, if you plan on changing the server secret on a content server, do so before pre-hashing content – modifying the server secret invalidates previously-generated hashes.

To perform this procedure, you must be a member of the Administrators group.

To create content server data packages

1. On each content server, locate the folders and files that you want to prehash and add to a data package.
Identify or create a folder where you want to save your data package later in this procedure.
2. On the server computer, open Windows PowerShell with Administrator privileges.
3. Do one or both of the following, depending on the types of content servers that you have:

NOTE

The value for the `-Path` parameter is the folder where your content is located. You must replace the example values in the commands below with a valid folder location on your content server that contains data that you want to prehash and add to a package.

- If the content that you want to prehash is on a file server, type the following command, and then press ENTER.

```
Publish-BCFileContent -Path D:\share -StageData
```

- If the content that you want to prehash is on a Web server, type the following command, and then press ENTER.

```
Publish-BCWebContent -Path D:\inetpub\wwwroot -StageData
```

4. Create the data package by running the following command on each of your content servers. Replace the example value (D:\temp) for the –Destination parameter with the location that you identified or created at the beginning of this procedure.

```
Export-BCDataPackage -Destination D:\temp
```

5. From the content server, access the share on your hosted cache servers where you want to preload content, and copy the data packages to the shares on the hosted cache servers.

To continue with this guide, see [Import Data Packages on the Hosted Cache Server \(Optional\)](#).

Import Data Packages on the Hosted Cache Server (Optional)

9/1/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

You can use this procedure to import data packages and preload content on your hosted cache servers.

This procedure is optional because you are not required to prehash and preload content on your hosted cache servers.

If you do not pre-load content, data is added to the hosted cache automatically as clients download it over the WAN connection.

You must be a member of the Administrators group to perform this procedure.

To import data packages on the hosted cache server

1. On the server computer, open Windows PowerShell with Administrator privileges.
2. Type the following command, replacing the value for the –Path parameter with the folder location where you have stored your data packages, and then press ENTER.

```
Import-BCCachePackage –Path D:\temp\PeerDistPackage.zip
```

3. If you have more than one hosted cache server where you want to preload content, perform this procedure on each hosted cache server.

To continue with this guide, see [Configure Client Automatic Hosted Cache Discovery by Service Connection Point](#).

Configure Client Automatic Hosted Cache Discovery by Service Connection Point

9/1/2018 • 3 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

With this procedure you can use Group Policy to enable and configure BranchCache hosted cache mode on domain-joined computers that are running the following BranchCache-capable Windows operating systems.

- Windows 10 Enterprise
- Windows 10 Education
- Windows 8.1 Enterprise
- Windows 8 Enterprise

NOTE

To configure domain-joined computers that are running Windows Server 2008 R2 or Windows 7, see the Windows Server 2008 R2 [BranchCache Deployment Guide](#).

Membership in **Domain Admins**, or equivalent is the minimum required to perform this procedure.

To use Group Policy to configure clients for hosted cache mode

1. On a computer upon which the Active Directory Domain Services server role is installed, open Server Manager, select the Local Server, click **Tools**, and then click **Group Policy Management**. The Group Policy Management console opens.
2. In the Group Policy Management console, expand the following path: **Forest: corp.contoso.com, Domains, corp.contoso.com, Group Policy Objects**, where *corp.contoso.com* is the name of the domain where the BranchCache client computer accounts that you want to configure are located.
3. Right-click **Group Policy Objects**, and then click **New**. The **New GPO** dialog box opens. In **Name**, type a name for the new Group Policy object (GPO). For example, if you want to name the object BranchCache Client Computers, type **BranchCache Client Computers**. Click **OK**.
4. In the Group Policy Management console, ensure that **Group Policy Objects** is selected, and in the details pane right-click the GPO that you just created. For example, if you named your GPO BranchCache Client Computers, right-click **BranchCache Client Computers**. Click **Edit**. The Group Policy Management Editor console opens.
5. In the Group Policy Management Editor console, expand the following path: **Computer Configuration, Policies, Administrative Templates: Policy definitions (ADMX files) retrieved from the local computer, Network, BranchCache**.
6. Click **BranchCache**, and then in the details pane, double-click **Turn on BranchCache**. The **Turn on BranchCache** dialog box opens.
7. In the **Turn on BranchCache** dialog box, click **Enabled**, and then click **OK**.
8. In the Group Policy Management Editor console, ensure that **BranchCache** is still selected, and then in the details pane double-click **Enable Automatic Hosted Cache Discovery by Service Connection Point**.

The policy setting dialog box opens.

9. In the **Enable Automatic Hosted Cache Discovery by Service Connection Point** dialog box, click **Enabled**, and then click **OK**.
10. To enable client computers to download and cache content from BranchCache file server-based content servers: In the Group Policy Management Editor console, ensure that **BranchCache** is still selected, and then in the details pane double-click **BranchCache for network files**. The **Configure BranchCache for network files** dialog box opens.
11. In the **Configure BranchCache for network files** dialog box, click **Enabled**. In **Options**, type a numeric value, in milliseconds, for the maximum round trip network latency time, and then click **OK**.

NOTE

By default, client computers cache content from file servers if the round trip network latency is longer than 80 milliseconds.

12. To configure the amount of hard disk space allocated on each client computer for the BranchCache cache: In the Group Policy Management Editor console, ensure that **BranchCache** is still selected, and then in the details pane double-click **Set percentage of disk space used for client computer cache**. The **Set percentage of disk space used for client computer cache** dialog box opens. Click **Enabled**, and then in **Options** type a numeric value that represents the percentage of hard disk space used on each client computer for the BranchCache cache. Click **OK**.
13. To specify the default age, in days, for which segments are valid in the BranchCache data cache on client computers: In the Group Policy Management Editor console, ensure that **BranchCache** is still selected, and then in the details pane double-click **Set age for segments in the data cache**. The **Set age for segments in the data cache** dialog box opens. Click **Enabled**, and then in the details pane type the number of days that you prefer. Click **OK**.
14. Configure additional BranchCache policy settings for client computers as appropriate for your deployment.
15. Refresh Group Policy on branch office client computers by running the command **gpupdate /force**, or by rebooting the client computers.

Your BranchCache Hosted Cache mode deployment is now complete.

For additional information on the technologies in this guide, see [Additional Resources](#).

BranchCache Additional Resources

9/1/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

For more information about the technologies that are discussed in this guide, see the following resources:

- [BranchCache in Windows Server 2016](#)
- [Install and Configure Content Servers](#)
- [BranchCache Network Shell and Windows PowerShell Commands](#)
- [Group Policy Overview for Windows Server 2012 R2](#)
- Windows Server 2008 R2 [BranchCache Deployment Guide](#)

BranchCache

9/1/2018 • 31 minutes to read • [Edit Online](#)

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

This topic, which is intended for Information Technology (IT) professionals, provides overview information about BranchCache, including BranchCache modes, features, capabilities, and the BranchCache functionality that is available in different operating systems.

NOTE

In addition to this topic, the following BranchCache documentation is available.

- [BranchCache Network Shell and Windows PowerShell Commands](#)
- [BranchCache Deployment Guide](#)

Who will be interested in BranchCache?

If you are a system administrator, network or storage solution architect, or other IT professional, BranchCache might interest you under the following circumstances:

- You design or support IT infrastructure for an organization that has two or more physical locations and a wide area network (WAN) connection from the branch offices to the main office.
- You design or support IT infrastructure for an organization that has deployed cloud technologies, and a WAN connection is used by workers to access data and applications at remote locations.
- You want to optimize WAN bandwidth usage by reducing the amount of network traffic between branch offices and the main office.
- You have deployed or are planning on deploying content servers at your main office that match the configurations that are described in this topic.
- The client computers in your branch offices are running Windows 10, Windows 8.1, Windows 8, or Windows 7 .

This topic includes the following sections:

- [What is BranchCache?](#)
- [BranchCache modes](#)
- [BranchCache-enabled content servers](#)
- [BranchCache and the cloud](#)
- [Content information versions](#)
- [How BranchCache handles content updates in files](#)
- [BranchCache installation guide](#)
- [Operating system versions for BranchCache](#)
- [BranchCache security](#)

- Content flow and processes
- Cache Security

What is BranchCache?

BranchCache is a wide area network (WAN) bandwidth optimization technology that is included in some editions of the Windows Server 2016 and Windows 10 operating systems, as well as in some editions of Windows Server 2012 R2, Windows 8.1, Windows Server 2012, Windows 8, Windows Server 2008 R2 and Windows 7. To optimize WAN bandwidth when users access content on remote servers, BranchCache fetches content from your main office or hosted cloud content servers and caches the content at branch office locations, allowing client computers at branch offices to access the content locally rather than over the WAN.

At branch offices, content is stored either on servers that are configured to host the cache or, when no server is available in the branch office, on client computers that are running Windows 10, Windows 8.1, Windows 8 or Windows 7. After a client computer requests and receives content from the main office and the content is cached at the branch office, other computers at the same branch office can obtain the content locally rather than downloading the content from the content server over the WAN link.

When subsequent requests for the same content are made by client computers, the clients download *content information* from the server instead of the actual content. Content information consists of hashes that are calculated using chunks of the original content, and are extremely small compared to the content in the original data. Client computers then use the content information to locate the content from a cache in the branch office, whether the cache is located on a client computer or on a server. Client computers and servers also use content information to secure cached content so that it cannot be accessed by unauthorized users.

BranchCache increases end user productivity by improving content query response times for clients and servers in branch offices, and can also help improve network performance by reducing traffic over WAN links.

BranchCache modes

BranchCache has two modes of operation: distributed cache mode and hosted cache mode.

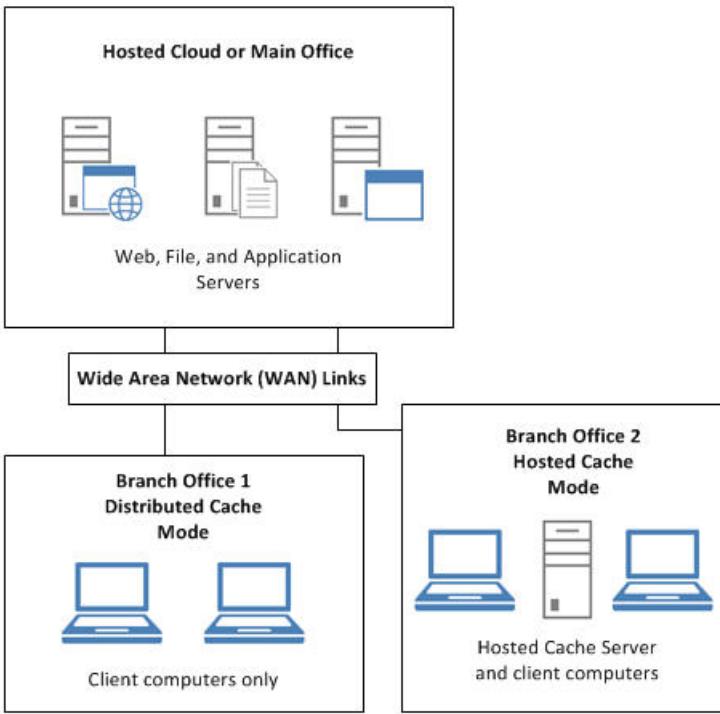
When you deploy BranchCache in distributed cache mode, the content cache at a branch office is distributed among client computers.

When you deploy BranchCache in hosted cache mode, the content cache at a branch office is hosted on one or more server computers, which are called hosted cache servers.

NOTE

You can deploy BranchCache using both modes, however only one mode can be used per branch office. For example, if you have two branch offices, one which has a server and one which does not, you can deploy BranchCache in hosted cache mode in the office that contains a server, while deploying BranchCache in distributed cache mode in the office that contains only client computers.

In the following illustration, BranchCache is deployed in both modes.



Distributed cache mode is best suited for small branch offices that do not contain a local server for use as a hosted cache server. Distributed cache mode allows you to deploy BranchCache with no additional hardware in branch offices.

If the branch office where you want to deploy BranchCache contains additional infrastructure, such as one or more servers that are running other workloads, deploying BranchCache in hosted cache mode is beneficial for the following reasons:

Increased cache availability

Hosted cache mode increases the cache efficiency because content is available even if the client that originally requested and cached the data is offline. Because the hosted cache server is always available, more content is cached, providing greater WAN bandwidth savings, and BranchCache efficiency is improved.

Centralized caching for multiple-subnet branch offices

Distributed cache mode operates on a single subnet. At a multiple-subnet branch office that is configured for distributed cache mode, a file downloaded to one subnet cannot be shared with client computers on other subnets.

Because of this, clients on other subnets, unable to discover that the file has already been downloaded, get the file from the main office content server, using WAN bandwidth in the process.

When you deploy hosted cache mode, however, this is not the case - all clients in a multiple-subnet branch office can access a single cache, which is stored on the hosted cache server, even if the clients are on different subnets. In addition, BranchCache in Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012 provides the ability to deploy more than one hosted cache server per branch office.

Caution

If you use BranchCache for SMB caching of files and folders, do not disable Offline Files. If you disable Offline Files, BranchCache SMB caching does not function correctly.

BranchCache-enabled content servers

When you deploy BranchCache, the source content is stored on BranchCache-enabled content servers in your main office or in a cloud data center. The following types of content servers are supported by BranchCache:

NOTE

Only source content - that is, content that client computers initially obtain from a BranchCache-enabled content server - is accelerated by BranchCache. Content that client computers obtain directly from other sources, such as Web servers on the Internet or Windows Update, is not cached by client computers or hosted cache servers and then shared with other computers in the branch office. If you want to accelerate Windows Update content, however, you can install a Windows Server Update Services (WSUS) application server at your main office or cloud data center and configure it as a BranchCache content server.

Web servers

Supported Web servers include computers that are running Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2 that have the Web Server (IIS) server role installed and that use Hypertext Transfer Protocol (HTTP) or HTTP Secure (HTTPS).

In addition, the Web server must have the BranchCache feature installed.

File servers

Supported file servers include computers that are running Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2 that have the File Services server role and the BranchCache for Network Files role service installed.

These file servers use Server Message Block (SMB) to exchange information between computers. After you complete installation of your file server, you must also share folders and enable hash generation for shared folders by using Group Policy or Local Computer Policy to enable BranchCache.

Application servers

Supported application servers include computers that are running Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2 with Background Intelligent Transfer Service (BITS) installed and enabled.

In addition, the application server must have the BranchCache feature installed. As examples of application servers, you can deploy Microsoft Windows Server Update Services (WSUS) and Microsoft System Center Configuration Manager Branch Distribution Point servers as BranchCache content servers.

BranchCache and the cloud

The cloud has enormous potential to reduce operational expenses and achieve new levels of scale, but moving workloads away from the people who depend on them can increase networking costs and hurt productivity. Users expect high performance and don't care where their applications and data are hosted.

BranchCache can improve the performance of networked applications and reduce bandwidth consumption with a shared cache of data. It improves productivity in branch offices and in headquarters, where workers are using servers that are deployed in the cloud.

Because BranchCache does not require new hardware or network topology changes, it is an excellent solution for improving communication between office locations and both public and private clouds.

NOTE

Because some Web proxies cannot process non-standard Content-Encoding headers, it is recommended that you use BranchCache with Hyper Text Transfer Protocol Secure (HTTPS) and not HTTP.

===== For more information about cloud technologies in Windows Server 2016, see [Software Defined Networking \(SDN\)](#).

Content information versions

There are two versions of content information:

- Content information that is compatible with computers running Windows Server 2008 R2 and Windows 7 is called version 1, or V1. With V1 BranchCache file segmentation, file segments are larger than in V2 and are of fixed size. Because of large fixed segment sizes, when a user makes a change that modifies the file length, not only is the segment with the change invalidated, but all of the segments to the end of the file are invalidated. The next call for the changed file by another user in the branch office therefore results in reduced WAN bandwidth savings because the changed content and all content after the change are sent over the WAN link.
- Content information that is compatible with computers running Windows Server 2016, Windows 10, Windows Server 2012 R2, Windows 8.1, Windows Server 2012, and Windows 8 is called version 2, or V2. V2 content information uses smaller, variable-sized segments that are more tolerant to changes within a file. This increases the probability that segments from an older version of the file can be reused when users access an updated version, causing them to retrieve only the changed portion of the file from the content server, and using less WAN bandwidth.

The following table provides information on the content information version that is used depending upon which client, content server, and hosted cache server operating systems you are using in your BranchCache deployment.

NOTE

In the table below, the acronym "OS" means operating system.

CLIENT OS	CONTENT SERVER OS	HOSTED CACHE SERVER OS	CONTENT INFORMATION VERSION
Windows Server 2008 R2 and Windows 7	Windows Server 2012 or later	Windows Server 2012 or later; none for distributed cache mode	V1
Windows Server 2012 or later; Windows 8 or later	Windows Server 2008 R2	Windows Server 2012 or later; none for distributed cache mode	V1
Windows Server 2012 or later; Windows 8 or later	Windows Server 2012 or later	Windows Server 2008 R2	V1
Windows Server 2012 or later; Windows 8 or later	Windows Server 2012 or later	Windows Server 2012 or later; none for distributed cache mode	V2

When you have content servers and hosted cache servers that are running Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012, they use the content information version that is appropriate based on the operating system of the BranchCache client that requests information.

When computers running Windows Server 2012 and Windows 8 or later operating systems request content, the content and hosted cache servers use V2 content information; when computers running Windows Server 2008 R2 and Windows 7 request content, the content and hosted cache servers use V1 content information.

IMPORTANT

When you deploy BranchCache in distributed cache mode, clients that use different content information versions do not share content with each other. For example, a client computer running Windows 7 and a client computer running Windows 10 that are installed in the same branch office do not share content with each other.

How BranchCache handles content updates in files

When branch office users modify or update the contents of documents, their changes are written directly to the content server in the main office without BranchCache's involvement. This is true whether the user downloaded the document from the content server or obtained it from either a hosted or distributed cache in the branch office.

When the modified file is requested by a different client in a branch office, the new segments of the file are downloaded from the main office server and added to the distributed or hosted cache in that branch. Because of this, branch office users always receive the most recent versions of cached content.

BranchCache installation guide

You can use Server Manager in Windows Server 2016 to install either the BranchCache feature or the BranchCache for Network Files role service of the File Services server role. You can use the following table to determine whether to install the role service or the feature.

FUNCTIONALITY	COMPUTER LOCATION	INSTALL THIS BRANCHCACHE ELEMENT
Content server (BITS-based application server)	Main office or cloud data center	BranchCache feature
Content server (Web server)	Main office or cloud data center	BranchCache feature
Content server (file server using the SMB protocol)	Main office or cloud data center	BranchCache for Network Files role service of the File Services server role
Hosted cache server	Branch office	BranchCache feature with hosted cache server mode enabled
BranchCache-enabled client computer	Branch office	No installation needed; just enable BranchCache and a BranchCache mode (distributed or hosted) on the client

To install either the role service or the feature, open Server Manager and select the computers where you want to enable BranchCache functionality. In Server Manager, click **Manage**, and then click **Add Roles and Features**. The **Add Roles and Features** wizard opens. As you run the wizard, make the following selections:

- On the wizard page **Select Installation Type**, select **Role-based or Feature-based Installation**.
- On the wizard page **Select Server Roles**, if you are installing a BranchCache-enabled file server, expand **File and Storage Services** and **File and iSCSI Services**, and then select **BranchCache for Network Files**. To save disk space, you can also select the **Data Deduplication** role service, and then continue through the wizard to installation and completion. If you do not want to install a BranchCache-enabled file server, do not install the File and Storage Services role with the BranchCache for Network Files role service.
- On the wizard page **Select features**, if you are installing a content server that is not a file server or you are installing a hosted cache server, select **BranchCache**, and then continue through the wizard to installation and completion. If you do not want to install a content server other than a file server or a hosted cache server, do not install the BranchCache feature.

Operating system versions for BranchCache

Following is a list of operating systems that support different types of BranchCache functionality.

Operating systems for BranchCache client computer functionality

The following operating systems provide BranchCache with support for Background Intelligent Transfer Service (BITS), Hyper Text Transfer Protocol (HTTP), and Server Message Block (SMB).

- Windows 10 Enterprise
- Windows 10 Education
- Windows 8.1 Enterprise
- Windows 8 Enterprise
- Windows 7 Enterprise
- Windows 7 Ultimate

In the following operating systems, BranchCache does not support HTTP and SMB functionality, but does support BranchCache BITS functionality.

- Windows 10 Pro, BITS support only
- Windows 8.1 Pro, BITS support only
- Windows 8 Pro, BITS support only
- Windows 7 Pro, BITS support only

NOTE

BranchCache is not available by default in the Windows Server 2008 or Windows Vista operating systems. On these operating systems, however, if you download and install the Windows Management Framework update, BranchCache functionality is available for the Background Intelligent Transfer Service (BITS) protocol only. For more information, and to download Windows Management Framework, see [Windows Management Framework \(Windows PowerShell 2.0, WinRM 2.0, and BITS 4.0\)](#) at <https://go.microsoft.com/fwlink/?LinkId=188677>.

Operating systems for BranchCache content server functionality

You can use the Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012 families of operating systems as BranchCache content servers.

In addition, the Windows Server 2008 R2 family of operating systems can be used as BranchCache content servers, with the following exceptions:

- BranchCache is not supported in Server Core installations of Windows Server 2008 R2 Enterprise with Hyper-V.
- BranchCache is not supported in Server Core installations of Windows Server 2008 R2 Datacenter with Hyper-V.

Operating systems for BranchCache hosted cache server functionality

You can use the Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012 families of operating systems as BranchCache hosted cache servers.

In addition, the following Windows Server 2008 R2 operating systems can be used as BranchCache hosted cache servers:

- Windows Server 2008 R2 Enterprise
- Windows Server 2008 R2 Enterprise with Hyper-V
- Windows Server 2008 R2 Enterprise Server Core Installation
- Windows Server 2008 R2 Enterprise Server Core Installation with Hyper-V
- Windows Server 2008 R2 for Itanium-Based Systems
- Windows Server 2008 R2 Datacenter
- Windows Server 2008 R2 Datacenter with Hyper-V
- Windows Server 2008 R2 Datacenter Server Core Installation with Hyper-V

BranchCache Security

BranchCache implements a secure-by-design approach that works seamlessly alongside your existing network security architectures, without the requirement for additional equipment or complex additional security configuration.

BranchCache is non-invasive and does not alter any Windows authentication or authorization processes. After you deploy BranchCache, authentication is still performed using domain credentials, and the way in which authorization with Access Control Lists (ACLs) functions is unchanged. In addition, other configurations continue to function just as they did before BranchCache deployment.

The BranchCache security model is based on the creation of metadata, which takes the form of a series of hashes. These hashes are also called content information.

After content information is created, it is used in BranchCache message exchanges rather than the actual data, and it is exchanged using the supported protocols (HTTP, HTTPS, and SMB).

Cached data is kept encrypted and cannot be accessed by clients that do not have permission to access content from the original source. Clients must be authenticated and authorized by the original content source before they can retrieve content metadata, and must possess content metadata to access the cache in the local office.

How BranchCache generates content information

Because content information is created from multiple elements, the value of the content information is always unique. These elements are:

- The actual content (such as Web pages or shared files) from which the hashes are derived.
- Configuration parameters, such as the hashing algorithm and block size. To generate content information, the content server divides the content into segments and then subdivides those segments into blocks. BranchCache uses secure cryptographic hashes to identify and verify each block and segment, supporting the SHA256 hash algorithm.
- A server secret. All content servers must be configured with a server secret, which is a binary value of arbitrary length.

NOTE

The use of a server secret ensures that client computers are not able to generate the content information themselves. This prevents malicious users from using brute force attacks with BranchCache-enabled client computers to guess minor changes in content across versions in situations in which the client had access to a previous version but does not have access to the current version.

Content information details

BranchCache uses the server secret as a key in order to derive a content-specific hash that is sent to authorized clients. Applying a hashing algorithm to the combined server secret and the Hash of Data generates this hash.

This hash is called the segment secret. BranchCache uses segment secrets to secure communications. In addition, BranchCache creates a Block Hash List, which is list of hashed data blocks, and the Hash of Data, which is generated by hashing the Block Hash List.

The content information includes the following:

- The Block Hash List:

```
BlockHashi = Hash(dataBlocki) 1<=i<=n
```

- The Hash of Data (HoD):

```
HoD = Hash(BlockHashList)
```

- Segment Secret (Kp):

```
Kp = HMAC(Ks, HoD)
```

BranchCache uses the Peer Content Caching protocol and the Retrieval Framework protocol to implement the processes that are required to ensure the secure caching and retrieval of data between content caches.

In addition, BranchCache handles content information with the same degree of security that it uses when handling and transmitting the actual content itself.

Content flow and processes

The flow of content information and actual content is divided into four phases:

1. [BranchCache processes: Request content](#)
2. [BranchCache processes: Locate content](#)
3. [BranchCache processes: Retrieve content](#)
4. [BranchCache processes: Cache content](#)

The following sections describe these phases.

BranchCache processes: Request content

In the first phase, the client computer in the branch office requests content, such as a file or a Web page, from a content server in a remote location, such as a main office. The content server verifies that the client computer is authorized to receive the requested content. If the client computer is authorized and both content server and client are BranchCache-enabled, the content server generates content information.

The content server then sends the content information to the client computer using the same protocol as would have been used for the actual content.

For example, if the client computer requested a Web page over HTTP, the content server sends the content information using HTTP. Because of this, the wire-level security guarantees of the content and the content information are identical.

After the initial portion of content information (Hash of Data + Segment Secret) is received, the client computer performs the following actions:

- Uses the Segment Secret (Kp) as the encryption key (Ke).
- Generates the Segment ID (HoHoDk) from the HoD and Kp:

$\text{HoHoDk} = \text{HMAC}(K_p, \text{HoD} + C)$, where C is the ASCII string "MS_P2P_CACHING" with NUL terminator.

The primary threat at this layer is the risk to the Segment Secret, however BranchCache encrypts the content data blocks to protect the Segment Secret. BranchCache does this by using the encryption key that is derived from the Segment Secret of the content segment within which the content blocks are located.

This approach ensures that an entity that is not in possession of the server secret cannot discover the actual content in a data block. The Segment Secret is treated with the same degree of security as the plaintext segment itself, because knowledge of the Segment Secret for a given segment enables an entity to obtain the segment from peers and then decrypt it. Knowledge of the Server Secret does not immediately yield any particular plaintext but can be used to derive certain types of data from the cipher text and then to possibly expose some partially known data to a brute-force guessing attack. The server secret, therefore, should be kept confidential.

BranchCache processes: Locate content

After the content information is received by the client computer, the client uses the Segment ID to locate the requested content in the local branch office cache, whether that cache is distributed between client computers or is located on a hosted cache server.

If the client computer is configured for hosted cache mode, it is configured with the computer name of the hosted cache server and contacts that server to retrieve the content.

If the client computer is configured for distributed cache mode, however, the content might be stored across multiple caches on multiple computers in the branch office. The client computer must discover where the content is located before the content is retrieved.

When they are configured for distributed cache mode, client computers locate content by using a discovery protocol that is based on the Web Services Dynamic Discovery (WS-Discovery) protocol. Clients send WS-Discovery multicast Probe messages to discover cached content over the network. Probe messages include the Segment ID, which enables clients to check whether the requested content matches the content stored in their cache. Clients that receive the initial Probe message reply to the querying client with unicast Probe-Match messages if the Segment ID matches content that is cached locally.

The success of the WS-Discovery process depends on the fact that the client that is performing the discovery has the correct content information, which was provided by the content server, for the content that it is requesting.

The main threat to data during the Request content phase is information disclosure, because access to the content information implies authorized access to content. To mitigate this risk, the discovery process does not reveal the content information, other than the Segment ID, which does not reveal anything about the plaintext segment that contains the content.

In addition, another client computer run by a malicious user on the same network subnet can see the BranchCache discovery traffic to the original content source going through the router.

If the requested content is not found in the branch office, the client requests the content directly from the content server across the WAN link.

After the content is received, it is added to the local cache, either on the client computer or on a hosted cache server. In this case, the content information prevents a client or hosted cache server from adding to the local cache any content that does not match the hashes. The process of verifying content by matching hashes ensures that only valid content is added to the cache, and the integrity of the local cache is protected.

BranchCache processes: Retrieve content

After a client computer locates the desired content on the content host, which is either a hosted cache server or a distributed cache mode client computer, the client computer begins the process of retrieving the content.

First the client computer sends a request to the content host for the first block that it requires. The request contains the Segment ID and block range that identify the desired content. Because only one block is returned, the block range contains only a single block. (Requests for multiple blocks are currently not supported.) The client also stores the request in its local Outstanding Request List.

Upon receiving a valid request message from a client, the content host checks whether the block specified in the request exists in the content host's content cache.

If the content host is in possession of the content block, then the content host sends a response that contains the Segment ID, the Block ID, the encrypted data block, and the initialization vector that is used for encrypting the block.

If the content host is not in possession of the content block, the content host sends an empty response message. This informs the client computer that the content host does not have the requested block. An empty response message contains the Segment ID and Block ID of the requested block, along with a zero-sized data block.

When the client computer receives the response from the content host, the client verifies that the message corresponds to a request message in its Outstanding Request List. (The Segment ID and block index must match that of an outstanding request.)

If this verification process is unsuccessful and the client computer does not have a corresponding request message in its Outstanding Request List, the client computer discards the message.

If this verification process is successful and the client computer has a corresponding request message in its Outstanding Request List, the client computer decrypts the block. The client then validates the decrypted block against the appropriate block hash from the content information that the client initially obtained from the original content server.

If the block validation is successful, the decrypted block is stored in the cache.

This process is repeated until the client has all of the required blocks.

NOTE

If the complete segments of content do not exist on one computer, the retrieval protocol retrieves and assembles content from a combination of sources: a set of distributed cache mode client computers, a hosted cache server, and - if the branch office caches do not contain the complete content - the original content server in the main office.

Before BranchCache sends content information or content, the data is encrypted. BranchCache encrypts the block in the response message. In Windows 7, the default encryption algorithm that BranchCache uses is AES-128, the encryption key is Ke, and the key size is 128 bits, as dictated by the encryption algorithm.

BranchCache generates an initialization vector that is suitable for the encryption algorithm and uses the encryption key to encrypt the block. BranchCache then records the encryption algorithm and the initialization vector in the message.

Servers and clients never exchange, share, or send each other the encryption key. The client receives the encryption key from the content server that hosts the source content. Then, using the encryption algorithm and initialization vector it received from the server, it decrypts the block. There is no other explicit authentication or authorization built into the download protocol.

Security threats

The primary security threats at this layer include:

- Tampering with data:

A client serving data to a requester tampers with the data. The BranchCache security model uses hashes to

confirm that neither the client nor the server has altered the data.

- Information disclosure:

BranchCache sends encrypted content to any client that specifies the appropriate Segment ID. Segment IDs are public, so any client can receive encrypted content. However, if a malicious user obtains encrypted content, they must know the encryption key to decrypt the content. The upper layer protocol performs authentication and then gives the content information to the authenticated and authorized client. The security of the content information is equivalent to the security provided to the content itself, and BranchCache never exposes the content information.

An attacker sniffs the wire to obtain the content. BranchCache encrypts all transfers between clients by using AES128 where the secret key is Ke, preventing data from being sniffed from the wire. Content information that is downloaded from the content server is protected in exactly the same way as the data itself would have been and is hence no more or less protected from information disclosure than if BranchCache had not been used at all.

- Denial of Service:

A client is overwhelmed by requests for data. BranchCache protocols incorporate queue management counters and timers to prevent clients from being overloaded.

BranchCache processes: Cache content

On distributed cache mode client computers and hosted cache servers that are located in branch offices, content caches are built up over time as content is retrieved over WAN links.

When client computers are configured with hosted cache mode, they add content to their own local cache and also offer data to the hosted cache server. The Hosted Cache Protocol provides a mechanism for clients to inform the hosted cache server about content and segment availability.

To upload content to the hosted cache server, the client informs the server that it has a segment that is available. The hosted cache server then retrieves all of the content information that is associated with the offered segment, and downloads the blocks within the segment that it actually needs. This process is repeated until the client has no more segments to offer the hosted cache server.

To update the hosted cache server by using the Hosted Cache Protocol, the following requirements must be met:

- The client computer is required to have a set of blocks within a segment that it can offer to the hosted cache server. The client must supply content information for the offered segment; this is comprised of the Segment ID, the segment Hash of Data, the Segment Secret, and a list of all block hashes that are contained within the segment.
- For hosted cache servers that are running Windows Server 2008 R2, a hosted cache server certificate and associated private key are required, and the certification authority (CA) that issued the certificate must be trusted by client computers in the branch office. This allows the client and server to participate successfully in HTTPS Server authentication.

IMPORTANT

Hosted cache servers that are running Windows Server 2016, Windows Server 2012 R2 , or Windows Server 2012 do not require a hosted cache server certificate and associated private key.

- The client computer is configured with the computer name of the hosted cache server and the Transmission Control Protocol (TCP) port number upon which the hosted cache server is listening for BranchCache traffic. The hosted cache server's certificate is bound to this port. The computer name of the hosted cache server can be a fully qualified domain name (FQDN), if the hosted cache server is a domain member computer; or

it can be the NetBIOS name of the computer if the hosted cache server is not a domain member.

- The client computer actively listens for incoming block requests. The port on which it is listening is passed as part of the offer messages from the client to the hosted cache server. This enables the hosted cache server to use BranchCache protocols to connect to the client computer to retrieve data blocks in the segment.
- The hosted cache server starts to listen for incoming HTTP requests when it is initialized.
- If the hosted cache server is configured to require client computer authentication, both the client and the hosted cache server are required to support HTTPS authentication.

Hosted cache mode cache population

The process of adding content to the hosted cache server's cache in a branch office begins when the client sends an INITIAL_OFFER_MESSAGE, which includes the Segment ID. The Segment ID in the INITIAL_OFFER_MESSAGE request is used to retrieve the corresponding segment Hash of Data, list of block hashes, and the Segment Secret from the hosted cache server's block cache. If the hosted cache server already has all the content information for a particular segment, the response to the INITIAL_OFFER_MESSAGE will be OK, and no request to download blocks occurs.

If the hosted cache server does not have all of the offered data blocks that are associated with the block hashes in the segment, the response to the INITIAL_OFFER_MESSAGE is INTERESTED. The client then sends a SEGMENT_INFO_MESSAGE that describes the single segment that is being offered. The hosted cache server responds with an OK message and initiates the download of the missing blocks from the offering client computer.

The segment Hash of Data, list of block hashes, and the segment secret are used to ensure that the content that is being downloaded has not been tampered with or otherwise altered. The downloaded blocks are then added to the hosted cache server's block cache.

Cache Security

This section provides information on how BranchCache secures cached data on client computers and on hosted cache servers.

Client computer cache security

The greatest threat to data stored in the BranchCache is tampering. If an attacker can tamper with content and content information that is stored in the cache, then it might be possible to use this to try and launch an attack against the computers that are using BranchCache. Attackers can initiate an attack by inserting malicious software in place of other data. BranchCache mitigates this threat by validating all content using block hashes found in the content information. If an attacker attempts to tamper with this data, it is discarded and is replaced with valid data from the original source.

A secondary threat to data stored in the BranchCache is information disclosure. In distributed cache mode, the client caches only the content that it has requested itself; however, that data is stored in clear text, and might be at risk. To help restrict cache access to the BranchCache Service only, the local cache is protected by file system permissions that are specified in an ACL.

Although the ACL is effective in preventing unauthorized users from accessing the cache, it is possible for a user with administrative privileges to gain access to the cache by manually changing the permissions that are specified in the ACL. BranchCache does not protect against the malicious use of an administrative account.

Data that is stored in the content cache is not encrypted, so if data leakage is a concern, you can use encryption technologies such as BitLocker or the Encrypting File System (EFS). The local cache that is used by BranchCache does not increase the information disclosure threat borne by a computer in the branch office; the cache contains only copies of files that reside unencrypted elsewhere on the disk.

Encrypting the entire disk is particularly important in environments in which the physical security of the clients is

difficult to ensure. For example, encrypting the entire disk helps to secure sensitive data on mobile computers that might be removed from the branch office environment.

Hosted cache server cache security

In hosted cache mode, the greatest threat to the security of the hosted cache server is information disclosure. BranchCache in a hosted cache environment behaves in a similar manner to distributed cache mode, with file system permission protecting the cached data. The difference is that the hosted cache server stores all of the content that any BranchCache-enabled computer in the branch office requests, rather than just the data that a single client requests. The consequences of unauthorized intrusion into this cache could be much more serious, because much more data is at risk.

In a hosted cache environment where the hosted cache server is running Windows Server 2008 R2, the use of encryption technologies such as BitLocker or EFS is advisable if any of the clients in the branch office can access sensitive data across the WAN link. It is also necessary to prevent physical access to the hosted cache, because disk encryption works only when the computer is turned off when the attacker gains physical access. If the computer is turned on or is in sleep mode, then disk encryption offers little protection.

NOTE

Hosted cache servers that are running Windows Server 2016, Windows Server 2012 R2, or Windows Server 2012 encrypt all data in the cache by default, so the use of additional encryption technologies is not required.

Even if a client is configured in hosted cache mode, it will still cache data locally, and you might want to take steps to protect the local cache in addition to the cache on the hosted cache server.

BranchCache Network Shell and Windows PowerShell Commands

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

In Windows Server 2016, you can configure and manage BranchCache by using either Windows PowerShell or the Network Shell (Netsh) commands for BranchCache.

In future versions of Windows, Microsoft might remove the netsh functionality for BranchCache. Microsoft recommends that you transition to Windows PowerShell if you currently use netsh to configure and manage BranchCache and other networking technologies.

Windows PowerShell and netsh command references are at the following locations. Although both command references were published for operating systems earlier than Windows Server 2016, these references are accurate for this operating system.

- [Netsh Commands for BranchCache in Windows Server 2008 R2](#)
- [BranchCache Cmdlets in Windows PowerShell for Windows Server 2012 .](#)

TIP

To view a list of Windows PowerShell commands for BranchCache at the Windows PowerShell prompt, type

```
Get-Command -Module BranchCache
```

at the Windows PowerShell prompt, and then press ENTER.

BranchCache Deployment Guide

9/1/2018 • 5 minutes to read • [Edit Online](#)

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

You can use this guide to learn how to deploy BranchCache in Windows Server 2016.

In addition to this topic, this guide contains the following sections.

- [Choosing a BranchCache Design](#)
- [Deploy BranchCache](#)

BranchCache Deployment Overview

BranchCache is a wide area network (WAN) bandwidth optimization technology that is included in some editions of Windows Server 2016, Windows Server® 2012 R2, Windows Server® 2012, Windows Server® 2008 R2, and related Windows client operating systems.

To optimize WAN bandwidth, BranchCache copies content from your main office content servers and caches the content at branch office locations, allowing client computers at branch offices to access the content locally rather than over the WAN.

At branch offices, content is cached either on servers that are running the BranchCache feature of Windows Server 2016, Windows Server 2012 R2 , Windows Server 2012 , or Windows Server 2008 R2 - or, if there are no servers available in the branch office, content is cached on client computers that are running Windows 10®, Windows® 8.1, Windows 8, or Windows 7® .

After a client computer requests and receives content from the main office or cloud datacenter and the content is cached at the branch office, other computers at the same branch office can obtain the content locally rather than contacting the content server over the WAN link.

Benefits of deploying BranchCache

BranchCache caches file, web, and application content at branch office locations, allowing client computers to access data using the local area network (LAN) rather than accessing the content over slow WAN connections.

BranchCache reduces both WAN traffic and the time that is required for branch office users to open files on the network. BranchCache always provides users with the most recent data, and it protects the security of your content by encrypting the caches on the hosted cache server and on client computers.

What this guide provides

This deployment guide allows you to deploy BranchCache in the following modes:

- Distributed cache mode. In this mode, branch office client computers download content from the content servers in the main office or cloud, and then cache the content for other computers in the same branch office. Distributed cache mode does not require a server computer in the branch office.
- Hosted cache mode. In this mode, branch office client computers download content from the content servers in the main office or cloud, and a hosted cache server retrieves the content from the clients. The hosted cache server then caches the content for other client computers.

This guide also provides instructions on how to deploy three types of content servers. Content servers contain the source content that is downloaded by branch office client computers, and one or more content server is required to

deploy BranchCache in either mode. The content server types are:

- **Web server-based content servers.** These content servers send content to BranchCache client computers using the HTTP and HTTPS protocols. These content servers must be running Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2 versions that support BranchCache and upon which the BranchCache feature is installed.
- **BITS-based application servers.** These content servers send content to BranchCache client computers using the Background Intelligent Transfer Service (BITS). These content servers must be running Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2 versions that support BranchCache and upon which the BranchCache feature is installed.
- **File server-based content servers.** These content servers must be running Windows Server 2016, Windows Server 2012 R2 , Windows Server 2012 , or Windows Server 2008 R2 versions that support BranchCache and upon which the File Services server role is installed. In addition, the **BranchCache for network files** role service of the File Services server role must be installed and configured. These content servers send content to BranchCache client computers using the Server Message Block (SMB) protocol.

For more information, see [Operating system versions for BranchCache](#).

BranchCache deployment requirements

Following are the requirements for deploying BranchCache by using this guide.

- **File and Web content servers** must be running one of the following operating systems to provide BranchCache functionality: Windows Server 2016, Windows Server 2012 R2 , Windows Server 2012 , or Windows Server 2008 R2 . Windows 8 and later clients continue to see benefits from BranchCache when accessing content servers that are running Windows Server 2008 R2 , however they are unable to make use of the new chunking and hashing technologies in Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012.
- **Client computers** must be running Windows 10, Windows 8.1, or Windows 8 to make use of the most recent deployment model and the chunking and hashing improvements that were introduced with Windows Server 2012 .
- **Hosted cache servers** must be running Windows Server 2016, Windows Server 2012 R2, or Windows Server 2012 to make use of the deployment improvements and scale features described in this document. A computer that is running one of these operating systems that is configured as a hosted cache server can continue to serve client computers that are running Windows 7 , but to do so, it must be equipped with a certificate that is suitable for Transport Layer Security (TLS), as described in the Windows Server 2008 R2 and Windows 7 [BranchCache Deployment Guide](#).
- **An Active Directory domain** is required to take advantage of Group Policy and hosted cache automatic discovery, but a domain is not required to use BranchCache. You can configure individual computers by using Windows PowerShell. In addition, it is not required that your domain controllers are running Windows Server 2012 or later to utilize new BranchCache Group Policy settings; you can import the BranchCache administrative templates onto domain controllers that are running earlier operating systems, or you can author the group policy objects remotely on other computers that are running Windows 10, Windows Server 2016, Windows 8.1, Windows Server 2012 R2, Windows 8, or Windows Server 2012.
- **Active Directory sites** are used to limit the scope of hosted cache servers that are automatically discovered. To automatically discover a hosted cache server, both the client and server computers must belong to the same site. BranchCache is designed to have a minimal impact on clients and servers and does not impose additional hardware requirements beyond those needed to run their respective operating systems.

BranchCache history and documentation

BranchCache was first introduced in Windows 7® and Windows Server® 2008 R2, and was improved in Windows Server 2012, Windows 8, and later operating systems.

NOTE

If you are deploying BranchCache in operating systems other than Windows Server 2016, the following documentation resources are available.

- For information about BranchCache in Windows 8, Windows 8.1, Windows Server 2012, and Windows Server 2012 R2, see [BranchCache Overview](#).
- For information about BranchCache in Windows 7 and Windows Server 2008 R2, see [BranchCache for Windows Server 2008 R2](#).

Choosing a BranchCache Design

9/1/2018 • 2 minutes to read • [Edit Online](#)

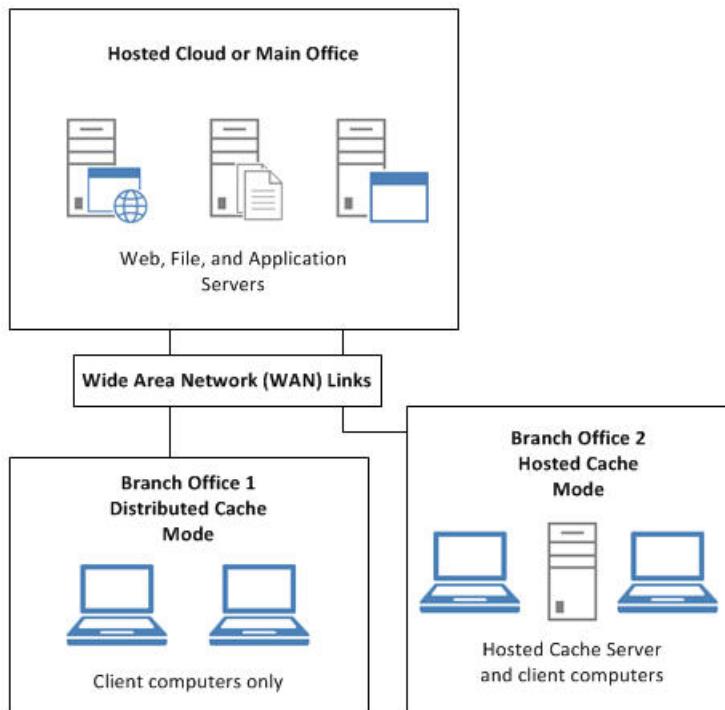
Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016

You can use this topic to learn about BranchCache modes and to select the best modes for your deployment.

You can use this guide to deploy BranchCache in the following modes and mode combinations.

- All branch offices are configured for distributed cache mode.
- All branch offices are configured for hosted cache mode and have a hosted cache server on site.
- Some branch offices are configured for distributed cache mode and some branch offices have a hosted cache server on site and are configured for hosted cache mode.

The following illustration depicts a dual mode installation, with one branch office configured for distributed cache mode and one branch office configured for hosted cache mode.



Before you deploy BranchCache, select the mode you prefer for each branch office in your organization.

Deploy BranchCache

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

The following sections provide information about deploying BranchCache in distributed and hosted cache modes.

- [Install and Configure Content Servers](#)
- [Deploy Hosted Cache Servers \(Optional\)](#)
- [Prehashing and Preloading Content on Hosted Cache Servers \(Optional\)](#)
- [Configure BranchCache Client Computers](#)

NOTE

The procedures in this guide do not include instructions for those cases in which the **User Account Control** dialog box opens to request your permission to continue. If this dialog box opens while you are performing the procedures in this guide, and if the dialog box was opened in response to your actions, click **Continue**.

Install and Configure Content Servers

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

When you deploy BranchCache in distributed cache mode or hosted cache mode, you must deploy one or more content servers at your main office or in the cloud. Content servers that are Web servers or application servers use the BranchCache feature. Content servers that are file servers use the BranchCache for network files role service of the File Services server role in Windows Server 2016.

See the following topics to deploy content servers.

- [Install Content Servers that Use the BranchCache Feature](#)
- [Install File Services Content Servers](#)

Install Content Servers that Use the BranchCache Feature

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

To deploy content servers that are Secure Hypertext Transfer Protocol (HTTPS) Web servers, Hypertext Transfer Protocol (HTTP) Web servers, and Background Intelligent Transfer service (BITS)-based application servers, such as Windows Server Update Services (WSUS) and System Center Configuration Manager branch distribution site system servers, you must install the BranchCache feature, start the BranchCache service, and (for WSUS servers only) perform additional configuration steps.

See the following topics to deploy content servers.

- [Install the BranchCache Feature](#)
- [Configure Windows Server Update Services \(WSUS\) Content Servers](#)

Install the BranchCache Feature

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to install the BranchCache feature and start the BranchCache service on a computer running Windows Server® 2016, Windows Server 2012 R2, or Windows Server 2012.

Membership in **Administrators** or equivalent is the minimum required to perform this procedure.

Before you perform this procedure, it is recommended that you install and configure your BITS-based application or Web server.

NOTE

To perform this procedure by using Windows PowerShell, run Windows PowerShell as an Administrator, type the following commands at the Windows PowerShell prompt, and then press ENTER.

`Install-WindowsFeature BranchCache`

`Restart-Computer`

To install and enable the BranchCache feature

1. In Server Manager, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features wizard opens. Click **Next**.
2. In **Select installation type**, ensure that **Role-based or feature-based installation** is selected, and then click **Next**.
3. In **Select destination server**, ensure that the correct server is selected, and then click **Next**.
4. In **Select server roles**, click **Next**.
5. In **Select features**, click **BranchCache**, and then click **Next**.
6. In **Confirm installation selections**, click **Install**. In **Installation progress**, the BranchCache feature installation proceeds. When installation is complete, click **Close**.

After you install the BranchCache feature, the BranchCache service - also called the PeerDistSvc - is enabled, and the start type is Automatic.

Configure Windows Server Update Services (WSUS) Content Servers

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

After installing the BranchCache feature and starting the BranchCache service, WSUS servers must be configured to store update files on the local computer.

When you configure WSUS servers to store update files on the local computer, both the update metadata and the update files are downloaded by and stored directly upon the WSUS server. This ensures that BranchCache client computers receive Microsoft product update files from the WSUS server rather than directly from the Microsoft Update Web site.

For more information about WSUS synchronization, see [Setting up Update Synchronizations](#)

Install File Services Content Servers

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

To deploy content servers that are running the File Services server role, you must install the BranchCache for network files role service of the File Services server role. In addition, you must enable BranchCache on file shares according to your requirements.

During the configuration of the content server, you can allow BranchCache publication of content for all file shares or you can select a subset of file shares to publish.

NOTE

When you deploy a BranchCache enabled file server or Web server as a content server, content information is now calculated offline, well before a BranchCache client requests a file. Because of this improvement, you do not need to configure hash publication for content servers, as you did in the previous version of BranchCache.

This automatic hash generation provides faster performance and more bandwidth savings, because content information is ready for the first client that requests the content, and calculations have already been performed.

See the following topics to deploy content servers.

- [Configure the File Services server role](#)
- [Enable Hash Publication for File Servers](#)
- [Enable BranchCache on a File Share \(Optional\)](#)

Configure the File Services server role

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can deploy BranchCache file server-based content servers on computers running Windows Server 2016 and the File Services server role with the **BranchCache for network files** role service installed.

- To install a BranchCache content server on a computer that does not already have File Services installed, see [Install a New File Server as a Content Server](#).
- To install a BranchCache content server on a computer that is already configured with the File Services server role, see [Configure an Existing File Server as a Content Server](#).

Install a New File Server as a Content Server

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to install the File Services server role and the **BranchCache for Network Files** role service on a computer running Windows Server 2016.

Membership in **Administrators**, or equivalent is the minimum required to perform this procedure.

NOTE

To perform this procedure by using Windows PowerShell, run Windows PowerShell as an Administrator, type the following commands at the Windows PowerShell prompt, and then press ENTER.

```
Install-WindowsFeature FS-BranchCache -IncludeManagementTools
```

```
Restart-Computer
```

To install the Data Deduplication role service, type the following command, and then press ENTER.

```
Install-WindowsFeature FS-Data-Deduplication -IncludeManagementTools
```

To install File Services and the BranchCache for network files role service

1. In Server Manager, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features Wizard opens. In **Before you begin**, click **Next**.
2. In **Select installation type**, ensure that **Role-based or feature-based installation** is selected, and then click **Next**.
3. In **Select destination server**, ensure that the correct server is selected, and then click **Next**.
4. In **Select server roles**, in **Roles**, note that the **File And Storage Services** role is already installed; click the arrow to the left of the role name to expand the selection of role services, and then click the arrow to the left of **File and iSCSI Services**.
5. Select the check boxes for **File Server** and **BranchCache for Network Files**.

TIP

It is recommended that you also select the check box for **Data Deduplication**.

Click **Next**.

6. In **Select features**, click **Next**.
7. In **Confirm installation selections**, review your selections, and then click **Install**. The **Installation progress** pane is displayed during installation. When installation is complete, click **Close**.

Configure an Existing File Server as a Content Server

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to install the **BranchCache for Network Files** role service of the File Services server role on a computer running Windows Server 2016.

IMPORTANT

If the File Services server role is not already installed, do not follow this procedure. Instead, see [Install a New File Server as a Content Server](#).

Membership in **Administrators**, or equivalent is the minimum required to perform this procedure.

NOTE

To perform this procedure by using Windows PowerShell, run Windows PowerShell as an Administrator, type the following commands at the Windows PowerShell prompt, and then press ENTER.

```
Install-WindowsFeature FS-BranchCache -IncludeManagementTools
```

To install the Data Deduplication role service, type the following command, and then press ENTER.

```
Install-WindowsFeature FS-Data-Deduplication -IncludeManagementTools
```

To install the BranchCache for Network Files role service

1. In Server Manager, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features wizard opens. Click **Next**.
2. In **Select installation type**, ensure that **Role-based or feature-based installation** is selected, and then click **Next**.
3. In **Select destination server**, ensure that the correct server is selected, and then click **Next**.
4. In **Select server roles**, in **Roles**, note that the **File And Storage Services** role is already installed; click the arrow to the left of the role name to expand the selection of role services, and then click the arrow to the left of **File and iSCSI Services**.
5. Select the check box for **BranchCache for Network Files**.

TIP

If you have not already done so, it is recommended that you also select the check box for **Data Deduplication**.

Click **Next**.

6. In **Select features**, click **Next**.
7. In **Confirm installation selections**, review your selections, and then click **Install**. The **Installation progress** pane is displayed during installation. When installation is complete, click **Close**.

Enable Hash Publication for File Servers

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can enable BranchCache hash publication on one file server or on multiple file servers.

- To enable hash publication on one file server using local computer Group Policy, see [Enable Hash Publication for Non-Domain Member File Servers](#).
- To enable hash publication on multiple file servers using domain Group Policy, see [Enable Hash Publication for Domain Member File Servers](#).

NOTE

If you have multiple file servers and you want to enable hash publication per share, rather than enabling hash publication for all shares, you can use the instructions in the topic [Enable Hash Publication for Non-Domain Member File Servers](#).

Enable Hash Publication for Non-Domain Member File Servers

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to configure hash publication for BranchCache using local computer Group Policy on a file server that is running Windows Server 2016 with the **BranchCache for Network Files** role service of the File Services server role installed.

This procedure is intended for use on a non-domain member file server. If you perform this procedure on a domain member file server and you also configure BranchCache using domain Group Policy, domain Group Policy settings override local Group Policy settings.

Membership in **Administrators**, or equivalent is the minimum required to perform this procedure.

NOTE

If you have one or more domain member file servers, you can add them to an organizational unit (OU) in Active Directory Domain Services and then use Group Policy to configure hash publication for all of the file servers at one time, rather than individually configuring each file server. For more information, see [Enable Hash Publication for Domain Member File Servers](#).

To enable hash publication for one file server

1. Open Windows PowerShell, type **mmc**, and then press ENTER. The Microsoft Management Console (MMC) opens.
2. In the MMC, on the **File** menu, click **Add/Remove Snap-in**. The **Add or Remove Snap-ins** dialog box opens.
3. In **Add or Remove Snap-ins**, in **Available snap-ins**, double-click **Group Policy Object Editor**. The Group Policy Wizard opens with the Local Computer object selected. Click **Finish**, and then click **OK**.
4. In the Local Group Policy Editor MMC, expand the following path: **Local Computer Policy, Computer Configuration, Administrative Templates, Network, Lanman Server**. Click **Lanman Server**.
5. In the details pane, double-click **Hash Publication for BranchCache**. The **Hash Publication for BranchCache** dialog box opens.
6. In the **Hash Publication for BranchCache** dialog box, click **Enabled**.
7. In **Options**, click **Allow hash publication for all shared folders**, and then click one of the following:
 - a. To enable hash publication for all shared folders on this computer, click **Allow hash publication for all shared folders**.
 - b. To enable hash publication only for shared folders for which BranchCache is enabled, click **Allow hash publication only for shared folders on which BranchCache is enabled**.
 - c. To disallow hash publication for all shared folders on the computer even if BranchCache is enabled on the file shares, click **Disallow hash publication on all shared folders**.
8. Click **OK**.

Enable Hash Publication for Domain Member File Servers

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

When you're using Active Directory Domain Services (AD DS), you can use domain Group Policy to enable BranchCache hash publication for multiple file servers. To do so, you must create an organizational unit (OU), add file servers to the OU, create a BranchCache hash publication Group Policy Object (GPO), and then configure the GPO.

See the following topics to enable hash publication for multiple file servers.

- [Create the BranchCache File Servers Organizational Unit](#)
- [Move File Servers to the BranchCache File Servers Organizational Unit](#)
- [Create the BranchCache Hash Publication Group Policy Object](#)
- [Configure the BranchCache Hash Publication Group Policy Object](#)

Create the BranchCache File Servers Organizational Unit

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to create an organizational unit (OU) in Active Directory Domain Services (AD DS) for BranchCache file servers.

Membership in **Domain Admins**, or equivalent is the minimum required to perform this procedure.

To create the BranchCache file servers organizational unit

1. On a computer where AD DS is installed, in Server Manager, click **Tools**, and then click **Active Directory Users and Computers**. The Active Directory Users and Computers console opens.
2. In the Active Directory Users and Computers console, right-click the domain to which you want to add an OU. For example, if your domain is named example.com, right click **example.com**. Point to **New**, and then click **Organizational Unit**. The **New Object - Organizational Unit** dialog box opens.
3. In the **New Object - Organizational Unit** dialog box, in **Name**, type a name for the new OU. For example, if you want to name the OU BranchCache file servers, type **BranchCache file servers**, and then click **OK**.

Move File Servers to the BranchCache File Servers Organizational Unit

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to add BranchCache file servers to an organizational unit (OU) in Active Directory Domain Services (AD DS).

Membership in **Domain Admins**, or equivalent is the minimum required to perform this procedure.

NOTE

You must create a BranchCache file servers OU in the Active Directory Users and Computers console before you add computer accounts to the OU with this procedure. For more information, see [Create the BranchCache File Servers Organizational Unit](#).

To move file servers to the BranchCache file servers organizational unit

1. On a computer where AD DS is installed, in Server Manager, click **Tools**, and then click **Active Directory Users and Computers**. The Active Directory Users and Computers console opens.
2. In the Active Directory Users and Computers console, locate the computer account for a BranchCache file server, left-click to select the account, and then drag and drop the computer account on the BranchCache file servers OU that you previously created. For example, if you previously created an OU named **BranchCache file servers**, drag and drop the computer account on the **BranchCache file servers** OU.
3. Repeat the previous step for each BranchCache file server in the domain that you want to move to the OU.

Create the BranchCache Hash Publication Group Policy Object

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to create the BranchCache hash publication Group Policy Object (GPO).

Membership in **Domain Admins**, or equivalent is the minimum required to perform this procedure.

NOTE

Before performing this procedure, you must create the BranchCache file servers organizational unit and move file servers into the OU. For more information, see [Enable Hash Publication for Domain Member File Servers](#).

To create the BranchCache hash publication Group Policy Object

1. Open Windows PowerShell, type **mmc**, and then press ENTER. The Microsoft Management Console (MMC) opens.
2. In the MMC, on the **File** menu, click **Add/Remove Snap-in**. The **Add or Remove Snap-ins** dialog box opens.
3. In **Add or Remove Snap-ins**, in **Available snap-ins**, double-click **Group Policy Management**, and then click **OK**.
4. In the Group Policy Management MMC, expand the path to the BranchCache file servers OU that you previously created. For example, if your forest is named example.com, your domain is named example1.com, and your OU is named BranchCache file servers, expand the following path: **Group Policy Management, Forest: example.com, Domains, example1.com, BranchCache file servers**.
5. Right-click **BranchCache file servers**, and then click **Create a GPO in this domain, and Link it here**. The **New GPO** dialog box opens. In **Name**, type a name for the new GPO. For example, if you want to name the object BranchCache Hash Publication, type **BranchCache Hash Publication**. Click **OK**.

Configure the BranchCache Hash Publication Group Policy Object

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to configure the BranchCache hash publication Group Policy Object (GPO) so that all file servers that you added to your OU have the same hash publication policy setting applied to them.

Membership in **Domain Admins**, or equivalent is the minimum required to perform this procedure.

NOTE

Before performing this procedure, you must create the BranchCache file servers organizational unit, move file servers into the OU, and create the BranchCache hash publication GPO. For more information, see [Enable Hash Publication for Domain Member File Servers](#).

To configure the BranchCache hash publication Group Policy Object

1. Run Windows PowerShell as an Administrator, type **mmc**, and then press ENTER. The Microsoft Management Console (MMC) opens.
2. In the MMC, on the **File** menu, click **Add/Remove Snap-in**. The **Add or Remove Snap-ins** dialog box opens.
3. In **Add or Remove Snap-ins**, in **Available snap-ins**, double-click **Group Policy Management**, and then click **OK**.
4. In the Group Policy Management MMC, expand the path to the BranchCache hash publication GPO that you previously created. For example, if your forest is named example.com, your domain is named example1.com, and your GPO is named **BranchCache Hash Publication**, expand the following path: **Group Policy Management, Forest: example.com, Domains, example1.com, Group Policy Objects, BranchCache Hash Publication**.
5. Right-click the **BranchCache Hash Publication** GPO and click **Edit**. The Group Policy Management Editor console opens.
6. In the Group Policy Management Editor console, expand the following path: **Computer Configuration, Policies, Administrative Templates, Network, Lanman Server**.
7. In the Group Policy Management Editor console, click **Lanman Server**. In the details pane, double-click **Hash Publication for BranchCache**. The **Hash Publication for BranchCache** dialog box opens.
8. In the **Hash Publication for BranchCache** dialog box, click **Enabled**.
9. In **Options**, click **Allow hash publication for all shared folders**, and then click one of the following:
 - a. To enable hash publication for all shared folders for all file servers that you added to the OU, click **Allow hash publication for all shared folders**.
 - b. To enable hash publication only for shared folders for which BranchCache is enabled, click **Allow hash publication only for shared folders on which BranchCache is enabled**.

- c. To disallow hash publication for all shared folders on the computer even if BranchCache is enabled on the file shares, click **Disallow hash publication on all shared folders**.

10. Click **OK**.

NOTE

In most cases, you must save the MMC console and refresh the view to display the configuration changes you have made.

Enable BranchCache on a File Share (Optional)

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to enable BranchCache on a file share.

IMPORTANT

You do not need to perform this procedure if you configure the hash publication setting with the value **Allow hash publication for all shared folders**.

Membership in **Administrators**, or equivalent is the minimum required to perform this procedure.

To enable BranchCache on a file share

1. Open Windows PowerShell, type **mmc**, and then press ENTER. The Microsoft Management Console (MMC) opens.
2. In the MMC, on the **File** menu, click **Add/Remove Snap-in**. The **Add or Remove Snap-ins** dialog box opens.
3. In **Add or Remove Snap-ins**, in **Available snap-ins**, double-click **Shared Folders**. The Shared Folders Wizard opens with the Local Computer object selected. Configure the View that you prefer, click **Finish**, and then click **OK**.
4. Double-click **Shared Folders (Local)**, and then click **Shares**.
5. In the details pane, right-click a share, and then click **Properties**. The share's **Properties** dialog box opens.
6. In the **Properties** dialog box, on the **General** tab, click **Offline Settings**. The **Offline Settings** dialog box opens.
7. Ensure that **Only the files and programs that users specify are available offline** is selected, and then click **Enable BranchCache**.
8. Click **OK** twice.

Deploy Hosted Cache Servers (Optional)

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to install and configure BranchCache hosted cache servers that are located in branch offices where you want to deploy BranchCache hosted cache mode. With BranchCache in Windows Server 2016, you can deploy multiple hosted cache servers in one branch office.

IMPORTANT

This step is optional because distributed cache mode does not require a hosted cache server computer in branch offices. If you are not planning on deploying hosted cache mode in any branch offices, you do not need to deploy a hosted cache server, and you do not need to perform the steps in this procedure.

You must be a member of **Administrators**, or equivalent to perform this procedure.

To install and configure a hosted cache server

1. On the computer that you want to configure as a hosted cache server, run the following command at a Windows PowerShell prompt to install the BranchCache feature.

```
Install-WindowsFeature BranchCache -IncludeManagementTools
```

2. Configure the computer as a hosted cache server by using one of the following commands:

- To configure a non-domain joined computer as a hosted cache server, type the following command at the Windows PowerShell prompt, and then press ENTER.

```
Enable-BCHostedServer
```

- To configure a domain joined computer as a hosted cache server, and to register a service connection point in Active Directory for automatic hosted cache server discovery by client computers, type the following command at the Windows PowerShell prompt, and then press ENTER.

```
Enable-BCHostedServer -RegisterSCP
```

3. To verify the correct configuration of the hosted cache server, type the following command at the Windows PowerShell prompt, and then press ENTER.

```
Get-BCStatus
```

NOTE

After you run this command, in the section **HostedCacheServerConfiguration**, the value for **HostedCacheServerEnabled** is **True**. If you configured a domain joined hosted cache server to register a service connection point (SCP) in Active Directory, the value for **HostedCacheScpRegistrationEnabled** is **True**.

Prehashing and Preloading Content on Hosted Cache Servers (Optional)

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to force the creation of content information - also called hashes - on BranchCache-enabled Web and file servers. You can also gather the data on file and web servers into packages that can be transferred to remote hosted cache servers. This provides you with the ability to preload content on remote hosted cache servers so that data is available for the first client access.

You must be a member of **Administrators**, or equivalent to perform this procedure.

To prehash content and preload the content on hosted cache servers

1. Log on to the file or Web server that contains the data that you wish to preload, and identify the folders and files that you wish to load on one or more remote hosted cache servers.
2. Run Windows PowerShell as an Administrator. For each folder and file, run either the `Publish-BCFileContent` command or the `Publish-BCWebContent` command, depending on the type of content server, to trigger hash generation and to add data to a data package.
3. After all the data has been added to the data package, export it by using the `Export-BCCachePackage` command to produce a data package file.
4. Move the data package file to the remote hosted cache servers by using your choice of file transfer technology. FTP, SMB, HTTP, DVD and portable hard disks are all viable transports.
5. Import the data package file on the remote hosted cache servers by using the `Import-BCCachePackage` command.

Configure BranchCache Client Computers

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the following topics to configure domain member and non-domain member client computers as BranchCache distributed cache or hosted cache mode clients.

- [Use Group Policy to Configure Domain Member Client Computers](#)
- [Use Windows PowerShell to Configure Non-Domain Member Client Computers](#)
- [Configure Firewall Rules for Non-Domain Members to Allow BranchCache Traffic](#)
- [Verify Client Computer Settings](#)

Use Group Policy to Configure Domain Member Client Computers

9/21/2018 • 4 minutes to read • [Edit Online](#)

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

In this section, you create a Group Policy Object for all of the computers in your organization, configure domain member client computers with distributed cache mode or hosted cache mode, and configure Windows Firewall with Advanced Security to allow BranchCache traffic.

This section contains the following procedures.

1. [To create a Group Policy Object and configure BranchCache modes](#)
2. [To configure Windows Firewall with Advanced Security Inbound Traffic Rules](#)
3. [To configure Windows Firewall with Advanced Security Outbound Traffic Rules](#)

TIP

In the following procedure, you are instructed to create a Group Policy Object in the Default Domain Policy, however, you can create the object in an organizational unit (OU) or other container that is appropriate for your deployment.

You must be a member of **Domain Admins**, or equivalent to perform these procedures.

To create a Group Policy Object and configure BranchCache modes

1. On a computer upon which the Active Directory Domain Services server role is installed, in Server Manager, click **Tools**, and then click **Group Policy Management**. The Group Policy Management console opens.
2. In the Group Policy Management console, expand the following path: **Forest: example.com, Domains, example.com, Group Policy Objects**, where *example.com* is the name of the domain where the BranchCache client computer accounts that you want to configure are located.
3. Right-click **Group Policy Objects**, and then click **New**. The **New GPO** dialog box opens. In **Name**, type a name for the new Group Policy Object (GPO). For example, if you want to name the object BranchCache Client Computers, type **BranchCache Client Computers**. Click **OK**.
4. In the Group Policy Management console, ensure that **Group Policy Objects** is selected, and in the details pane right-click the GPO that you just created. For example, if you named your GPO BranchCache Client Computers, right-click **BranchCache Client Computers**. Click **Edit**. The Group Policy Management Editor console opens.
5. In the Group Policy Management Editor console, expand the following path: **Computer Configuration, Policies, Administrative Templates: Policy definitions (ADMX files) retrieved from the local computer, Network, BranchCache**.
6. Click **BranchCache**, and then in the details pane, double-click **Turn on BranchCache**. The policy setting dialog box opens.
7. In the **Turn on BranchCache** dialog box, click **Enabled**, and then click **OK**.

8. To enable BranchCache distributed cache mode, in the details pane, double-click **Set BranchCache Distributed Cache mode**. The policy setting dialog box opens.
9. In the **Set BranchCache Distributed Cache mode** dialog box, click **Enabled**, and then click **OK**.
10. If you have one or more branch offices where you are deploying BranchCache in hosted cache mode, and you have deployed hosted cache servers in those offices, double-click **Enable Automatic Hosted Cache Discovery by Service Connection Point**. The policy setting dialog box opens.
11. In the **Enable Automatic Hosted Cache Discovery by Service Connection Point** dialog box, click **Enabled**, and then click **OK**.

NOTE

When you enable both the **Set BranchCache Distributed Cache mode** and the **Enable Automatic Hosted Cache Discovery by Service Connection Point** policy settings, client computers operate in BranchCache distributed cache mode unless they find a hosted cache server in the branch office, at which point they operate in hosted cache mode.

12. Use the procedures below to configure firewall settings on client computers by using Group Policy.

To configure Windows Firewall with Advanced Security Inbound Traffic Rules

1. In the Group Policy Management console, expand the following path: **Forest: example.com, Domains, example.com, Group Policy Objects**, where *example.com* is the name of the domain where the BranchCache client computer accounts that you want to configure are located.
2. In the Group Policy Management console, ensure that **Group Policy Objects** is selected, and in the details pane right-click the BranchCache client computers GPO that you created previously. For example, if you named your GPO BranchCache Client Computers, right-click **BranchCache Client Computers**. Click **Edit**. The Group Policy Management Editor console opens.
3. In the Group Policy Management Editor console, expand the following path: **Computer Configuration, Policies, Windows Settings, Security Settings, Windows Firewall with Advanced Security, Windows Firewall with Advanced Security - LDAP, Inbound Rules**.
4. Right-click **Inbound Rules**, and then click **New Rule**. The New Inbound Rule Wizard opens.
5. In **Rule Type**, click **Predefined**, expand the list of choices, and then click **BranchCache - Content Retrieval (Uses HTTP)**. Click **Next**.
6. In **Predefined Rules**, click **Next**.
7. In **Action**, ensure that **Allow the connection** is selected, and then click **Finish**.

IMPORTANT

You must select **Allow the connection** for the BranchCache client to be able to receive traffic on this port.

8. To create the WS-Discovery firewall exception, again right-click **Inbound Rules**, and then click **New Rule**. The New Inbound Rule Wizard opens.
9. In **Rule Type**, click **Predefined**, expand the list of choices, and then click **BranchCache - Peer Discovery (Uses WSD)**. Click **Next**.
10. In **Predefined Rules**, click **Next**.

11. In **Action**, ensure that **Allow the connection** is selected, and then click **Finish**.

IMPORTANT

You must select **Allow the connection** for the BranchCache client to be able to receive traffic on this port.

To configure Windows Firewall with Advanced Security Outbound Traffic Rules

1. In the Group Policy Management Editor console, right-click **Outbound Rules**, and then click **New Rule**.
The New Outbound Rule Wizard opens.
2. In **Rule Type**, click **Predefined**, expand the list of choices, and then click **BranchCache - Content Retrieval (Uses HTTP)**. Click **Next**.
3. In **Predefined Rules**, click **Next**.
4. In **Action**, ensure that **Allow the connection** is selected, and then click **Finish**.

IMPORTANT

You must select **Allow the connection** for the BranchCache client to be able to send traffic on this port.

5. To create the WS-Discovery firewall exception, again right-click **Outbound Rules**, and then click **New Rule**.
The New Outbound Rule Wizard opens.
6. In **Rule Type**, click **Predefined**, expand the list of choices, and then click **BranchCache - Peer Discovery (Uses WSD)**. Click **Next**.
7. In **Predefined Rules**, click **Next**.
8. In **Action**, ensure that **Allow the connection** is selected, and then click **Finish**.

IMPORTANT

You must select **Allow the connection** for the BranchCache client to be able to send traffic on this port.

Use Windows PowerShell to Configure Non-Domain Member Client Computers

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to manually configure a BranchCache client computer for distributed cache mode or hosted cache mode.

NOTE

If you have configured BranchCache client computers using Group Policy, the Group Policy settings override any manual configuration of client computers to which the policies are applied.

Membership in **Administrators**, or equivalent is the minimum required to perform this procedure.

To enable BranchCache distributed or hosted cache mode

1. On the BranchCache client computer that you want to configure, run Windows PowerShell as an Administrator, and then do one of the following.

- To configure the client computer for BranchCache distributed cache mode, type the following command, and then press ENTER.

```
Enable-BCDistributed
```

- To configure the client computer for BranchCache hosted cache mode, type the following command, and then press ENTER.

```
Enable-BCHostedClient
```

TIP

If you want to specify the available hosted cache servers, use the `-ServerNames` parameter with a comma separated list of your hosted cache servers as the parameter value. For example, if you have two hosted cache servers named HCS1 and HCS2, configure the client computer for hosted cache mode with the following command.

```
Enable-BCHostedClient -ServerNames HCS1,HCS2
```

Configure Firewall Rules for Non-Domain Members to Allow BranchCache Traffic

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the information in this topic to configure third party firewall products and to manually configure a client computer with firewall rules that allow BranchCache to run in distributed cache mode.

NOTE

- If you have configured BranchCache client computers using Group Policy, the Group Policy settings override any manual configuration of client computers to which the policies are applied.
- If you have deployed BranchCache with DirectAccess, you can use the settings in this topic to configure IPsec rules to allow BranchCache traffic.

Membership in **Administrators**, or equivalent is the minimum required to make these configuration changes.

[MS-PCCRD]: Peer Content Caching and Retrieval Discovery Protocol

Distributed cache clients must allow inbound and outbound MS-PCCRD traffic, which is carried in the Web Services Dynamic Discovery (WS-Discovery) protocol.

Firewall settings must allow multicast traffic in addition to inbound and outbound traffic. You can use the following settings to configure firewall exceptions for distributed cache mode.

IPv4 multicast: 239.255.255.250

IPv6 multicast: FF02::C

Inbound traffic: Local port: 3702, Remote port: ephemeral

Outbound traffic: Local port: ephemeral, Remote port: 3702

Program: %systemroot%\system32\svchost.exe (BranchCache Service [PeerDistSvc])

[MS-PCCRR]: Peer Content Caching and Retrieval: Retrieval Protocol

Distributed cache clients must allow inbound and outbound MS-PCCRR traffic, which is carried in the HTTP 1.1 protocol as documented in request for comments (RFC) 2616.

Firewall settings must allow inbound and outbound traffic. You can use the following settings to configure firewall exceptions for distributed cache mode.

Inbound traffic: Local port: 80, Remote port: ephemeral

Outbound traffic: Local port: ephemeral, Remote port: 80

Verify Client Computer Settings

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to verify that the client computer is correctly configured for BranchCache.

NOTE

This procedure includes steps for manually updating Group Policy and for restarting the BranchCache service. You do not need to perform these actions if you reboot the computer, as they will occur automatically in this circumstance.

You must be a member of **Administrators**, or equivalent to perform this procedure.

To verify BranchCache client computer settings

1. To refresh Group Policy on the client computer whose BranchCache configuration you want to verify, run Windows PowerShell as an Administrator, type the following command, and then press ENTER.

```
gpupdate /force
```

2. For client computers that are configured in hosted cache mode and are configured to automatically discover hosted cache servers by service connection point, run the following commands to stop and restart the BranchCache service.

```
net stop peerdistsvc
```

```
net start peerdistsvc
```

3. Inspect the current BranchCache operational mode by running the following command.

```
Get-BCStatus
```

4. In Windows PowerShell, review the output of the **Get-BCStatus** command.

The value for **BranchCacheIsEnabled** should be **True**.

In **ClientSettings**, the value for **CurrentClientMode** should be **DistributedClient** or **HostedCacheClient**, depending on the mode that you configured using this guide.

In **ClientSettings**, if you configured hosted cache mode and provided the names of your hosted cache servers during configuration, or if the client has automatically located hosted cache servers using service connection points, **HostedCacheServerList** should have a value that is the same as the name or names of your hosted cache servers. For example, if your hosted cache server is named HCS1 and your domain is corp.contoso.com, the value for **HostedCacheServerList** is **HCS1.corp.contoso.com**.

5. If any of the BranchCache settings listed above do not have the correct values, use the steps in this guide to verify the Group Policy or Local Computer Policy settings, as well as the firewall exceptions, that you configured, and ensure that they are correct. In addition, either restart the computer or follow the steps in this procedure to refresh Group Policy and restart the BranchCache service.

DirectAccess

3/23/2018 • 2 minutes to read • [Edit Online](#)

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

In Windows Server 2016, **DirectAccess and VPN** is a role service of the **Remote Access** server role.

DirectAccess allows connectivity for remote users to organization network resources without the need for traditional Virtual Private Network (VPN) connections.

DirectAccess documentation is now located in the [Remote access and server management](#) section of the Windows Server 2016 table of contents, under [Remote Access](#). For more information, see [DirectAccess](#).

Domain Name System (DNS)

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

Domain Name System (DNS) is one of the industry-standard suite of protocols that comprise TCP/IP, and together the DNS Client and DNS Server provide computer name-to-IP address mapping name resolution services to computers and users.

NOTE

In addition to this topic, the following DNS content is available.

- [What's New in DNS Client](#)
- [What's New in DNS Server](#)
- [DNS Policy Scenario Guide](#)
- Video: [Windows Server 2016: DNS management in IPAM](#)

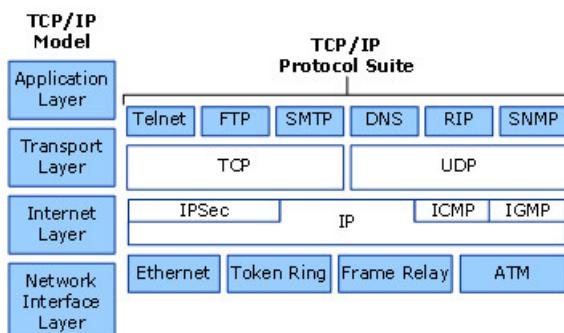
In Windows Server 2016, DNS is a server role that you can install by using Server Manager or Windows PowerShell commands. If you are installing a new Active Directory forest and domain, DNS is automatically installed with Active Directory as the Global Catalogue server for the forest and domain.

Active Directory Domain Services (AD DS) uses DNS as its domain controller location mechanism. When any of the principal Active Directory operations is performed, such as authentication, updating, or searching, computers use DNS to locate Active Directory domain controllers. In addition, domain controllers use DNS to locate each other.

The DNS Client service is included in all client and server versions of the Windows operating system, and is running by default upon operating system installation. When you configure a TCP/IP network connection with the IP address of a DNS server, the DNS Client queries the DNS server to discover domain controllers, and to resolve computer names to IP addresses. For example, when a network user with an Active Directory user account logs in to an Active Directory domain, the DNS Client service queries the DNS server to locate a domain controller for the Active Directory domain. When the DNS server responds to the query and provides the domain controller's IP address to the client, the client contacts the domain controller and the authentication process can begin.

The Windows Server 2016 DNS Server and DNS Client services use the DNS protocol that is included in the TCP/IP protocol suite. DNS is part of the application layer of the TCP/IP reference model, as shown in the following illustration.

DNS in TCP/IP



What's New in DNS Client in Windows Server 2016

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

This topic describes the Domain Name System (DNS) client functionality that is new or changed in Windows 10 and Windows Server 2016 and later versions of these operating systems.

Updates to DNS Client

DNS Client service binding: In Windows 10, the DNS Client service offers enhanced support for computers with more than one network interface. For multi-homed computers, DNS resolution is optimized in the following ways:

- When a DNS server that is configured on a specific interface is used to resolve a DNS query, the DNS Client service will bind to this interface before sending the DNS query.

By binding to a specific interface, the DNS client can clearly specify the interface where name resolution occurs, enabling applications to optimize communications with the DNS client over this network interface.

- If the DNS server that is used is designated by a Group Policy setting from the Name Resolution Policy Table (NRPT), the DNS Client service does not bind to a specific interface.

NOTE

Changes to the DNS Client service in Windows 10 are also present in computers running Windows Server 2016 and later versions.

See also

- [What's New in DNS Server in Windows Server 2016](#)

What's New in DNS Server in Windows Server

9/1/2018 • 8 minutes to read • [Edit Online](#)

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

This topic describes the Domain Name System (DNS) server functionality that is new or changed in Windows Server 2016.

In Windows Server 2016, DNS Server offers enhanced support in the following areas.

FUNCTIONALITY	NEW OR IMPROVED	DESCRIPTION
DNS Policies	New	You can configure DNS policies to specify how a DNS server responds to DNS queries. DNS responses can be based on client IP address (location), time of the day, and several other parameters. DNS policies enable location-aware DNS, traffic management, load balancing, split-brain DNS, and other scenarios.
Response Rate Limiting (RRL)	New	You can enable response rate limiting on your DNS servers. By doing this, you avoid the possibility of malicious systems using your DNS servers to initiate a denial of service attack on a DNS client.
DNS-based Authentication of Named Entities (DANE)	New	You can use TLSA (Transport Layer Security Authentication) records to provide information to DNS clients that state what CA they should expect a certificate from for your domain name. This prevents man-in-the-middle attacks where someone might corrupt the DNS cache to point to their own website, and provide a certificate they issued from a different CA.
Unknown record support	New	You can add records which are not explicitly supported by the Windows DNS server using the unknown record functionality.
IPv6 root hints	New	You can use the native IPV6 root hints support to perform internet name resolution using the IPV6 root servers.
Windows PowerShell Support	Improved	New Windows PowerShell cmdlets are available for DNS Server.

DNS Policies

You can use DNS Policy for Geo-Location based traffic management, intelligent DNS responses based on the time of day, to manage a single DNS server configured for split-brain deployment, applying filters on DNS queries, and more. The following items provide more detail about these capabilities.

- **Application Load Balancing.** When you have deployed multiple instances of an application at different locations, you can use DNS policy to balance the traffic load between the different application instances, dynamically allocating the traffic load for the application.
- **Geo-Location Based Traffic Management.** You can use DNS Policy to allow primary and secondary DNS servers to respond to DNS client queries based on the geographical location of both the client and the resource to which the client is attempting to connect, providing the client with the IP address of the closest resource.
- **Split Brain DNS.** With split-brain DNS, DNS records are split into different Zone Scopes on the same DNS server, and DNS clients receive a response based on whether the clients are internal or external clients. You can configure split-brain DNS for Active Directory integrated zones or for zones on standalone DNS servers.
- **Filtering.** You can configure DNS policy to create query filters that are based on criteria that you supply. Query filters in DNS policy allow you to configure the DNS server to respond in a custom manner based on the DNS query and DNS client that sends the DNS query.
- **Forensics.** You can use DNS policy to redirect malicious DNS clients to a non-existent IP address instead of directing them to the computer they are trying to reach.
- **Time of day based redirection.** You can use DNS policy to distribute application traffic across different geographically distributed instances of an application by using DNS policies that are based on the time of day.

You can also use DNS policies for Active Directory integrated DNS zones.

For more information, see the [DNS Policy Scenario Guide](#).

Response Rate Limiting

You can configure RRL settings to control how to respond to requests to a DNS client when your server receives several requests targeting the same client. By doing this, you can prevent someone from sending a Denial of Service (Dos) attack using your DNS servers. For instance, a bot net can send requests to your DNS server using the IP address of a third computer as the requestor. Without RRL, your DNS servers might respond to all the requests, flooding the third computer. When you use RRL, you can configure the following settings:

- **Responses per second.** This is the maximum number of times the same response will be given to a client within one second.
- **Errors per second.** This is the maximum number of times an error response will be sent to the same client within one second.
- **Window.** This is the number of seconds for which responses to a client will be suspended if too many requests are made.
- **Leak rate.** This is how frequently the DNS server will respond to a query during the time responses are suspended. For instance, if the server suspends responses to a client for 10 seconds, and the leak rate is 5, the server will still respond to one query for every 5 queries sent. This allows the legitimate clients to get responses even when the DNS server is applying response rate limiting on their subnet or FQDN.
- **TC rate.** This is used to tell the client to try connecting with TCP when responses to the client are suspended. For instance, if the TC rate is 3, and the server suspends responses to a given client, the server will issue a request for TCP connection for every 3 queries received. Make sure the value for TC rate is lower than the

leak rate, to give the client the option to connect via TCP before leaking responses.

- **Maximum responses.** This is the maximum number of responses the server will issue to a client while responses are suspended.
- **White list domains.** This is a list of domains to be excluded from RRL settings.
- **White list subnets.** This is a list of subnets to be excluded from RRL settings.
- **White list server interfaces.** This is a list of DNS server interfaces to be excluded from RRL settings.

DANE support

You can use DANE support (RFC 6394 and 6698) to specify to your DNS clients what CA they should expect certificates to be issued from for domains names hosted in your DNS server. This prevents a form of man-in-the-middle attack where someone is able to corrupt a DNS cache and point a DNS name to their own IP address.

For instance, imagine you host a secure website that uses SSL at www.contoso.com by using a certificate from a well-known authority named CA1. Someone might still be able to get a certificate for www.contoso.com from a different, not-so-well-known, certificate authority named CA2. Then, the entity hosting the fake www.contoso.com website might be able to corrupt the DNS cache of a client or server to point www.contoso.com to their fake site. The end user will be presented a certificate from CA2, and may simply acknowledge it and connect to the fake site. With DANE, the client would make a request to the DNS server for contoso.com asking for the TLSA record and learn that the certificate for www.contoso.com was issued by CA1. If presented with a certificate from another CA, the connection is aborted.

Unknown record support

An "Unknown Record" is an RR whose RDATA format is not known to the DNS server. The newly added support for unknown record (RFC 3597) types means that you can add the unsupported record types into the Windows DNS server zones in the binary on-wire format. The windows caching resolver already has the ability to process unknown record types. Windows DNS server will not do any record specific processing for the unknown records, but will send it back in responses if queries are received for it.

IPv6 root hints

The IPV6 root hints, as published by IANA, have been added to the windows DNS server. The internet name queries can now use IPv6 root servers for performing name resolutions.

Windows PowerShell support

The following new Windows PowerShell cmdlets and parameters are introduced in Windows Server 2016.

- **Add-DnsServerRecursionScope.** This cmdlet creates a new recursion scope on the DNS server. Recursion scopes are used by DNS policies to specify a list of forwarders to be used in a DNS query.
- **Remove-DnsServerRecursionScope.** This cmdlet removes existing recursion scopes.
- **Set-DnsServerRecursionScope.** This cmdlet changes the settings of an existing recursion scope.
- **Get-DnsServerRecursionScope.** This cmdlet retrieves information about existing recursion scopes.
- **Add-DnsServerClientSubnet.** This cmdlet creates a new DNS client subnet. Subnets are used by DNS policies to identify where a DNS client is located.
- **Remove-DnsServerClientSubnet.** This cmdlet removes existing DNS client subnets.
- **Set-DnsServerClientSubnet.** This cmdlet changes the settings of an existing DNS client subnet.

- **Get-DnsServerClientSubnet**. This cmdlet retrieves information about existing DNS client subnets.
- **Add-DnsServerQueryResolutionPolicy**. This cmdlet creates a new DNS query resolution policy. DNS query resolution policies are used to specify how, or if, a query is responded to, based on different criteria.
- **Remove-DnsServerQueryResolutionPolicy**. This cmdlet removes existing DNS policies.
- **Set-DnsServerQueryResolutionPolicy**. This cmdlet changes the settings of an existing DNS policy.
- **Get-DnsServerQueryResolutionPolicy**. This cmdlet retrieves information about existing DNS policies.
- **Enable-DnsServerPolicy**. This cmdlet enables existing DNS policies.
- **Disable-DnsServerPolicy**. This cmdlet disables existing DNS policies.
- **Add-DnsServerZoneTransferPolicy**. This cmdlet creates a new DNS server zone transfer policy. DNS zone transfer policies specify whether to deny or ignore a zone transfer based on different criteria.
- **Remove-DnsServerZoneTransferPolicy**. This cmdlet removes existing DNS server zone transfer policies.
- **Set-DnsServerZoneTransferPolicy**. This cmdlet changes settings of an existing DNS server zone transfer policy.
- **Get-DnsServerResponseRateLimiting**. This cmdlet retrieves RRL settings.
- **Set-DnsServerResponseRateLimiting**. This cmdlet changes RRL settings.
- **Add-DnsServerResponseRateLimitingExceptionlist**. This cmdlet creates an RRL exception list on the DNS server.
- **Get-DnsServerResponseRateLimitingExceptionlist**. This cmdlet retrieves RRL exception lists.
- **Remove-DnsServerResponseRateLimitingExceptionlist**. This cmdlet removes an existing RRL exception list.
- **Set-DnsServerResponseRateLimitingExceptionlist**. This cmdlet changes RRL exception lists.
- **Add-DnsServerResourceRecord**. This cmdlet was updated to support unknown record type.
- **Get-DnsServerResourceRecord**. This cmdlet was updated to support unknown record type.
- **Remove-DnsServerResourceRecord**. This cmdlet was updated to support unknown record type.
- **Set-DnsServerResourceRecord**. This cmdlet was updated to support unknown record type

For more information, see the following Windows Server 2016 Windows PowerShell command reference topics.

- [DnsServer Module](#)
- [DnsClient Module](#)

See also

- [What's New in DNS Client](#)

DNS Policy Scenario Guide

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

This guide is intended for use by DNS, network, and systems administrators.

DNS Policy is a new feature for DNS in Windows Server® 2016. You can use this guide to learn how to use DNS policy to control how a DNS server processes name resolution queries based on different parameters that you define in policies.

This guide contains DNS policy overview information, as well as specific DNS policy scenarios that provide you with instructions on how to configure DNS server behavior to accomplish your goals, including geo-location based traffic management for primary and secondary DNS servers, application high availability, split-brain DNS, and more.

This guide contains the following sections.

- [DNS Policies Overview](#)
- [Use DNS Policy for Geo-Location Based Traffic Management with Primary Servers](#)
- [Use DNS Policy for Geo-Location Based Traffic Management with Primary-Secondary Deployments](#)
- [Use DNS Policy for Intelligent DNS Responses Based on the Time of Day](#)
- [DNS Responses Based on Time of Day with an Azure Cloud App Server](#)
- [Use DNS Policy for Split-Brain DNS Deployment](#)
- [Use DNS Policy for Split-Brain DNS in Active Directory](#)
- [Use DNS Policy for Applying Filters on DNS Queries](#)
- [Use DNS Policy for Application Load Balancing](#)
- [Use DNS Policy for Application Load Balancing With Geo-Location Awareness](#)

DNS Policies Overview

9/1/2018 • 11 minutes to read • [Edit Online](#)

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

You can use this topic to learn about DNS Policy, which is new in Windows Server 2016. You can use DNS Policy for Geo-Location based traffic management, intelligent DNS responses based on the time of day, to manage a single DNS server configured for split-brain deployment, applying filters on DNS queries, and more. The following items provide more detail about these capabilities.

- **Application Load Balancing.** When you have deployed multiple instances of an application at different locations, you can use DNS policy to balance the traffic load between the different application instances, dynamically allocating the traffic load for the application.
- **Geo-Location Based Traffic Management.** You can use DNS Policy to allow primary and secondary DNS servers to respond to DNS client queries based on the geographical location of both the client and the resource to which the client is attempting to connect, providing the client with the IP address of the closest resource.
- **Split Brain DNS.** With split-brain DNS, DNS records are split into different Zone Scopes on the same DNS server, and DNS clients receive a response based on whether the clients are internal or external clients. You can configure split-brain DNS for Active Directory integrated zones or for zones on standalone DNS servers.
- **Filtering.** You can configure DNS policy to create query filters that are based on criteria that you supply. Query filters in DNS policy allow you to configure the DNS server to respond in a custom manner based on the DNS query and DNS client that sends the DNS query.
- **Forensics.** You can use DNS policy to redirect malicious DNS clients to a non-existent IP address instead of directing them to the computer they are trying to reach.
- **Time of day based redirection.** You can use DNS policy to distribute application traffic across different geographically distributed instances of an application by using DNS policies that are based on the time of day.

New Concepts

In order to create policies to support the scenarios listed above, it is necessary to be able to identify groups of records in a zone, groups of clients on a network, among other elements. These elements are represented by the following new DNS objects:

- **Client subnet:** a client subnet object represents an IPv4 or IPv6 subnet from which queries are submitted to a DNS server. You can create subnets to later define policies to be applied based on what subnet the requests come from. For instance, in a split brain DNS scenario, the request for resolution for a name such as *www.microsoft.com* can be answered with an internal IP address to clients from internal subnets, and a different IP address to clients in external subnets.
- **Recursion scope:** recursion scopes are unique instances of a group of settings that control recursion on a DNS server. A recursion scope contains a list of forwarders and specifies whether recursion is enabled. A DNS server can have many recursion scopes. DNS server recursion policies allow you to choose a recursion scope for a set of queries. If the DNS server is not authoritative for certain queries, DNS server recursion policies allow you to control how to resolve those queries. You can specify which forwarders to use and

whether to use recursion.

- **Zone scopes:** a DNS zone can have multiple zone scopes, with each zone scope containing their own set of DNS records. The same record can be present in multiple scopes, with different IP addresses. Also, zone transfers are done at the zone scope level. That means that records from a zone scope in a primary zone will be transferred to the same zone scope in a secondary zone.

Types of Policy

DNS Policies are divided by level and type. You can use Query Resolution Policies to define how queries are processed, and Zone Transfer Policies to define how zone transfers occur. You can apply Each policy type at the server level or the zone level.

Query Resolution Policies

You can use DNS Query Resolution Policies to specify how incoming resolution queries are handled by a DNS server. Every DNS Query Resolution Policy contains the following elements:

FIELD	DESCRIPTION	POSSIBLE VALUES
Name	Policy name	- Up to 256 characters - Can contain any character valid for a file name
State	Policy state	- Enable (default) - Disabled
Level	Policy level	- Server - Zone
Processing order	Once a query is classified by level and applies on, the server finds the first policy for which the query matches the criteria and applies it to query	- Numeric value - Unique value per policy containing the same level and applies on value
Action	Action to be performed by DNS server	- Allow (default for zone level) - Deny (default on server level) - Ignore
Criteria	Policy condition (AND/OR) and list of criterion to be met for policy to be applied	- Condition operator (AND/OR) - List of criteria (see the criterion table below)
Scope	List of zone scopes and weighted values per scope. Weighted values are used for load balancing distribution. For instance, if this list includes datacenter1 with a weight of 3 and datacenter2 with a weight of 5 the server will respond with a record from datacentre1 three times out of eight requests	- List of zone scopes (by name) and weights

NOTE

Server level policies can only have the values **Deny** or **Ignore** as an action.

The DNS policy criteria field is composed of two elements:

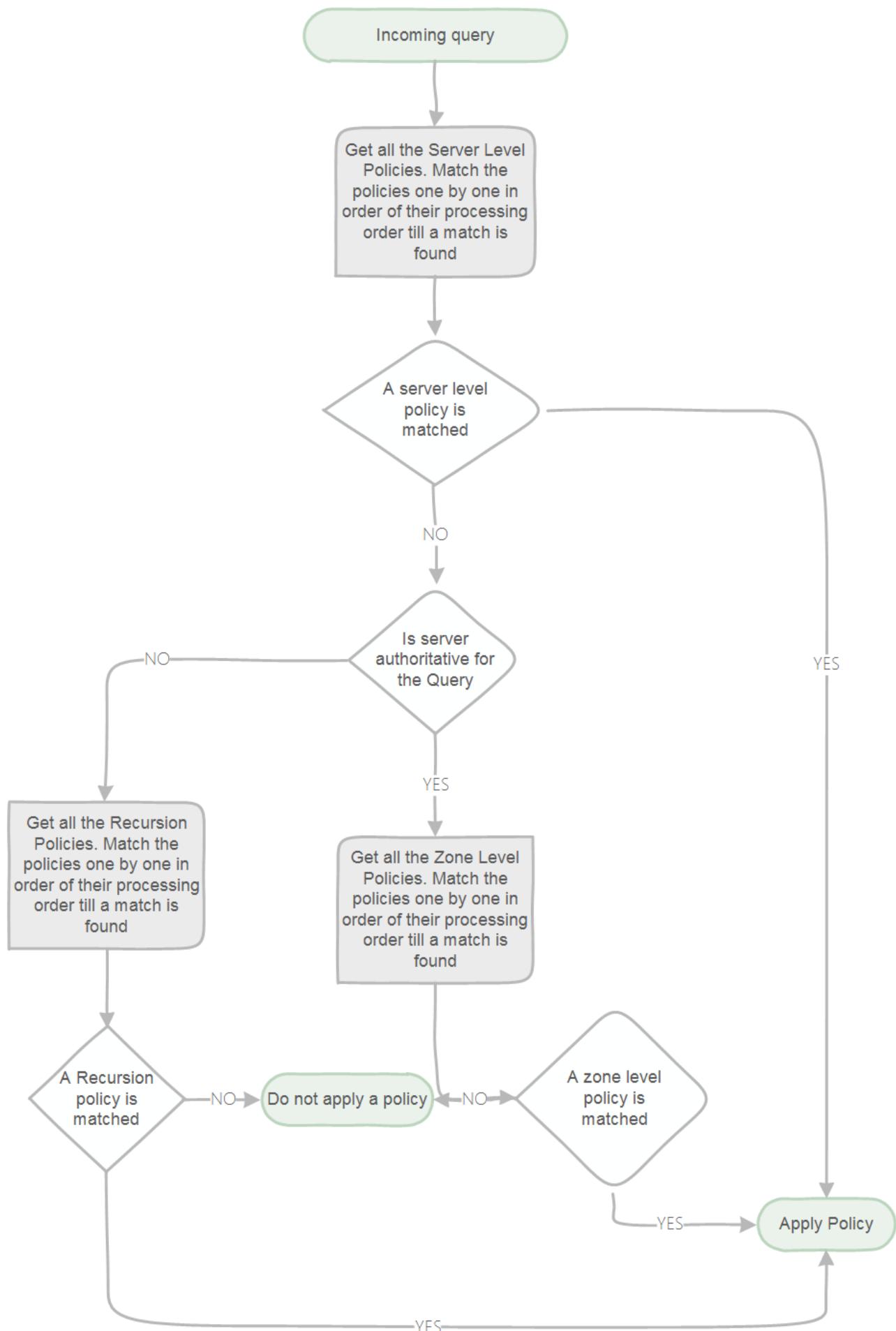
Name	Description	Sample Values
Client Subnet	Transport protocol used in the query. Possible entries are UDP and TCP	- EQ, Spain, France - resolves to true if the subnet is identified as either Spain or France - NE, Canada, Mexico - resolves to true if the client subnet is any subnet other than Canada and Mexico
Transport Protocol	Transport protocol used in the query. Possible entries are UDP and TCP	- EQ, TCP - EQ, UDP
Internet Protocol	Network protocol used in the query. Possible entries are IPv4 and IPv6	- EQ, IPv4 - EQ, IPv6
Server Interface IP address	IP address for the incoming DNS server network interface	- EQ, 10.0.0.1 - EQ, 192.168.1.1
FQDN	FQDN of record in the query, with the possibility of using a wild card	- EQ, www.contoso.com - resolves to true only if the query is trying to resolve the <i>www.contoso.com</i> FQDN - EQ, *.contoso.com, *.woodgrove.com - resolves to true if the query is for any record ending in <i>contoso.com</i> OR <i>woodgrove.com</i>
Query Type	Type of record being queried (A, SRV, TXT)	- EQ, TXT, SRV - resolves to true if the query is requesting a TXT OR SRV record - EQ, MX - resolves to true if the query is requesting an MX record
Time of Day	Time of day the query is received	- EQ, 10:00-12:00, 22:00-23:00 - resolves to true if the query is received between 10 AM and noon, OR between 10PM and 11PM

Using the table above as a starting point, the table below could be used to define a criterion that is used to match queries for any type of records but SRV records in the contoso.com domain coming from a client in the 10.0.0.0/24 subnet via TCP between 8 and 10 PM through interface 10.0.0.3:

Name	Value
Client Subnet	EQ,10.0.0.0/24
Transport Protocol	EQ,TCP
Server Interface IP address	EQ,10.0.0.3
FQDN	EQ,*.contoso.com
Query Type	NE,SRV
Time of Day	EQ,20:00-22:00

You can create multiple query resolution policies of the same level, as long as they have a different value for the

processing order. When multiple policies are available, the DNS server processes incoming queries in the following manner:



Recursion policies are a special **type** of server level policies. Recursion policies control how the DNS server performs recursion for a query. Recursion policies apply only when query processing reaches the recursion path. You can choose a value of DENY or IGNORE for recursion for a set of queries. Alternatively, you can choose a set of forwarders for a set of queries.

You can use recursion policies to implement a Split-brain DNS configuration. In this configuration, the DNS server performs recursion for a set of clients for a query, while the DNS server does not perform recursion for other clients for that query.

Recursion policies contains the same elements a regular DNS query resolution policy contains, along with the elements in the table below:

NAME	DESCRIPTION
Apply on recursion	Specifies that this policy should only be used for recursion.
Recursion Scope	Name of the recursion scope.

NOTE

Recursion policies can only be created at the server level.

Zone Transfer Policies

Zone transfer policies control whether a zone transfer is allowed or not by your DNS server. You can create policies for zone transfer at either the server level or the zone level. Server level policies apply on every zone transfer query that occurs on the DNS server. Zone level policies apply only on the queries on a zone hosted on the DNS server. The most common use for zone level policies is to implement blocked or safe lists.

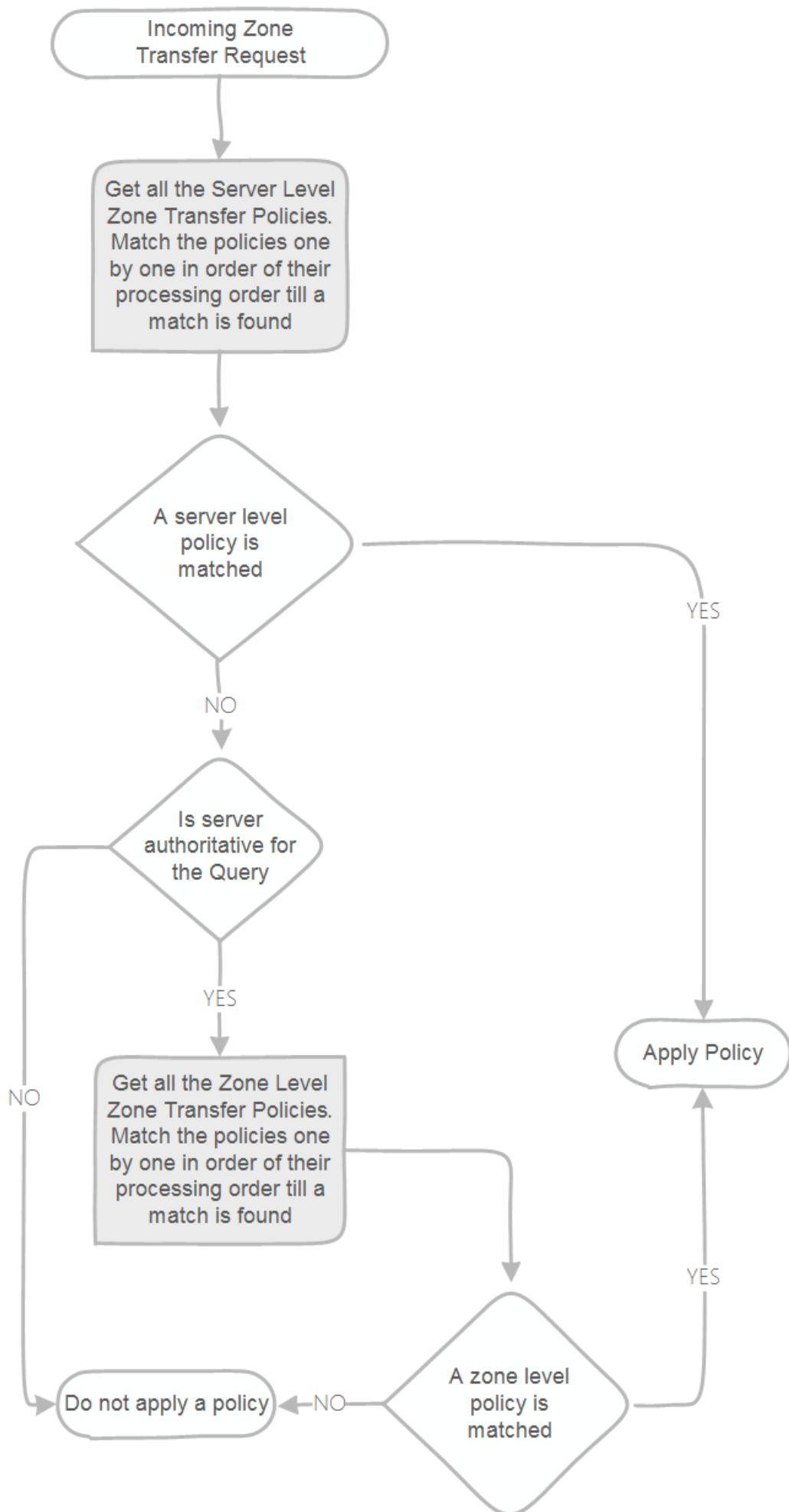
NOTE

Zone transfer policies can only use DENY or IGNORE as actions.

You can use the server level zone transfer policy below to deny a zone transfer for the contoso.com domain from a given subnet:

```
Add-DnsServerZoneTransferPolicy -Name DenyTransferOfCOnsotostFabrikam -Zone contoso.com -Action DENY -  
ClientSubnet "EQ,192.168.1.0/24"
```

You can create multiple zone transfer policies of the same level, as long as they have a different value for the processing order. When multiple policies are available, the DNS server processes incoming queries in the following manner:



Managing DNS Policies

You can create and manage DNS Policies by using PowerShell. The examples below go through different sample scenarios that you can configure through DNS Policies:

Traffic Management

You can direct traffic based on an FQDN to different servers depending on the location of the DNS client. The example below shows how to create traffic management policies to direct the customers from a certain subnet to a North American datacenter and from another subnet to a European datacenter.

```
Add-DnsServerClientSubnet -Name "NorthAmericaSubnet" -IPv4Subnet "172.21.33.0/24"
Add-DnsServerClientSubnet -Name "EuropeSubnet" -IPv4Subnet "172.17.44.0/24"
Add-DnsServerZoneScope -ZoneName "Contoso.com" -Name "NorthAmericaZoneScope"
Add-DnsServerZoneScope -ZoneName "Contoso.com" -Name "EuropeZoneScope"
Add-DnsServerResourceRecord -ZoneName "Contoso.com" -A -Name "www" -IPv4Address "172.17.97.97" -ZoneScope
"EuropeZoneScope"
Add-DnsServerResourceRecord -ZoneName "Contoso.com" -A -Name "www" -IPv4Address "172.21.21.21" -ZoneScope
"NorthAmericaZoneScope"
Add-DnsServerQueryResolutionPolicy -Name "NorthAmericaPolicy" -Action ALLOW -ClientSubnet
"eq, NorthAmericaSubnet" -ZoneScope "NorthAmericaZoneScope,1" -ZoneName "Contoso.com"
Add-DnsServerQueryResolutionPolicy -Name "EuropePolicy" -Action ALLOW -ClientSubnet "eq, EuropeSubnet" -
ZoneScope "EuropeZoneScope,1" -ZoneName contoso.com
```

The first two lines of the script create client subnet objects for North America and Europe. The two lines after that create a zone scope within the contoso.com domain, one for each region. The two lines after that create a record in each zone that associates www.contoso.com to different IP address, one for Europe, another one for North America. Finally, the last lines of the script create two DNS Query Resolution Policies, one to be applied to the North America subnet, another to the Europe subnet.

Block queries for a domain

You can use a DNS Query Resolution Policy to block queries to a domain. The example below blocks all queries to treyresearch.net:

```
Add-DnsServerQueryResolutionPolicy -Name "BlackholePolicy" -Action IGNORE -FQDN "EQ, *.treyresearch.com"
```

Block queries from a subnet

You can also block queries coming from a specific subnet. The script below creates a subnet for 172.0.33.0/24 and then creates a policy to ignore all queries coming from that subnet:

```
Add-DnsServerClientSubnet -Name "MaliciousSubnet06" -IPv4Subnet 172.0.33.0/24
Add-DnsServerQueryResolutionPolicy -Name "BlackholePolicyMalicious06" -Action IGNORE -ClientSubnet
"EQ, MaliciousSubnet06"
```

Allow recursion for internal clients

You can control recursion by using a DNS Query Resolution Policy. The sample below can be used to enable recursion for internal clients, while disabling it for external clients in a split brain scenario.

```
Set-DnsServerRecursionScope -Name . -EnableRecursion $False
Add-DnsServerRecursionScope -Name "InternalClients" -EnableRecursion $True
Add-DnsServerQueryResolutionPolicy -Name "SplitBrainPolicy" -Action ALLOW -ApplyOnRecursion -RecursionScope
"InternalClients" -ServerInterfaceIP "EQ, 10.0.0.34"
```

The first line in the script changes the default recursion scope, simply named as "." (dot) to disable recursion. The second line creates a recursion scope named *InternalClients* with recursion enabled. And the third line creates a policy to apply the newly create recursion scope to any queries coming in through a server interface that has 10.0.0.34 as an IP address.

Create a server level zone transfer policy

You can control zone transfer in a more granular form by using DNS Zone Transfer policies. The sample script below can be used to allow zone transfers for any server on a given subnet:

```
Add-DnsServerClientSubnet -Name "AllowedSubnet" -IPv4Subnet 172.21.33.0/24  
Add-DnsServerZoneTransferPolicy -Name "NorthAmericaPolicy" -Action IGNORE -ClientSubnet "ne,AllowedSubnet"
```

The first line in the script creates a subnet object named *AllowedSubnet* with the IP block 172.21.33.0/24. The second line creates a zone transfer policy to allow zone transfers to any DNS server on the subnet previously created.

Create a zone level zone transfer policy

You can also create zone level zone transfer policies. The example below ignores any request for a zone transfer for contoso.com coming in from a server interface that has an IP address of 10.0.0.33:

```
Add-DnsServerZoneTransferPolicy -Name "InternalTransfers" -Action IGNORE -ServerInterfaceIP "ne,10.0.0.33" -  
PassThru -ZoneName "contoso.com"
```

DNS Policy Scenarios

For information on how to use DNS policy for specific scenarios, see the following topics in this guide.

- [Use DNS Policy for Geo-Location Based Traffic Management with Primary Servers](#)
- [Use DNS Policy for Geo-Location Based Traffic Management with Primary-Secondary Deployments](#)
- [Use DNS Policy for Intelligent DNS Responses Based on the Time of Day](#)
- [DNS Responses Based on Time of Day with an Azure Cloud App Server](#)
- [Use DNS Policy for Split-Brain DNS Deployment](#)
- [Use DNS Policy for Split-Brain DNS in Active Directory](#)
- [Use DNS Policy for Applying Filters on DNS Queries](#)
- [Use DNS Policy for Application Load Balancing](#)
- [Use DNS Policy for Application Load Balancing With Geo-Location Awareness](#)

Use DNS Policy for Geo-Location Based Traffic Management with Primary Servers

9/1/2018 • 8 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to configure DNS Policy to allow primary DNS servers to respond to DNS client queries based on the geographical location of both the client and the resource to which the client is attempting to connect, providing the client with the IP address of the closest resource.

IMPORTANT

This scenario illustrates how to deploy DNS policy for geo-location based traffic management when you are using only primary DNS servers. You can also accomplish geo-location based traffic management when you have both primary and secondary DNS servers. If you have a primary-secondary deployment, first complete the steps in this topic, and then complete the steps that are provided in the topic [Use DNS Policy for Geo-Location Based Traffic Management with Primary-Secondary Deployments](#).

With new DNS policies, you can create a DNS policy that allows the DNS server to respond to a client query asking for the IP address of a Web server. Instances of the Web server might be located in different datacenters at different physical locations. DNS can assess the client and Web server locations, then respond to the client request by providing the client with a Web server IP address for a Web server that is physically located closer to the client.

You can use the following DNS policy parameters to control the DNS server responses to queries from DNS clients.

- **Client Subnet.** Name of a predefined client subnet. Used to verify the subnet from which the query was sent.
- **Transport Protocol.** Transport protocol used in the query. Possible entries are **UDP** and **TCP**.
- **Internet Protocol.** Network protocol used in the query. Possible entries are **IPv4** and **IPv6**.
- **Server Interface IP address.** IP address of the network interface of the DNS server which received the DNS request.
- **FQDN.** The Fully Qualified Domain Name (FQDN) of the record in the query, with the possibility of using a wild card.
- **Query Type.** Type of record being queried (A, SRV, TXT, etc.).
- **Time of Day.** Time of day the query is received.

You can combine the following criteria with a logical operator (AND/OR) to formulate policy expressions. When these expressions match, the policies are expected to perform one of the following actions.

- **Ignore.** The DNS server silently drops the query.
- **Deny.** The DNS server responds that query with a failure response.
- **Allow.** The DNS server responds back with traffic managed response.

Geo-Location Based Traffic Management Example

Following is an example of how you can use DNS policy to achieve traffic redirection on the basis of the physical location of the client that performs a DNS query.

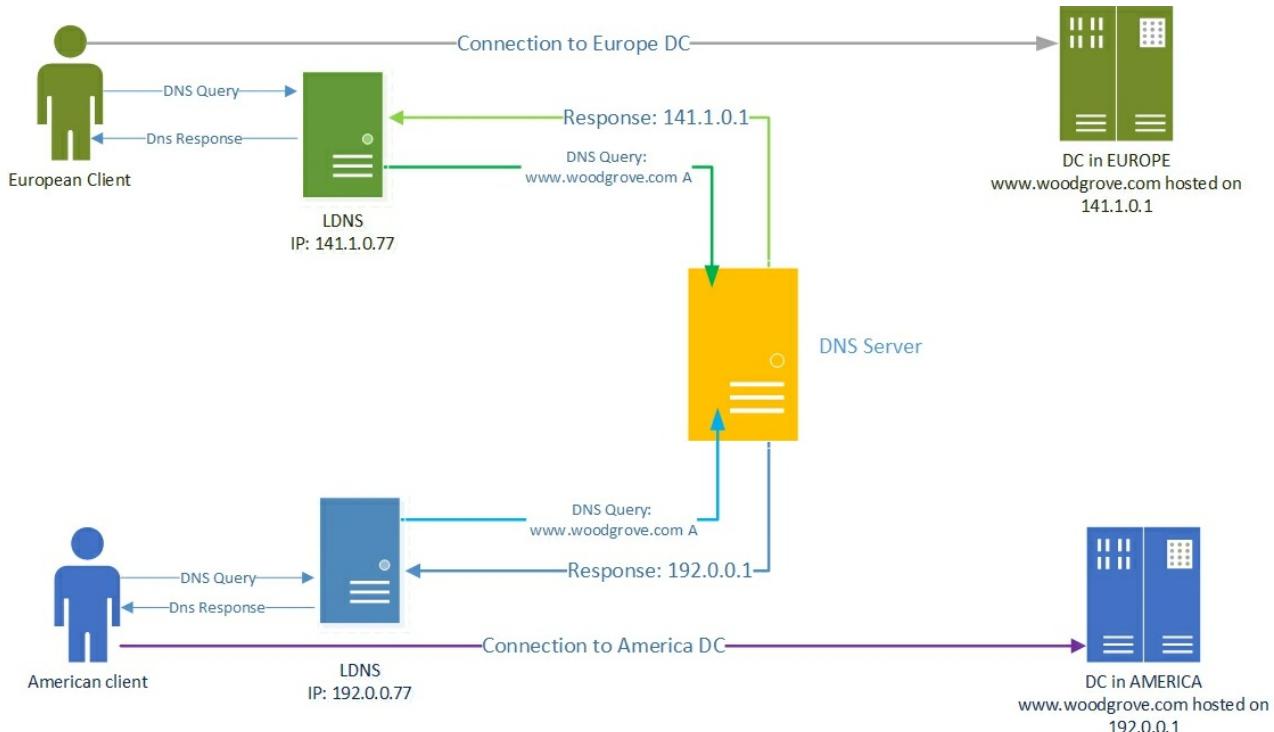
This example uses two fictional companies - Contoso Cloud Services, which provides web and domain hosting

solutions; and Woodgrove Food Services, which provides food delivery services in multiple cities across the globe, and which has a Web site named woodgrove.com.

Contoso Cloud Services has two datacenters, one in the U.S. and another in Europe. The European datacenter hosts a food ordering portal for woodgrove.com.

To ensure that woodgrove.com customers get a responsive experience from their website, Woodgrove wants European clients directed to the European datacenter and American clients directed to the U.S. datacenter. Customers located elsewhere in the world can be directed to either of the datacenters.

The following illustration depicts this scenario.



How the DNS name resolution process works

During the name resolution process, the user tries to connect to www.woodgrove.com. This results in a DNS name resolution request that is sent to the DNS server that is configured in the Network Connection properties on the user's computer. Typically, this is the DNS server provided by the local ISP acting as a caching resolver, and is referred as the LDNS.

If the DNS name is not present in the local cache of LDNS, the LDNS server forwards the query to the DNS server that is authoritative for woodgrove.com. The authoritative DNS server responds with the requested record (www.woodgrove.com) to the LDNS server, which in turn caches the record locally before sending it to the user's computer.

Because Contoso Cloud Services uses DNS Server policies, the authoritative DNS server that hosts contoso.com is configured to return geo-location based traffic managed responses. This results in the direction of European Clients to the European datacenter and the direction of American Clients to the U.S. datacenter, as depicted in the illustration.

In this scenario, the authoritative DNS server usually sees the name resolution request coming from the LDNS server and, very rarely, from the user's computer. Because of this, the source IP address in the name resolution request as seen by the authoritative DNS server is that of the LDNS server and not that of the user's computer. However, using the IP address of the LDNS server when you configure geo-location based query responses provides a fair estimate of the geo-location of the user, because the user is querying the DNS server of his local ISP.

NOTE

DNS policies utilize the sender IP in the UDP/TCP packet that contains the DNS query. If the query reaches the primary server through multiple resolver/LDNS hops, the policy will consider only the IP of the last resolver from which the DNS server receives the query.

How to configure DNS Policy for Geo-Location Based Query Responses

To configure DNS policy for geo-location based query responses, you must perform the following steps.

1. [Create the DNS Client Subnets](#)
2. [Create the Scopes of the Zone](#)
3. [Add Records to the Zone Scopes](#)
4. [Create the Policies](#)

NOTE

You must perform these steps on the DNS server that is authoritative for the zone you want to configure. Membership in **DnsAdmins**, or equivalent, is required to perform the following procedures.

The following sections provide detailed configuration instructions.

IMPORTANT

The following sections include example Windows PowerShell commands that contain example values for many parameters. Ensure that you replace example values in these commands with values that are appropriate for your deployment before you run these commands.

Create the DNS Client Subnets

The first step is to identify the subnets or IP address space of the regions for which you want to redirect traffic. For example, if you want to redirect traffic for the U.S. and Europe, you need to identify the subnets or IP address spaces of these regions.

You can obtain this information from Geo-IP maps. Based on these Geo-IP distributions, you must create the "DNS Client Subnets." A DNS Client Subnet is a logical grouping of IPv4 or IPv6 subnets from which queries are sent to a DNS server.

You can use the following Windows PowerShell commands to create DNS Client Subnets.

```
Add-DnsServerClientSubnet -Name "USSubnet" -IPv4Subnet "192.0.0.0/24"  
Add-DnsServerClientSubnet -Name "EuropeSubnet" -IPv4Subnet "141.1.0.0/24"
```

For more information, see [Add-DnsServerClientSubnet](#).

Create Zone Scopes

After the client subnets are configured, you must partition the zone whose traffic you want to redirect into two different zone scopes, one scope for each of the DNS Client Subnets that you have configured.

For example, if you want to redirect traffic for the DNS name www.woodgrove.com, you must create two different zone scopes in the woodgrove.com zone, one for the U.S. and one for Europe.

A zone scope is a unique instance of the zone. A DNS zone can have multiple zone scopes, with each zone scope containing its own set of DNS records. The same record can be present in multiple scopes, with different IP addresses or the same IP addresses.

NOTE

By default, a zone scope exists on the DNS zones. This zone scope has the same name as the zone and legacy DNS operations work on this scope.

You can use the following Windows PowerShell commands to create zone scopes.

```
Add-DnsServerZoneScope -ZoneName "woodgrove.com" -Name "USZoneScope"  
Add-DnsServerZoneScope -ZoneName "woodgrove.com" -Name "EuropeZoneScope"
```

For more information, see [Add-DnsServerZoneScope](#).

Add Records to the Zone Scopes

Now you must add the records representing the web server host into the two zone scopes.

For example, **USZoneScope** and **EuropeZoneScope**. In USZoneScope, you can add the record www.woodgrove.com with the IP address 192.0.0.1, which is located in a U.S. datacenter; and in EuropeZoneScope you can add the same record (www.woodgrove.com) with the IP address 141.1.0.1 in the European datacenter.

You can use the following Windows PowerShell commands to add records to the zone scopes.

```
Add-DnsServerResourceRecord -ZoneName "woodgrove.com" -A -Name "www" -IPv4Address "192.0.0.1" -ZoneScope  
"USZoneScope"  
  
Add-DnsServerResourceRecord -ZoneName "woodgrove.com" -A -Name "www" -IPv4Address "141.1.0.1" -ZoneScope  
"EuropeZoneScope"
```

In this example, you must also use the following Windows PowerShell commands to add records into the default zone scope to ensure that the rest of the world can still access the woodgrove.com web server from either of the two datacenters.

```
Add-DnsServerResourceRecord -ZoneName "woodgrove.com" -A -Name "www" -IPv4Address "192.0.0.1"  
Add-DnsServerResourceRecord -ZoneName "woodgrove.com" -A -Name "www" -IPv4Address "141.1.0.1"
```

The **ZoneScope** parameter is not included when you add a record in the default scope. This is the same as adding records to a standard DNS zone.

For more information, see [Add-DnsServerResourceRecord](#).

Create the Policies

After you have created the subnets, the partitions (zone scopes), and you have added records, you must create policies that connect the subnets and partitions, so that when a query comes from a source in one of the DNS client subnets, the query response is returned from the correct scope of the zone. No policies are required for mapping the default zone scope.

You can use the following Windows PowerShell commands to create a DNS policy that links the DNS Client Subnets and the zone scopes.

```
Add-DnsServerQueryResolutionPolicy -Name "USPolicy" -Action ALLOW -ClientSubnet "eq,USSubnet" -ZoneScope "USZoneScope,1" -ZoneName "woodgrove.com"
```

```
Add-DnsServerQueryResolutionPolicy -Name "EuropePolicy" -Action ALLOW -ClientSubnet "eq,EuropeSubnet" -ZoneScope "EuropeZoneScope,1" -ZoneName "woodgrove.com"
```

For more information, see [Add-DnsServerQueryResolutionPolicy](#).

Now the DNS server is configured with the required DNS policies to redirect traffic based on geo-location.

When the DNS server receives name resolution queries, the DNS server evaluates the fields in the DNS request against the configured DNS policies. If the source IP address in the name resolution request matches any of the policies, the associated zone scope is used to respond to the query, and the user is directed to the resource that is geographically closest to them.

You can create thousands of DNS policies according to your traffic management requirements, and all new policies are applied dynamically - without restarting the DNS server - on incoming queries.

Use DNS Policy for Geo-Location Based Traffic Management with Primary-Secondary Deployments

9/1/2018 • 8 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to create DNS policy for geo-location based traffic management when your DNS deployment includes both primary and secondary DNS servers.

The previous scenario, [Use DNS Policy for Geo-Location Based Traffic Management with Primary Servers](#), provided instructions for configuring DNS policy for geo-location based traffic management on a primary DNS server. In the Internet infrastructure, however, the DNS servers are widely deployed in a primary-secondary model, where the writable copy of a zone is stored on select and secure primary servers, and read-only copies of the zone are kept on multiple secondary servers.

The secondary servers use the zone transfer protocols Authoritative Transfer (AXFR) and Incremental Zone Transfer (IXFR) to request and receive zone updates that include new changes to the zones on the primary DNS servers.

NOTE

For more information about AXFR, see the Internet Engineering Task Force (IETF) [Request for Comments 5936](#). For more information about IXFR, see the Internet Engineering Task Force (IETF) [Request for Comments 1995](#).

Primary-Secondary Geo-Location Based Traffic Management Example

Following is an example of how you can use DNS policy in a primary-secondary deployment to achieve traffic redirection on the basis of the physical location of the client that performs a DNS query.

This example uses two fictional companies - Contoso Cloud Services, which provides web and domain hosting solutions; and Woodgrove Food Services, which provides food delivery services in multiple cities across the globe, and which has a Web site named woodgrove.com.

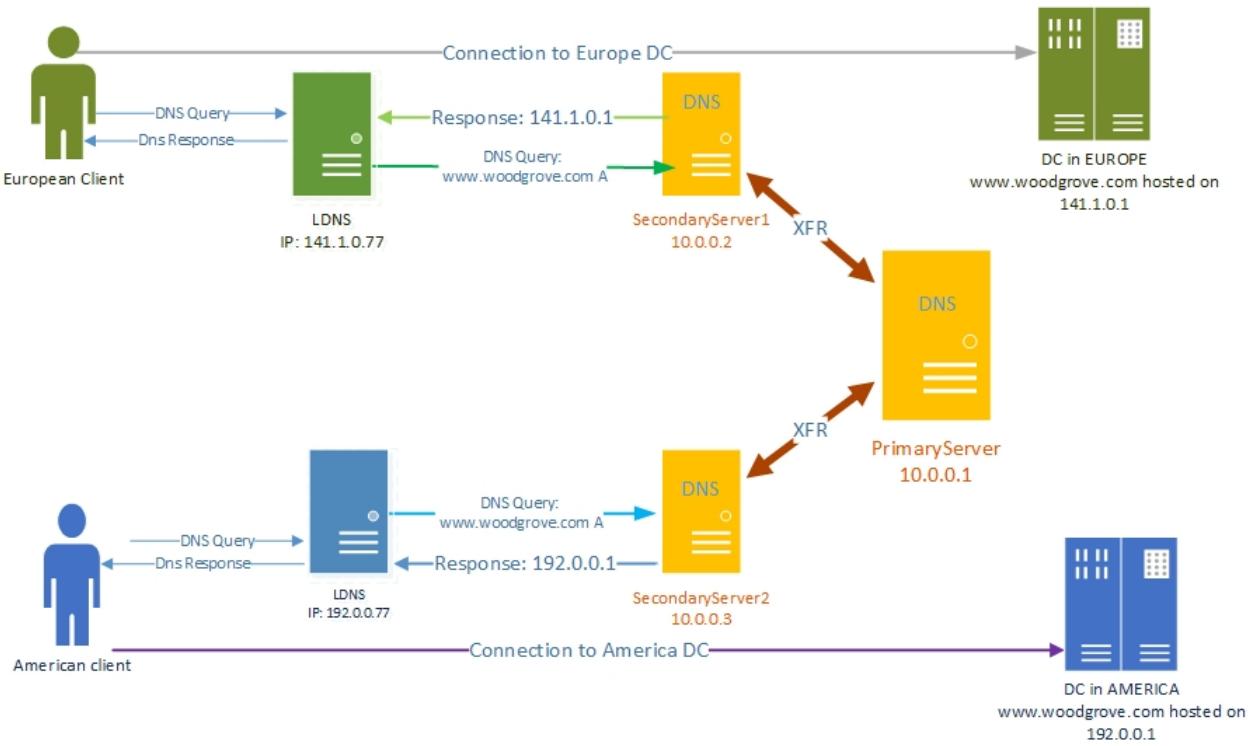
To ensure that woodgrove.com customers get a responsive experience from their website, Woodgrove wants European clients directed to the European datacenter and American clients directed to the U.S. datacenter. Customers located elsewhere in the world can be directed to either of the datacenters.

Contoso Cloud Services has two datacenters, one in the U.S. and another in Europe, upon which Contoso hosts its food ordering portal for woodgrove.com.

The Contoso DNS deployment includes two secondary servers: **SecondaryServer1**, with the IP address 10.0.0.2; and **SecondaryServer2**, with the IP address 10.0.0.3. These secondary servers are acting as name servers in the two different regions, with SecondaryServer1 located in Europe and SecondaryServer2 located in the U.S.

There is a primary writable zone copy on **PrimaryServer** (IP address 10.0.0.1), where the zone changes are made. With regular zone transfers to the secondary servers, the secondary servers are always up to date with any new changes to the zone on the PrimaryServer.

The following illustration depicts this scenario.



How the DNS Primary-Secondary System Works

When you deploy geo-location based traffic management in a primary-secondary DNS deployment, it is important to understand how normal primary-secondary zone transfers occur before learning about zone scope level transfers. The following sections provide information on zone and zone scope level transfers.

- [Zone transfers in a DNS primary-secondary deployment](#)
- [Zone scope level transfers in a DNS primary-secondary deployment](#)

Zone transfers in a DNS primary-secondary deployment

You can create a DNS primary-secondary deployment and synchronize zones with the following steps.

1. When you install DNS, the primary zone is created on the primary DNS server.
2. On the secondary server, create the zones and specify the primary servers.
3. On the primary servers, you can add the secondary servers as trusted secondaries on the primary zone.
4. The secondary zones make a full zone transfer request (AXFR) and receive the copy of the zone.
5. When needed, the primary servers send notifications to the secondary servers about zone updates.
6. Secondary servers make an incremental zone transfer request (IXFR). Because of this, the secondary servers remain synchronized with the primary server.

Zone scope level transfers in a DNS primary-secondary deployment

The traffic management scenario requires additional steps to partition the zones into different zone scopes. Because of this, additional steps are required to transfer the data inside the zone scopes to the secondary servers, and to transfer policies and DNS Client Subnets to the secondary servers.

After you configure your DNS infrastructure with primary and secondary servers, zone scope level transfers are performed automatically by DNS, using the following processes.

To ensure the Zone scope level transfer, DNS servers use the Extension Mechanisms for DNS (EDNS0) OPT RR. All zone transfer (AXFR or IXFR) requests from the zones with scopes originate with an EDNS0 OPT RR, whose option ID is set to "65433" by default. For more information about EDNS0, see the IETF [Request for Comments 6891](#).

The value of the OPT RR is the zone scope name for which the request is being sent. When a primary DNS server

receives this packet from a trusted secondary server, it interprets the request as coming for that zone scope.

If the primary server has that zone scope it responds with the transfer (XFR) data from that scope. The response contains an OPT RR with the same option ID "65433" and value set to the same zone scope. The secondary servers receive this response, retrieve the scope information from the response, and update that particular scope of the zone.

After this process, the primary server maintains a list of trusted secondaries which have sent such a zone scope request for notifications.

For any further update in a zone scope, an IXFR notification is sent to the secondary servers, with the same OPT RR. The zone scope receiving that notification makes the IXFR request containing that OPT RR and the same process as described above follows.

How to configure DNS Policy for Primary-Secondary Geo-Location Based Traffic Management

Before you begin, ensure that you have completed all of the steps in the topic [Use DNS Policy for Geo-Location Based Traffic Management with Primary Servers](#), and your primary DNS server is configured with zones, zone scopes, DNS Client Subnets, and DNS policy.

NOTE

The instructions in this topic to copy DNS Client Subnets, zone scopes, and DNS policies from DNS primary servers to DNS secondary servers are for your initial DNS setup and validation. In the future you might want to change the DNS Client Subnets, zone scopes, and policies settings on the primary server. In this circumstance, you can create automation scripts to keep the secondary servers synchronized with the primary server.

To configure DNS policy for primary-secondary geo-location based query responses, you must perform the following steps.

- [Create the Secondary Zones](#)
- [Configure the Zone Transfer Settings on the Primary Zone](#)
- [Copy the DNS Client Subnets](#)
- [Create the Zone Scopes on the Secondary Server](#)
- [Configure DNS policy](#)

The following sections provide detailed configuration instructions.

IMPORTANT

The following sections include example Windows PowerShell commands that contain example values for many parameters. Ensure that you replace example values in these commands with values that are appropriate for your deployment before you run these commands.

Membership in **DnsAdmins**, or equivalent, is required to perform the following procedures.

Create the Secondary Zones

You can create the secondary copy of the zone you want to replicate to SecondaryServer1 and SecondaryServer2 (assuming the cmdlets are being executed remotely from a single management client).

For example, you can create the secondary copy of www.woodgrove.com on SecondaryServer1 and SecondarySesrver2.

You can use the following Windows PowerShell commands to create the secondary zones.

```
Add-DnsServerSecondaryZone -Name "woodgrove.com" -ZoneFile "woodgrove.com.dns" -MasterServers 10.0.0.1 -ComputerName SecondaryServer1  
  
Add-DnsServerSecondaryZone -Name "woodgrove.com" -ZoneFile "woodgrove.com.dns" -MasterServers 10.0.0.1 -ComputerName SecondaryServer2
```

For more information, see [Add-DnsServerSecondaryZone](#).

Configure the Zone Transfer Settings on the Primary Zone

You must configure the primary zone settings so that:

1. Zone transfers from the primary server to the specified secondary servers are allowed.
2. Zone update notifications are sent by the primary server to the secondary servers.

You can use the following Windows PowerShell commands to configure the zone transfer settings on the primary zone.

NOTE

In the following example command, the parameter **-Notify** specifies that the primary server will send notifications about updates to the select list of secondaries.

```
Set-DnsServerPrimaryZone -Name "woodgrove.com" -Notify Notify -SecondaryServers "10.0.0.2,10.0.0.3" -SecureSecondaries TransferToSecureServers -ComputerName PrimaryServer
```

For more information, see [Set-DnsServerPrimaryZone](#).

Copy the DNS Client Subnets

You must copy the DNS Client Subnets from the primary server to the secondary servers.

You can use the following Windows PowerShell commands to copy the subnets to the secondary servers.

```
Get-DnsServerClientSubnet -ComputerName PrimaryServer | Add-DnsServerClientSubnet -ComputerName SecondaryServer1  
  
Get-DnsServerClientSubnet -ComputerName PrimaryServer | Add-DnsServerClientSubnet -ComputerName SecondaryServer2
```

For more information, see [Add-DnsServerClientSubnet](#).

Create the Zone Scopes on the Secondary Server

You must create the zone scopes on the secondary servers. In DNS, the zone scopes also start requesting XFRs from the primary server. With any change on the zone scopes on the primary server, a notification that contains the zone scope information is sent to the secondary servers. The secondary servers can then update their zone scopes with incremental change.

You can use the following Windows PowerShell commands to create the zone scopes on the secondary servers.

```
Get-DnsServerZoneScope -ZoneName "woodgrove.com" -ComputerName PrimaryServer | Add-DnsServerZoneScope -ZoneName "woodgrove.com" -ComputerName SecondaryServer1 -ErrorAction Ignore  
  
Get-DnsServerZoneScope -ZoneName "woodgrove.com" -ComputerName PrimaryServer | Add-DnsServerZoneScope -ZoneName "woodgrove.com" -ComputerName SecondaryServer2 -ErrorAction Ignore
```

NOTE

In these example commands, the **-ErrorAction Ignore** parameter is included, because a default zone scope exists on every zone. The default zone scope cannot be created or deleted. Pipelining will result in an attempt to create that scope and it will fail. Alternatively, you can create the non-default zone scopes on two secondary zones.

For more information, see [Add-DnsServerZoneScope](#).

Configure DNS policy

After you have created the subnets, the partitions (zone scopes), and you have added records, you must create policies that connect the subnets and partitions, so that when a query comes from a source in one of the DNS client subnets, the query response is returned from the correct scope of the zone. No policies are required for mapping the default zone scope.

You can use the following Windows PowerShell commands to create a DNS policy that links the DNS Client Subnets and the zone scopes.

```
$policy = Get-DnsServerQueryResolutionPolicy -ZoneName "woodgrove.com" -ComputerName PrimaryServer  
$policy | Add-DnsServerQueryResolutionPolicy -ZoneName "woodgrove.com" -ComputerName SecondaryServer1  
$policy | Add-DnsServerQueryResolutionPolicy -ZoneName "woodgrove.com" -ComputerName SecondaryServer2
```

For more information, see [Add-DnsServerQueryResolutionPolicy](#).

Now the secondary DNS servers are configured with the required DNS policies to redirect traffic based on geo-location.

When the DNS server receives name resolution queries, the DNS server evaluates the fields in the DNS request against the configured DNS policies. If the source IP address in the name resolution request matches any of the policies, the associated zone scope is used to respond to the query, and the user is directed to the resource that is geographically closest to them.

You can create thousands of DNS policies according to your traffic management requirements, and all new policies are applied dynamically - without restarting the DNS server - on incoming queries.

Use DNS Policy for Intelligent DNS Responses Based on the Time of Day

9/1/2018 • 8 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to distribute application traffic across different geographically distributed instances of an application by using DNS policies that are based on the time of day.

This scenario is useful in situations where you want to direct traffic in one time zone to alternate application servers, such as Web servers, that are located in another time zone. This allows you to load balance traffic across application instances during peak time periods when your primary servers are overloaded with traffic.

Example of Intelligent DNS Responses Based on the Time of Day

Following is an example of how you can use DNS policy to balance application traffic based on the time of day.

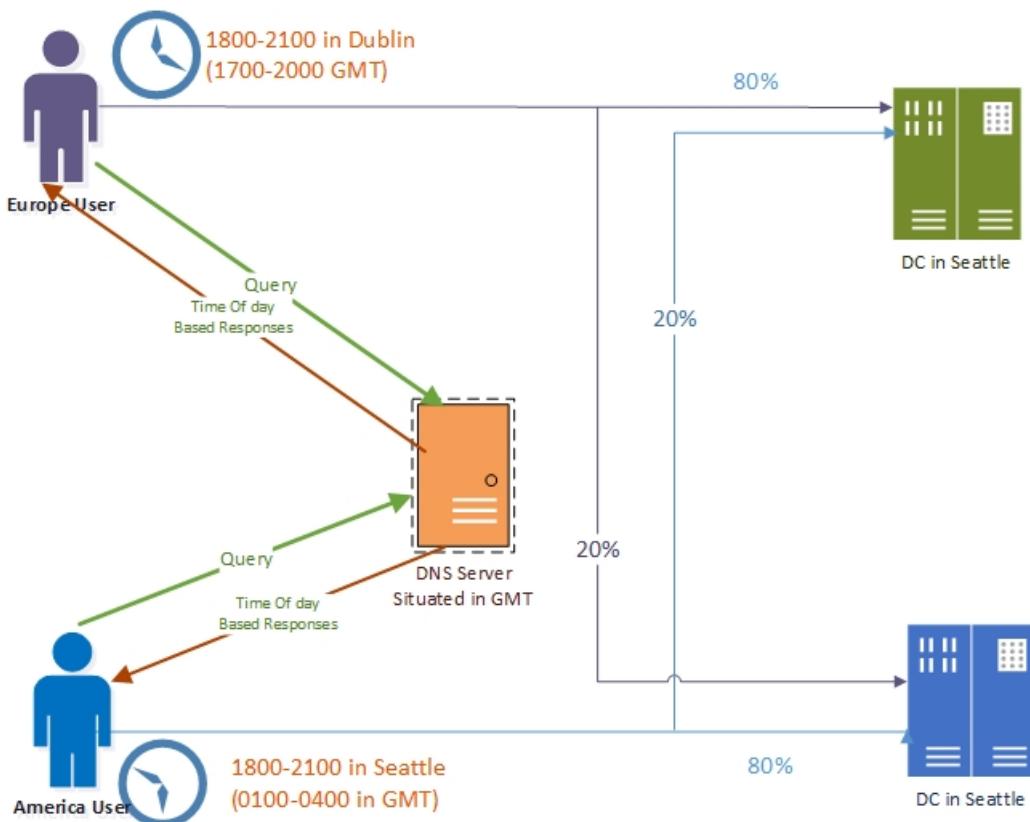
This example uses one fictional company, Contoso Gift Services, which provides online gifting solutions across the globe through their Web site, contosogiftservices.com.

The contosogiftservices.com Web site is hosted in two datacenters, one in Seattle (North America) and another in Dublin (Europe). The DNS servers are configured for sending geo-location aware responses using DNS policy. With a recent surge in business, contosogiftservices.com has a higher number of visitors every day, and some of the customers have reported service availability issues.

Contoso Gift Services performs a site analysis, and discovers that every evening between 6 PM and 9 PM local time, there is a surge in the traffic to the Web servers. The Web servers cannot scale to handle the increased traffic at these peak hours, resulting in denial of service to customers. The same peak hour traffic overload happens in both the European and American datacenters. At other times of day, the servers handle traffic volumes that are well below their maximum capability.

To ensure that contosogiftservices.com customers get a responsive experience from the Web site, Contoso Gift Services wants to redirect some Dublin traffic to the Seattle application servers between 6 PM and 9 PM in Dublin; and they want to redirect some Seattle traffic to the Dublin application servers between 6 PM and 9 PM in Seattle.

The following illustration depicts this scenario.



How Intelligent DNS Responses Based on Time of Day Works

When the DNS server is configured with time of day DNS policy, between 6 PM and 9 PM at each geographical location, the DNS server does the following.

- Answers the first four queries it receives with the IP address of the Web server in the local datacenter.
- Answers the fifth query it receives with the IP address of the Web server in the remote datacenter.

This policy-based behavior offloads twenty per cent of the local Web server's traffic load to the remote Web server, easing the strain on the local application server and improving site performance for customers.

During off-peak hours, the DNS servers perform normal geo-locations based traffic management. In addition, DNS clients that send queries from locations other than North America or Europe, the DNS server load balances the traffic across the Seattle and Dublin datacenters.

When multiple DNS policies are configured in DNS, they are an ordered set of rules, and they are processed by DNS from highest priority to lowest priority. DNS uses the first policy that matches the circumstances, including time of day. For this reason, more specific policies should have higher priority. If you create time of day policies and give them high priority in the list of policies, DNS processes and uses these policies first if they match the parameters of the DNS client query and the criteria defined in the policy. If they don't match, DNS moves down the list of policies to process the default policies until it finds a match.

For more information about policy types and criteria, see [DNS Policies Overview](#).

How to Configure DNS Policy for Intelligent DNS Responses Based on Time of Day

To configure DNS policy for time of day application load balancing based query responses, you must perform the following steps.

- [Create the DNS Client Subnets](#)
- [Create the Zone Scopes](#)
- [Add Records to the Zone Scopes](#)
- [Create the DNS Policies](#)

NOTE

You must perform these steps on the DNS server that is authoritative for the zone you want to configure. Membership in **DnsAdmins**, or equivalent, is required to perform the following procedures.

The following sections provide detailed configuration instructions.

IMPORTANT

The following sections include example Windows PowerShell commands that contain example values for many parameters. Ensure that you replace example values in these commands with values that are appropriate for your deployment before you run these commands.

Create the DNS Client Subnets

The first step is to identify the subnets or IP address space of the regions for which you want to redirect traffic. For example, if you want to redirect traffic for the U.S. and Europe, you need to identify the subnets or IP address spaces of these regions.

You can obtain this information from Geo-IP maps. Based on these Geo-IP distributions, you must create the "DNS Client Subnets." A DNS Client Subnet is a logical grouping of IPv4 or IPv6 subnets from which queries are sent to a DNS server.

You can use the following Windows PowerShell commands to create DNS Client Subnets.

```
Add-DnsServerClientSubnet -Name "AmericaSubnet" -IPv4Subnet "192.0.0.0/24, 182.0.0.0/24"  
Add-DnsServerClientSubnet -Name "EuropeSubnet" -IPv4Subnet "141.1.0.0/24, 151.1.0.0/24"
```

For more information, see [Add-DnsServerClientSubnet](#).

Create the Zone Scopes

After the client subnets are configured, you must partition the zone whose traffic you want to redirect into two different zone scopes, one scope for each of the DNS Client Subnets that you have configured.

For example, if you want to redirect traffic for the DNS name www.contosogiftservices.com, you must create two different zone scopes in the contosogiftservices.com zone, one for the U.S. and one for Europe.

A zone scope is a unique instance of the zone. A DNS zone can have multiple zone scopes, with each zone scope containing its own set of DNS records. The same record can be present in multiple scopes, with different IP addresses or the same IP addresses.

NOTE

By default, a zone scope exists on the DNS zones. This zone scope has the same name as the zone, and legacy DNS operations work on this scope.

You can use the following Windows PowerShell commands to create zone scopes.

```
Add-DnsServerZoneScope -ZoneName "contosogiftservices.com" -Name "SeattleZoneScope"  
Add-DnsServerZoneScope -ZoneName "contosogiftservices.com" -Name "DublinZoneScope"
```

For more information, see [Add-DnsServerZoneScope](#).

Add Records to the Zone Scopes

Now you must add the records representing the web server host into the two zone scopes.

For example, in **SeattleZoneScope**, the record **www.contosogiftservices.com** is added with IP address 192.0.0.1, which is located in a Seattle datacenter. Similarly, in **DublinZoneScope**, the record **www.contosogiftservices.com** is added with IP address 141.1.0.3 in the Dublin datacenter

You can use the following Windows PowerShell commands to add records to the zone scopes.

```
Add-DnsServerResourceRecord -ZoneName "contosogiftservices.com" -A -Name "www" -IPv4Address "192.0.0.1" -ZoneScope "SeattleZoneScope"

Add-DnsServerResourceRecord -ZoneName "contosogiftservices.com" -A -Name "www" -IPv4Address "141.1.0.3" -ZoneScope "DublinZoneScope"
```

The ZoneScope parameter is not included when you add a record in the default scope. This is the same as adding records to a standard DNS zone.

For more information, see [Add-DnsServerResourceRecord](#).

Create the DNS Policies

After you have created the subnets, the partitions (zone scopes), and you have added records, you must create policies that connect the subnets and partitions, so that when a query comes from a source in one of the DNS client subnets, the query response is returned from the correct scope of the zone. No policies are required for mapping the default zone scope.

After you configure these DNS policies, the DNS server behavior is as follows:

1. European DNS clients receive the IP address of the Web server in the Dublin datacenter in their DNS query response.
2. American DNS clients receive the IP address of the Web server in the Seattle datacenter in their DNS query response.
3. Between 6 PM and 9 PM in Dublin, 20% of the queries from European clients receive the IP address of the Web server in the Seattle datacenter in their DNS query response.
4. Between 6 PM and 9 PM in Seattle, 20% of the queries from the American clients receive the IP address of the Web server in the Dublin datacenter in their DNS query response.
5. Half of the queries from the rest of the world receive the IP address of the Seattle datacenter and the other half receive the IP address of the Dublin datacenter.

You can use the following Windows PowerShell commands to create a DNS policy that links the DNS Client Subnets and the zone scopes.

NOTE

In this example, the DNS server is in the GMT time zone, so the peak hour time periods must be expressed in the equivalent GMT time.

```
Add-DnsServerQueryResolutionPolicy -Name "America6To9Policy" -Action ALLOW -ClientSubnet "eq,AmericaSubnet" -  
ZoneScope "SeattleZoneScope,4;DublinZoneScope,1" -TimeOfDay "EQ,01:00-04:00" -ZoneName  
"contosogiftservices.com" -ProcessingOrder 1  
  
Add-DnsServerQueryResolutionPolicy -Name "Europe6To9Policy" -Action ALLOW -ClientSubnet "eq,EuropeSubnet" -  
ZoneScope "SeattleZoneScope,1;DublinZoneScope,4" -TimeOfDay "EQ,17:00-20:00" -ZoneName  
"contosogiftservices.com" -ProcessingOrder 2  
  
Add-DnsServerQueryResolutionPolicy -Name "AmericaPolicy" -Action ALLOW -ClientSubnet "eq,AmericaSubnet" -  
ZoneScope "SeattleZoneScope,1" -ZoneName "contosogiftservices.com" -ProcessingOrder 3  
  
Add-DnsServerQueryResolutionPolicy -Name "EuropePolicy" -Action ALLOW -ClientSubnet "eq,EuropeSubnet" -  
ZoneScope "DublinZoneScope,1" -ZoneName "contosogiftservices.com" -ProcessingOrder 4  
  
Add-DnsServerQueryResolutionPolicy -Name "RestOfWorldPolicy" -Action ALLOW --ZoneScope  
"DublinZoneScope,1;SeattleZoneScope,1" -ZoneName "contosogiftservices.com" -ProcessingOrder 5
```

For more information, see [Add-DnsServerQueryResolutionPolicy](#).

Now the DNS server is configured with the required DNS policies to redirect traffic based on geo-location and time of day.

When the DNS server receives name resolution queries, the DNS server evaluates the fields in the DNS request against the configured DNS policies. If the source IP address in the name resolution request matches any of the policies, the associated zone scope is used to respond to the query, and the user is directed to the resource that is geographically closest to them.

You can create thousands of DNS policies according to your traffic management requirements, and all new policies are applied dynamically - without restarting the DNS server - on incoming queries.

DNS Responses Based on Time of Day with an Azure Cloud App Server

9/1/2018 • 6 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to distribute application traffic across different geographically distributed instances of an application by using DNS policies that are based on the time of day.

This scenario is useful in situations where you want to direct traffic in one time zone to alternate application servers, such as Web servers that are hosted on Microsoft Azure, that are located in another time zone. This allows you to load balance traffic across application instances during peak time periods when your primary servers are overloaded with traffic.

NOTE

To learn how to use DNS policy for intelligent DNS responses without using Azure, see [Use DNS Policy for Intelligent DNS Responses Based on the Time of Day](#).

Example of Intelligent DNS Responses Based on the Time of Day with Azure Cloud App Server

Following is an example of how you can use DNS policy to balance application traffic based on the time of day.

This example uses one fictional company, Contoso Gift Services, which provides online gifting solutions across the globe through their Web site, contosogiftservices.com.

The contosogiftservices.com web site is hosted only at a single on-premises datacenter in Seattle (with public IP 192.68.30.2).

The DNS server is also located in the on-premises datacenter.

With a recent surge in business, contosogiftservices.com has a higher number of visitors every day, and some of the customers have reported service availability issues.

Contoso Gift Services performs a site analysis, and discovers that every evening between 6 PM and 9 PM local time, there is a surge in the traffic to the Seattle Web server. The Web server cannot scale to handle the increased traffic at these peak hours, resulting in denial of service to customers.

To ensure that contosogiftservices.com customers get a responsive experience from the Web site, Contoso Gift Services decides that during these hours it will rent a virtual machine (VM) on Microsoft Azure to host a copy of its Web server.

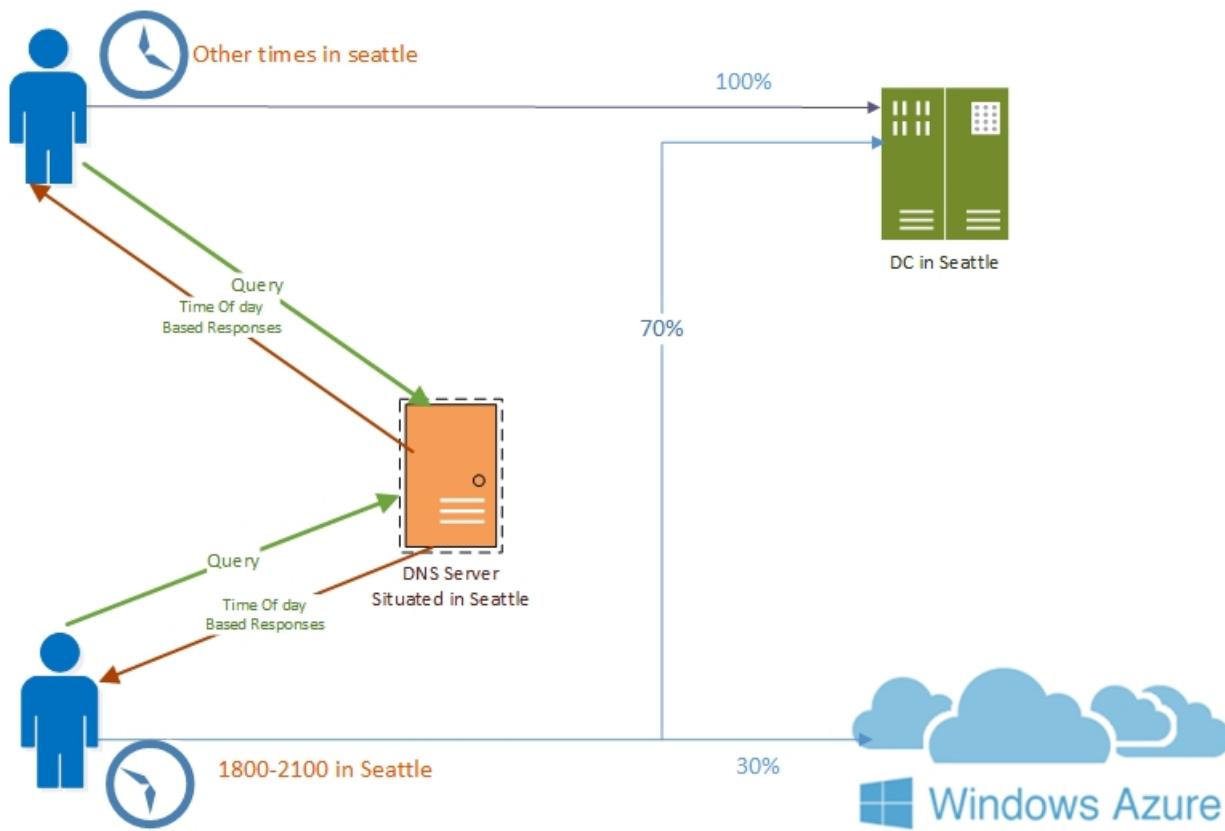
Contoso Gift Services gets a public IP address from Azure for the VM (192.68.31.44) and develops the automation to deploy the Web Server every day on Azure between 5-10 PM, allowing for a one hour contingency period.

NOTE

For more information about Azure VMs, see [Virtual Machines documentation](#)

The DNS servers are configured with zone scopes and DNS policies so that between 5-9 PM every day, 30% of queries are sent to the instance of the Web server that is running in Azure.

The following illustration depicts this scenario.



How Intelligent DNS Responses Based on Time of Day with Azure App Server Works

This article demonstrates how to configure the DNS server to answer DNS queries with two different application server IP addresses - one web server is in Seattle and the other is in an Azure datacenter.

After the configuration of a new DNS policy that is based on the peak hours of 6 PM to 9 PM in Seattle, the DNS server sends seventy per cent of the DNS responses to clients containing the IP address of the Seattle Web server, and thirty per cent of the DNS responses to clients containing the IP address of the Azure Web server, thereby directing client traffic to the new Azure Web server, and preventing the Seattle Web server from becoming overloaded.

At all other times of day, the normal query processing takes place and responses are sent from default zone scope which contains a record for the web server in the on-premises datacenter.

The TTL of 10 minutes on the Azure record ensures that the record is expired from the LDNS cache before the VM is removed from Azure. One of the benefits of such scaling is that you can keep your DNS data on-premises, and keep scaling out to Azure as demand requires.

How to Configure DNS Policy for Intelligent DNS Responses Based on Time of Day with Azure App Server

To configure DNS policy for time of day application load balancing based query responses, you must perform the following steps.

- Create the Zone Scopes

- [Add Records to the Zone Scopes](#)
- [Create the DNS Policies](#)

NOTE

You must perform these steps on the DNS server that is authoritative for the zone you want to configure. Membership in DnsAdmins, or equivalent, is required to perform the following procedures.

The following sections provide detailed configuration instructions.

IMPORTANT

The following sections include example Windows PowerShell commands that contain example values for many parameters. Ensure that you replace example values in these commands with values that are appropriate for your deployment before you run these commands.

Create the Zone Scopes

A zone scope is a unique instance of the zone. A DNS zone can have multiple zone scopes, with each zone scope containing its own set of DNS records. The same record can be present in multiple scopes, with different IP addresses or the same IP addresses.

NOTE

By default, a zone scope exists on the DNS zones. This zone scope has the same name as the zone, and legacy DNS operations work on this scope.

You can use the following example command to create a zone scope to host the Azure records.

```
Add-DnsServerZoneScope -ZoneName "contosogiftservices.com" -Name "AzureZoneScope"
```

For more information, see [Add-DnsServerZoneScope](#)

Add Records to the Zone Scopes

The next step is to add the records representing the Web server host into the zone scopes.

In AzureZoneScope, the record www.contosogiftservices.com is added with IP address 192.68.31.44, which is located in the Azure public cloud.

Similarly, in the default zone scope (contosogiftservices.com), a record (www.contosogiftservices.com) is added with IP address 192.68.30.2 of the Web server running in the Seattle on-premises datacenter.

In the second cmdlet below, the –ZoneScope parameter is not included. Because of this, the records are added in the default ZoneScope.

In addition, the TTL of the record for Azure VMs is kept at 600s (10 mins) so that the LDNS do not cache it for a longer time - which would interfere with load balancing. Also, the Azure VMs are available for 1 extra hour as a contingency to ensure that even clients with cached records are able to resolve.

```
Add-DnsServerResourceRecord -ZoneName "contosogiftservices.com" -A -Name "www" -IPv4Address "192.68.31.44" -ZoneScope "AzureZoneScope" -TimeToLive 600  
Add-DnsServerResourceRecord -ZoneName "contosogiftservices.com" -A -Name "www" -IPv4Address "192.68.30.2"
```

For more information, see [Add-DnsServerResourceRecord](#).

Create the DNS Policies

After the zone scopes are created, you can create DNS policies that distribute the incoming queries across these scopes so that the following occurs.

1. From 6 PM to 9 PM daily, 30% of clients receive the IP address of the Web server in the Azure datacenter in the DNS response, while 70% of clients receive the IP address of the Seattle on-premises Web server.
2. At all other times, all the clients receive the IP address of the Seattle on-premises Web server.

The time of the day has to be expressed in local time of the DNS server.

You can use the following example command to create the DNS policy.

```
Add-DnsServerQueryResolutionPolicy -Name "Contoso6To9Policy" -Action ALLOW --ZoneScope  
"contosogiftservices.com,7;AzureZoneScope,3" -TimeOfDay "EQ,18:00-21:00" -ZoneName "contosogiftservices.com" -  
ProcessingOrder 1
```

For more information, see [Add-DnsServerQueryResolutionPolicy](#).

Now the DNS server is configured with the required DNS policies to redirect traffic to the Azure Web server based on time of day.

Note the expression:

```
-ZoneScope "contosogiftservices.com,7;AzureZoneScope,3" -TimeOfDay "EQ,18:00-21:00"
```

This expression configures the DNS server with a ZoneScope and weight combination that instructs the DNS server to send the IP address of the Seattle Web server seventy per cent of the time, while sending the IP address of the Azure Web server thirty per cent of the time.

You can create thousands of DNS policies according to your traffic management requirements, and all new policies are applied dynamically - without restarting the DNS server - on incoming queries.

Use DNS Policy for Split-Brain DNS Deployment

9/1/2018 • 8 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016

You can use this topic to learn how to configure DNS policy in Windows Server® 2016 for split-brain DNS deployments, where there are two versions of a single zone - one for the internal users on your organization intranet, and one for the external users, who are typically users on the Internet.

NOTE

For information on how to use DNS Policy for split-brain DNS deployment with Active Directory integrated DNS Zones, see [Use DNS Policy for Split-Brain DNS in Active Directory](#).

Previously, this scenario required that DNS administrators maintain two different DNS servers, each providing services to each set of users, internal and external. If only a few records inside the zone were split-brained or both instances of the zone (internal and external) were delegated to the same parent domain, this became a management conundrum.

Another configuration scenario for split-brain deployment is Selective Recursion Control for DNS name resolution. In some circumstances, the Enterprise DNS servers are expected to perform recursive resolution over the Internet for the internal users, while they also must act as authoritative name servers for external users, and block recursion for them.

This topic contains the following sections.

- [Example of DNS Split-Brain Deployment](#)
- [Example of DNS Selective Recursion Control](#)

Example of DNS Split-Brain Deployment

Following is an example of how you can use DNS policy to accomplish the previously described scenario of split-brain DNS.

This section contains the following topics.

- [How DNS Split-Brain Deployment Works](#)
- [How to Configure DNS Split-Brain Deployment](#)

This example uses one fictional company, Contoso, which maintains a career Web site at www.career.contoso.com.

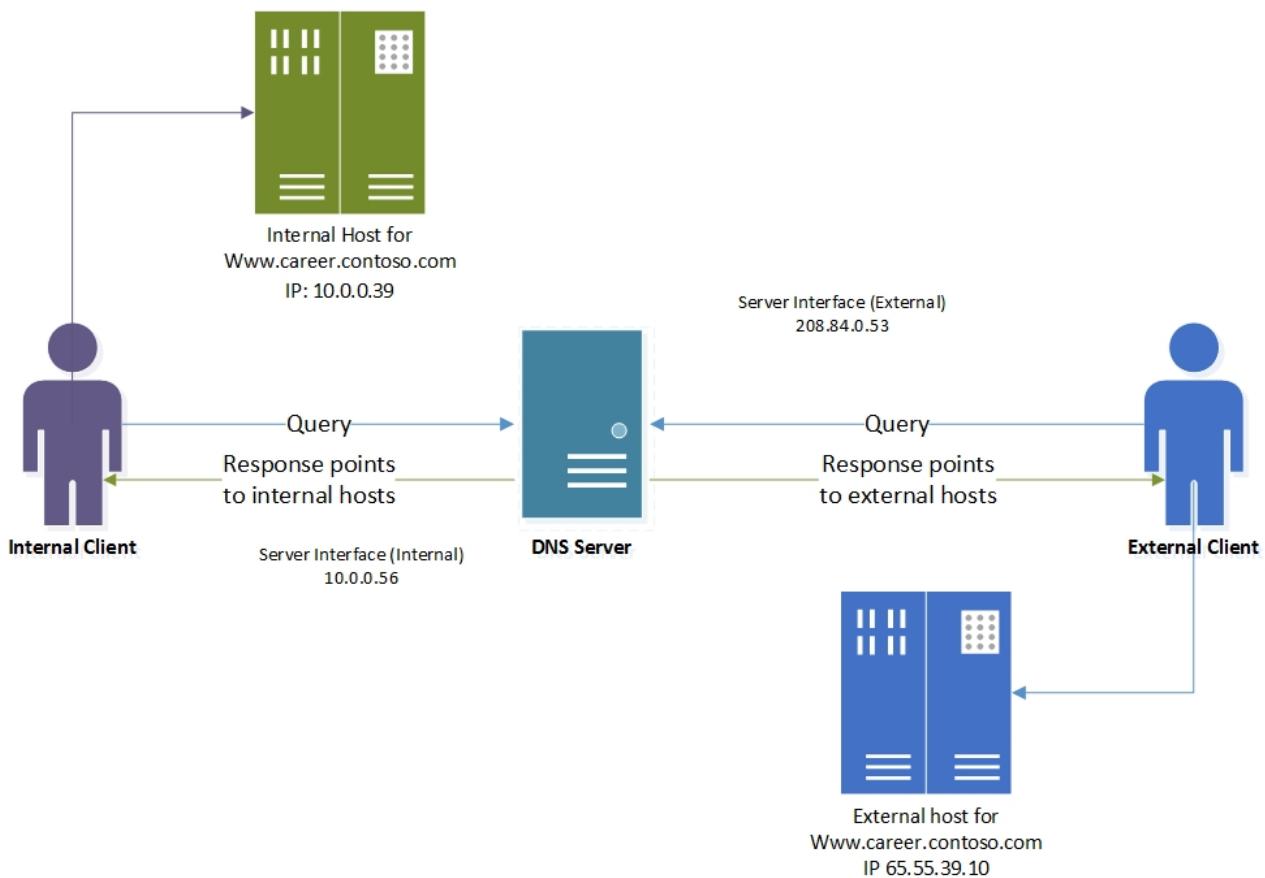
The site has two versions, one for the internal users where internal job postings are available. This internal site is available at the local IP address 10.0.0.39.

The second version is the public version of the same site, which is available at the public IP address 65.55.39.10.

In the absence of DNS policy, the administrator is required to host these two zones on separate Windows Server DNS servers and manage them separately.

Using DNS policies these zones can now be hosted on the same DNS server.

The following illustration depicts this scenario.



How DNS Split-Brain Deployment Works

When the DNS server is configured with the required DNS policies, each name resolution request is evaluated against the policies on the DNS server.

The server interface is used in this example as the criteria to differentiate between the internal and external clients.

If the server interface upon which the query is received matches any of the policies, the associated zone scope is used to respond to the query.

So, in our example, the DNS queries for **www.career.contoso.com** that are received on the private IP (**10.0.0.56**) receive a DNS response that contains an internal IP address; and the DNS queries that are received on the public network interface receive a DNS response that contains the public IP address in the default zone scope (this is the same as normal query resolution).

How to Configure DNS Split-Brain Deployment

To configure DNS Split-Brain Deployment by using DNS Policy, you must use the following steps.

- [Create the Zone Scopes](#)
- [Add Records to the Zone Scopes](#)
- [Create the DNS Policies](#)

The following sections provide detailed configuration instructions.

IMPORTANT

The following sections include example Windows PowerShell commands that contain example values for many parameters. Ensure that you replace example values in these commands with values that are appropriate for your deployment before you run these commands.

Create the Zone Scopes

A zone scope is a unique instance of the zone. A DNS zone can have multiple zone scopes, with each zone scope containing its own set of DNS records. The same record can be present in multiple scopes, with different IP addresses or the same IP addresses.

NOTE

By default, a zone scope exists on the DNS zones. This zone scope has the same name as the zone, and legacy DNS operations work on this scope. This default zone scope will host the external version of www.career.contoso.com.

You can use the following example command to partition the zone scope contoso.com to create an internal zone scope. The internal zone scope will be used to keep the internal version of www.career.contoso.com.

```
Add-DnsServerZoneScope -ZoneName "contoso.com" -Name "internal"
```

For more information, see [Add-DnsServerZoneScope](#)

Add Records to the Zone Scopes

The next step is to add the records representing the Web server host into the two zone scopes - internal and default (for external clients).

In the internal zone scope, the record **www.career.contoso.com** is added with the IP address 10.0.0.39, which is a private IP; and in the default zone scope the same record, **www.career.contoso.com**, is added with the IP address 65.55.39.10.

No **-ZoneScope** parameter is provided in the following example commands when the record is being added to the default zone scope. This is similar to adding records to a vanilla zone.

```
Add-DnsServerResourceRecord -ZoneName "contoso.com" -A -Name "www.career" -IPv4Address "65.55.39.10"  
Add-DnsServerResourceRecord -ZoneName "contoso.com" -A -Name "www.career" -IPv4Address "10.0.0.39" -ZoneScope "internal"
```

For more information, see [Add-DnsServerResourceRecord](#).

Create the DNS Policies

After you have identified the server interfaces for the external network and internal network and you have created the zone scopes, you must create DNS policies that connect the internal and external zone scopes.

NOTE

This example uses the server interface as the criteria to differentiate between the internal and external clients. Another method to differentiate between external and internal clients is by using client subnets as a criteria. If you can identify the subnets to which the internal clients belong, you can configure DNS policy to differentiate based on client subnet. For information on how to configure traffic management using client subnet criteria, see [Use DNS Policy for Geo-Location Based Traffic Management with Primary Servers](#).

When the DNS server receives a query on the private interface, the DNS query response is returned from the internal zone scope.

NOTE

No policies are required for mapping the default zone scope.

In the following example command, 10.0.0.56 is the IP address on the private network interface, as shown in the previous illustration.

```
Add-DnsServerQueryResolutionPolicy -Name "SplitBrainZonePolicy" -Action ALLOW -ServerInterface "eq,10.0.0.56" -ZoneScope "internal,1" -ZoneName contoso.com
```

For more information, see [Add-DnsServerQueryResolutionPolicy](#).

Example of DNS Selective Recursion Control

Following is an example of how you can use DNS policy to accomplish the previously described scenario of DNS selective recursion control.

This section contains the following topics.

- [How DNS Selective Recursion Control Works](#)
- [How to Configure DNS Selective Recursion Control](#)

This example uses the same fictional company as in the previous example, Contoso, which maintains a career Web site at www.career.contoso.com.

In the DNS split-brain deployment example, the same DNS server responds to both the external and internal clients and provides them with different answers.

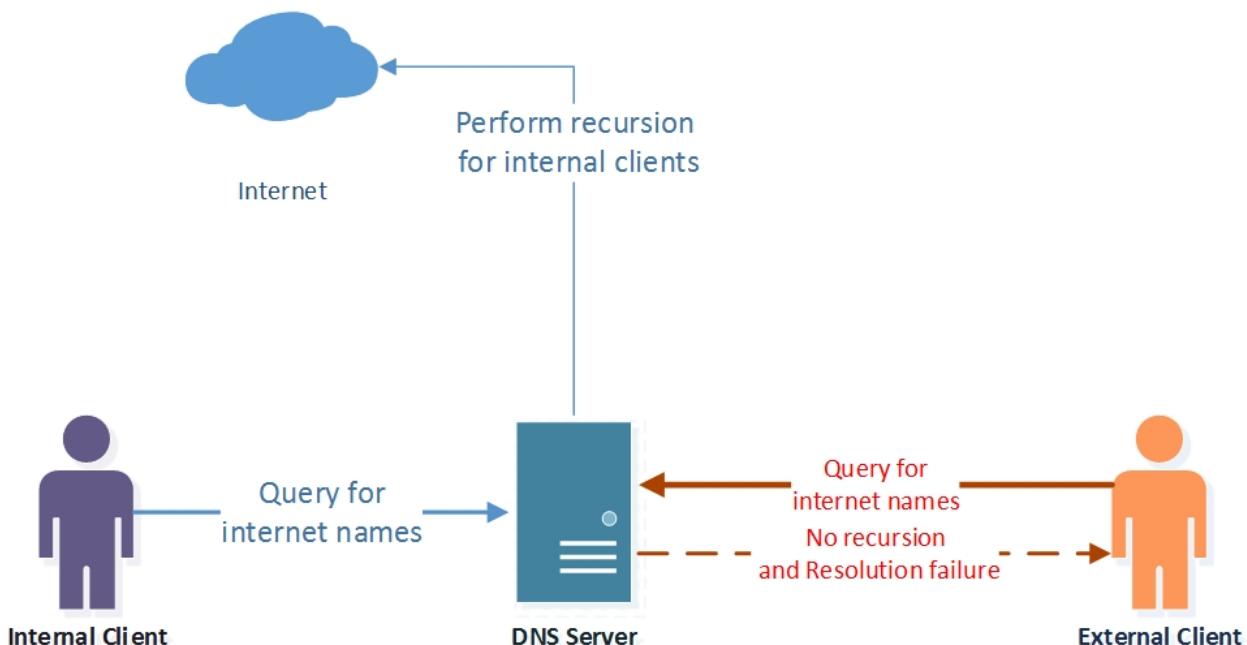
Some DNS deployments might require the same DNS server to perform recursive name resolution for internal clients in addition to acting as the authoritative name server for external clients. This circumstance is called DNS selective recursion control.

In previous versions of Windows Server, enabling recursion meant that it was enabled on the whole DNS server for all zones. Because the DNS server is also listening to external queries, recursion is enabled for both internal and external clients, making the DNS server an open resolver.

A DNS server that is configured as an open resolver might be vulnerable to resource exhaustion and can be abused by malicious clients to create reflection attacks.

Because of this, Contoso DNS administrators do not want the DNS server for contoso.com to perform recursive name resolution for external clients. There is only a need for recursion control for internal clients, while recursion control can be blocked for external clients.

The following illustration depicts this scenario.



How DNS Selective Recursion Control Works

If a query for which the Contoso DNS server is non-authoritative is received, such as for www.microsoft.com, then the name resolution request is evaluated against the policies on the DNS server.

Because these queries do not fall under any zone, the zone level policies (as defined in the split-brain example) are not evaluated.

The DNS server evaluates the recursion policies, and the queries that are received on the private interface match the **SplitBrainRecursionPolicy**. This policy points to a recursion scope where recursion is enabled.

The DNS server then performs recursion to get the answer for www.microsoft.com from the Internet, and caches the response locally.

If the query is received on the external interface, no DNS policies match, and the default recursion setting - which in this case is **Disabled** - is applied.

This prevents the server from acting as an open resolver for external clients, while it is acting as a caching resolver for internal clients.

How to Configure DNS Selective Recursion Control

To configure DNS selective recursion control by using DNS Policy, you must use the following steps.

- [Create DNS Recursion Scopes](#)
- [Create DNS Recursion Policies](#)

Create DNS Recursion Scopes

Recursion scopes are unique instances of a group of settings that control recursion on a DNS server. A recursion scope contains a list of forwarders and specifies whether recursion is enabled. A DNS server can have many recursion scopes.

The legacy recursion setting and list of forwarders are referred to as the default recursion scope. You cannot add or remove the default recursion scope, identified by the name dot (".").

In this example, the default recursion setting is disabled, while a new recursion scope for internal clients is created where recursion is enabled.

```
Set-DnsServerRecursionScope -Name . -EnableRecursion $False  
Add-DnsServerRecursionScope -Name "InternalClients" -EnableRecursion $True
```

For more information, see [Add-DnsServerRecursionScope](#)

Create DNS Recursion Policies

You can create DNS server recursion policies to choose a recursion scope for a set of queries that match specific criteria.

If the DNS server is not authoritative for some queries, DNS server recursion policies allow you to control how to resolve the queries.

In this example, the internal recursion scope with recursion enabled is associated with the private network interface

You can use the following example command to configure DNS recursion policies.

```
Add-DnsServerQueryResolutionPolicy -Name "SplitBrainRecursionPolicy" -Action ALLOW -ApplyOnRecursion -  
RecursionScope "InternalClients" -ServerInterfaceIP "EQ,10.0.0.39"
```

For more information, see [Add-DnsServerQueryResolutionPolicy](#).

Now the DNS server is configured with the required DNS policies for either a split-brain name server or a DNS server with selective recursion control enabled for internal clients.

You can create thousands of DNS policies according to your traffic management requirements, and all new policies are applied dynamically - without restarting the DNS server - on incoming queries.

For more information, see [DNS Policy Scenario Guide](#).

Use DNS Policy for Split-Brain DNS in Active Directory

9/1/2018 • 6 minutes to read • [Edit Online](#)

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

You can use this topic to leverage the traffic management capabilities of DNS policies for split-brain deployments with Active Directory integrated DNS zones in Windows Server 2016.

In Windows Server 2016, DNS policies support is extended to Active Directory integrated DNS zones. Active Directory integration provides multi-master high availability capabilities to the DNS server.

Previously, this scenario required that DNS administrators maintain two different DNS servers, each providing services to each set of users, internal and external. If only a few records inside the zone were split-brained or both instances of the zone (internal and external) were delegated to the same parent domain, this became a management conundrum.

NOTE

- DNS deployments are split-brain when there are two versions of a single zone, one version for internal users on the organization intranet, and one version for external users – who are, typically, users on the Internet.
- The topic [Use DNS Policy for Split-Brain DNS Deployment](#) explains how you can use DNS policies and zone scopes to deploy a split-brain DNS system on a single Windows Server 2016 DNS server.

Example Split-Brain DNS in Active Directory

This example uses one fictional company, Contoso, which maintains a career Web site at www.career.contoso.com.

The site has two versions, one for the internal users where internal job postings are available. This internal site is available at the local IP address 10.0.0.39.

The second version is the public version of the same site, which is available at the public IP address 65.55.39.10.

In the absence of DNS policy, the administrator is required to host these two zones on separate Windows Server DNS servers and manage them separately.

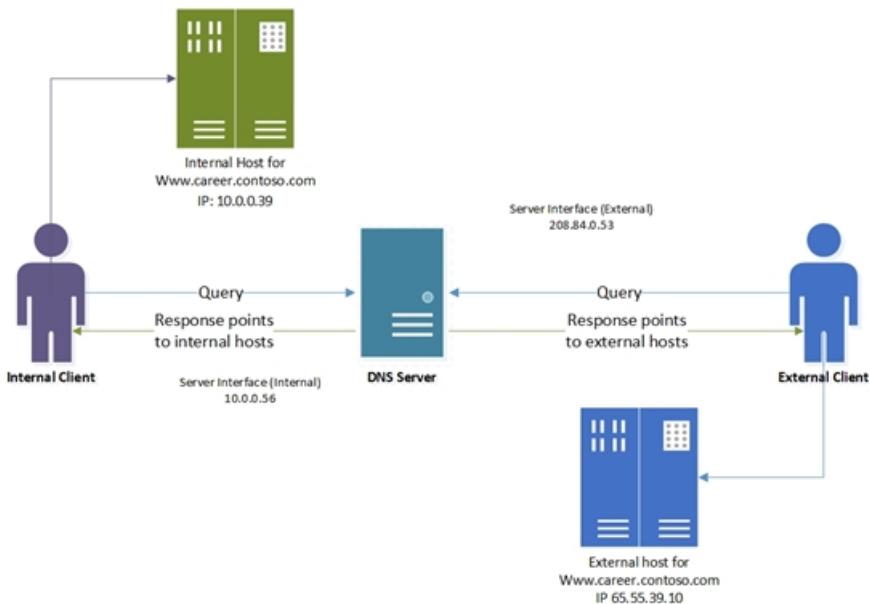
Using DNS policies these zones can now be hosted on the same DNS server.

If the DNS server for contoso.com is Active Directory integrated, and is listening on two network interfaces, the Contoso DNS Administrator can follow the steps in this topic to achieve a split-brain deployment.

The DNS Administrator configures the DNS server interfaces with the following IP addresses.

- The Internet facing network adapter is configured with a public IP address of 208.84.0.53 for external queries.
- The Intranet facing network adapter is configured with a private IP address of 10.0.0.56 for internal queries.

The following illustration depicts this scenario.



How DNS Policy for Split-Brain DNS in Active Directory Works

When the DNS server is configured with the required DNS policies, each name resolution request is evaluated against the policies on the DNS server.

The server Interface is used in this example as the criteria to differentiate between the internal and external clients.

If the server interface upon which the query is received matches any of the policies, the associated zone scope is used to respond to the query.

So, in our example, the DNS queries for `www.career.contoso.com` that are received on the private IP (10.0.0.56) receive a DNS response that contains an internal IP address; and the DNS queries that are received on the public network interface receive a DNS response that contains the public IP address in the default zone scope (this is the same as normal query resolution).

Support for Dynamic DNS (DDNS) updates and scavenging is supported only on the default zone scope. Because the internal clients are serviced by the default zone scope, Contoso DNS Administrators can continue using the existing mechanisms (dynamic DNS or static) to update the records in `contoso.com`. For non-default zone scopes (such as the external scope in this example), DDNS or scavenging support is not available.

High Availability of policies

DNS policies are not Active Directory integrated. Because of this, DNS policies are not replicated to the other DNS servers that are hosting the same Active Directory integrated zone.

DNS policies are stored on the local DNS server. You can easily export DNS policies from one server to another by using the following example Windows PowerShell commands.

```
$policies = Get-DnsServerQueryResolutionPolicy -ZoneName "contoso.com" -ComputerName Server01
$policies | Add-DnsServerQueryResolutionPolicy -ZoneName "contoso.com" -ComputerName Server02
```

For more information, see the following Windows PowerShell reference topics.

- [Get-DnsServerQueryResolutionPolicy](#)
- [Add-DnsServerQueryResolutionPolicy](#)

How to Configure DNS Policy for Split-Brain DNS in Active Directory

To configure DNS Split-Brain Deployment by using DNS Policy, you must use the following sections, which

provide detailed configuration instructions.

Add the Active Directory integrated zone

You can use the following example command to add the Active Directory integrated contoso.com zone to the DNS server.

```
Add-DnsServerPrimaryZone -Name "contoso.com" -ReplicationScope "Domain" -PassThru
```

For more information, see [Add-DnsServerPrimaryZone](#).

Create the Scopes of the Zone

You can use this section to partition the zone contoso.com to create an external zone scope.

A zone scope is a unique instance of the zone. A DNS zone can have multiple zone scopes, with each zone scope containing its own set of DNS records. The same record can be present in multiple scopes, with different IP addresses or the same IP addresses.

Because you are adding this new zone scope in an Active Directory integrated zone, the zone scope and the records inside it will replicate via Active Directory to other replica servers in the domain.

By default, a zone scope exists in every DNS zone. This zone scope has the same name as the zone, and legacy DNS operations work on this scope. This default zone scope will host the internal version of www.career.contoso.com.

You can use the following example command to create the zone scope on the DNS server.

```
Add-DnsServerZoneScope -ZoneName "contoso.com" -Name "external"
```

For more information, see [Add-DnsServerZoneScope](#).

Add Records to the Zone Scopes

The next step is to add the records representing the web server host into the two zone scopes- external and default (for internal clients).

In the default internal zone scope, the record www.career.contoso.com is added with IP address 10.0.0.39, which is a private IP address; and in the external zone scope, the same record (www.career.contoso.com) is added with the public IP address 65.55.39.10.

The records (both in the default internal zone scope and the external zone scope) will automatically replicate across the domain with their respective zone scopes.

You can use the following example command to add records to the zone scopes on the DNS server.

```
Add-DnsServerResourceRecord -ZoneName "contoso.com" -A -Name "www.career" -IPv4Address "65.55.39.10" -ZoneScope "external"  
Add-DnsServerResourceRecord -ZoneName "contoso.com" -A -Name "www.career" -IPv4Address "10.0.0.39"
```

NOTE

The **-ZoneScope** parameter is not included when the record is added to the default zone scope. This action is same as adding records to a normal zone.

For more information, see [Add-DnsServerResourceRecord](#).

Create the DNS Policies

After you have identified the server interfaces for the external network and internal network and you have created the zone scopes, you must create DNS policies that connect the internal and external zone scopes.

NOTE

This example uses the server interface (the `-ServerInterface` parameter in the example command below) as the criteria to differentiate between the internal and external clients. Another method to differentiate between external and internal clients is by using client subnets as a criteria. If you can identify the subnets to which the internal clients belong, you can configure DNS policy to differentiate based on client subnet. For information on how to configure traffic management using client subnet criteria, see [Use DNS Policy for Geo-Location Based Traffic Management with Primary Servers](#).

After you configure policies, when a DNS query is received on the public interface, the answer is returned from the external scope of the zone.

NOTE

No policies are required for mapping the default internal zone scope.

```
Add-DnsServerQueryResolutionPolicy -Name "SplitBrainZonePolicy" -Action ALLOW -ServerInterface  
"eq,208.84.0.53" -ZoneScope "external,1" -ZoneName contoso.com
```

NOTE

208.84.0.53 is the IP address on the public network interface.

For more information, see [Add-DnsServerQueryResolutionPolicy](#).

Now the DNS server is configured with the required DNS policies for a split-brain name server with an Active Directory integrated DNS zone.

You can create thousands of DNS policies according to your traffic management requirements, and all new policies are applied dynamically - without restarting the DNS server - on incoming queries.

Use DNS Policy for Applying Filters on DNS Queries

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to configure DNS policy in Windows Server® 2016 to create query filters that are based on criteria that you supply.

Query filters in DNS policy allow you to configure the DNS server to respond in a custom manner based on the DNS query and DNS client that sends the DNS query.

For example, you can configure DNS policy with query filter Block List that blocks DNS queries from known malicious domains, which prevents DNS from responding to queries from these domains. Because no response is sent from the DNS server, the malicious domain member's DNS query times out.

Another example is to create a query filter Allow List that allows only a specific set of clients to resolve certain names.

Query filter criteria

You can create query filters with any logical combination (AND/OR/NOT) of the following criteria.

NAME	DESCRIPTION
Client Subnet	Name of a predefined client subnet. Used to verify the subnet from which the query was sent.
Transport Protocol	Transport protocol used in the query. Possible values are UDP and TCP.
Internet Protocol	Network protocol used in the query. Possible values are IPv4 and IPv6.
Server Interface IP address	IP address of the network interface of the DNS server that received the DNS request
FQDN	Fully Qualified Domain Name of record in the query, with the possibility of using a wild card.
Query Type	Type of record being queried (A, SRV, TXT, etc.)
Time of Day	Time of day the query is received.

The following examples show you how to create filters for DNS policy that either block or allow DNS name resolution queries.

NOTE

The example commands in this topic use the Windows PowerShell command **Add-DnsServerQueryResolutionPolicy**. For more information, see [Add-DnsServerQueryResolutionPolicy](#).

Block queries from a domain

In some circumstances you might want to block DNS name resolution for domains that you have identified as malicious, or for domains that do not comply with the usage guidelines of your organization. You can accomplish blocking queries for domains by using DNS policy.

The policy that you configure in this example is not created on any particular zone – instead you create a Server Level Policy that is applied to all zones configured on the DNS server. Server Level Policies are the first to be evaluated and thus first to be matched when a query is received by the DNS server.

The following example command configures a Server Level Policy to block any queries with the domain **suffix contosomalicious.com**.

```
Add-DnsServerQueryResolutionPolicy -Name "BlockListPolicy" -Action IGNORE -FQDN "EQ,*.contosomalicious.com" -PassThru
```

NOTE

When you configure the **Action** parameter with the value **IGNORE**, the DNS server is configured to drop queries with no response at all. This causes the DNS client in the malicious domain to time out.

Block queries from a subnet

With this example, you can block queries from a subnet if it is found to be infected by some malware and is trying to contact malicious sites using your DNS server.

```
` Add-DnsServerClientSubnet -Name "MaliciousSubnet06" -IPv4Subnet 172.0.33.0/24 -PassThru
```

```
Add-DnsServerQueryResolutionPolicy -Name "BlockListPolicyMalicious06" -Action IGNORE -ClientSubnet "EQ,MaliciousSubnet06" -PassThru`
```

The following example demonstrates how you can use the subnet criteria in combination with the FQDN criteria to block queries for certain malicious domains from infected subnets.

```
Add-DnsServerQueryResolutionPolicy -Name "BlockListPolicyMalicious06" -Action IGNORE -ClientSubnet "EQ,MaliciousSubnet06" -FQDN "EQ,*.contosomalicious.com" -PassThru
```

Block a type of query

You might need to block name resolution for certain types of queries on your servers. For example, you can block the 'ANY' query, which can be used maliciously to create amplification attacks.

```
Add-DnsServerQueryResolutionPolicy -Name "BlockListPolicyQType" -Action IGNORE -QType "EQ,ANY" -PassThru
```

Allow queries only from a domain

You can not only use DNS policy to block queries, you can use them to automatically approve queries from specific domains or subnets. When you configure Allow Lists, the DNS server only processes queries from allowed domains, while blocking all other queries from other domains.

The following example command allows only computers and devices in the contoso.com and child domains to query the DNS server.

```
Add-DnsServerQueryResolutionPolicy -Name "AllowListPolicyDomain" -Action IGNORE -FQDN "NE,*.contoso.com" -PassThru
```

Allow queries only from a subnet

You can also create Allow Lists for IP subnets, so that all queries not originating from these subnets are ignored.

```
Add-DnsServerClientSubnet -Name "AllowedSubnet06" -IPv4Subnet 172.0.33.0/24 -PassThru  
Add-DnsServerQueryResolutionPolicy -Name "AllowListPolicySubnet" -Action IGNORE -ClientSubnet "NE,  
AllowedSubnet06" -PassThru
```

Allow only certain QTypes

You can apply Allow Lists to QTYPEs.

For example, if you have external customers querying DNS server interface 164.8.1.1, only certain QTYPEs are allowed to be queried, while there are other QTYPEs like SRV or TXT records which are used by internal servers for name resolution or for monitoring purposes.

```
Add-DnsServerQueryResolutionPolicy -Name "AllowListQType" -Action IGNORE -QType "NE,A,AAAA,MX,NS,SOA" -  
ServerInterface "EQ,164.8.1.1" -PassThru
```

You can create thousands of DNS policies according to your traffic management requirements, and all new policies are applied dynamically - without restarting the DNS server - on incoming queries.

Use DNS Policy for Application Load Balancing

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to configure DNS policy to perform application load balancing.

Previous versions of Windows Server DNS only provided load balancing by using round robin responses; but with DNS in Windows Server 2016, you can configure DNS policy for application load balancing.

When you have deployed multiple instances of an application, you can use DNS policy to balance the traffic load between the different application instances, thereby dynamically allocating the traffic load for the application.

Example of Application Load Balancing

Following is an example of how you can use DNS policy for application load balancing.

This example uses one fictional company - Contoso Gift Services - which provides online gifting services, and which has a Web site named **contosogiftservices.com**.

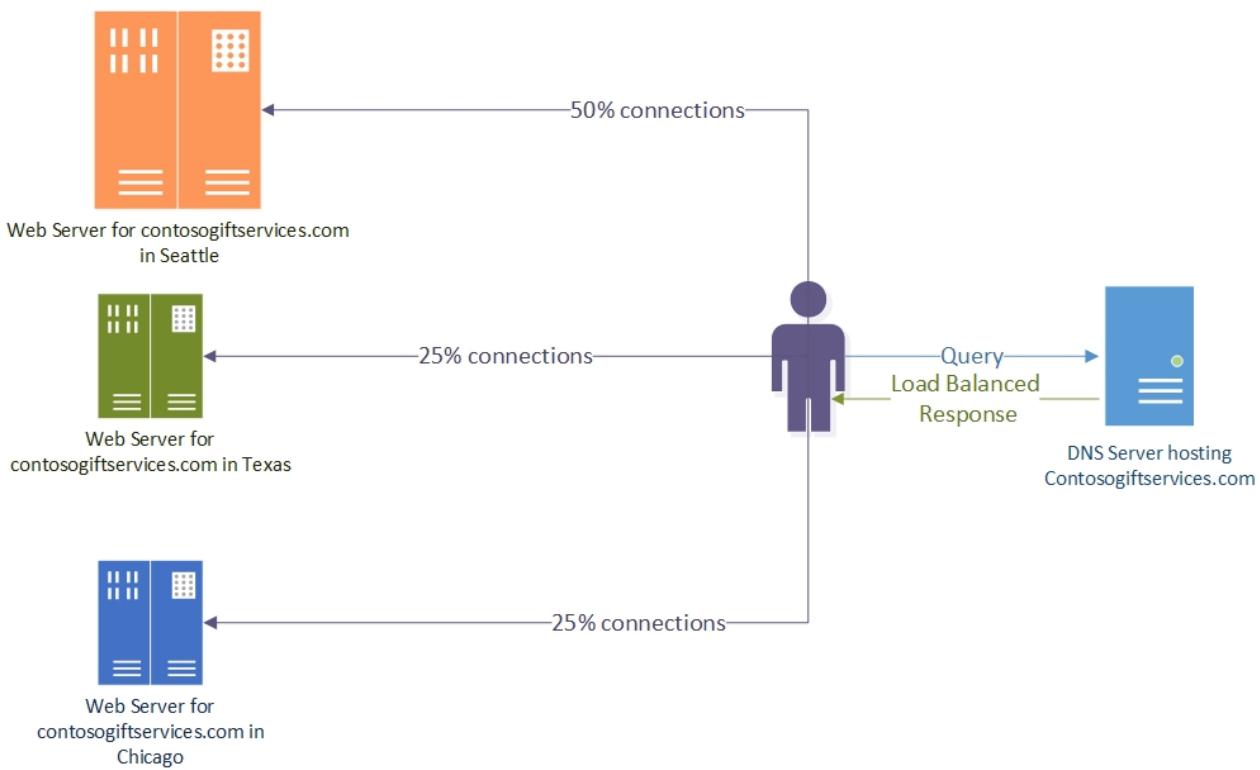
The contosogiftservices.com website is hosted in multiple datacenters that each have different IP addresses.

In North America, which is the primary market for Contoso Gift Services, the Web site is hosted in three datacenters: Chicago, IL, Dallas, TX and Seattle, WA.

The Seattle Web server has the best hardware configuration and can handle twice as much load as the other two sites. Contoso Gift Services wants application traffic directed in the following manner.

- Because the Seattle Web server includes more resources, half of the application's clients are directed to this server
- One quarter of the application's clients are directed to the Dallas, TX datacenter
- One quarter of the application's clients are directed to the Chicago, IL, datacenter

The following illustration depicts this scenario.



How Application Load Balancing Works

After you have configured the DNS server with DNS policy for application load balancing using this example scenario, the DNS server responds 50% of the time with the Seattle Web server address, 25% of the time with the Dallas Web server address, and 25% of the time with the Chicago Web server address.

Thus for every four queries the DNS server receives, it responds with two responses for Seattle and one each for Dallas and Chicago.

One possible issue with load balancing with DNS policy is the caching of DNS records by the DNS client and the resolver/LDNS, which can interfere with load balancing because the client or resolver do not send a query to the DNS server.

You can mitigate the effect of this behavior by using a low Time-to-Live (TTL) value for the DNS records that should be load balanced.

How to Configure Application Load Balancing

The following sections show you how to configure DNS policy for application load balancing.

Create the Zone Scopes

You must first create the scopes of the zone contosogiftservices.com for the datacenters where they are hosted.

A zone scope is a unique instance of the zone. A DNS zone can have multiple zone scopes, with each zone scope containing its own set of DNS records. The same record can be present in multiple scopes, with different IP addresses or the same IP addresses.

NOTE

By default, a zone scope exists on the DNS zones. This zone scope has the same name as the zone, and legacy DNS operations work on this scope.

You can use the following Windows PowerShell commands to create zone scopes.

```
Add-DnsServerZoneScope -ZoneName "contosogiftservices.com" -Name "SeattleZoneScope"  
Add-DnsServerZoneScope -ZoneName "contosogiftservices.com" -Name "DallasZoneScope"  
Add-DnsServerZoneScope -ZoneName "contosogiftservices.com" -Name "ChicagoZoneScope"
```

For more information, see [Add-DnsServerZoneScope](#)

Add Records to the Zone Scopes

Now you must add the records representing the web server host into the zone scopes.

In **SeattleZoneScope**, you can add the record www.contosogiftservices.com with IP address 192.0.0.1, which is located in the Seattle datacenter.

In **ChicagoZoneScope**, you can add the same record (www.contosogiftservices.com) with IP address 182.0.0.1 in the Chicago datacenter.

Similarly in **DallasZoneScope**, you can add a record (www.contosogiftservices.com) with IP address 162.0.0.1 in the Chicago datacenter.

You can use the following Windows PowerShell commands to add records to the zone scopes.

```
Add-DnsServerResourceRecord -ZoneName "contosogiftservices.com" -A -Name "www" -IPv4Address "192.0.0.1" -  
ZoneScope "SeattleZoneScope"  
  
Add-DnsServerResourceRecord -ZoneName "contosogiftservices.com" -A -Name "www" -IPv4Address "182.0.0.1" -  
ZoneScope "ChicagoZoneScope"  
  
Add-DnsServerResourceRecord -ZoneName "contosogiftservices.com" -A -Name "www" -IPv4Address "162.0.0.1" -  
ZoneScope "DallasZoneScope"
```

For more information, see [Add-DnsServerResourceRecord](#).

Create the DNS Policies

After you have created the partitions (zone scopes) and you have added records, you must create DNS policies that distribute the incoming queries across these scopes so that 50% of queries for contosogiftservices.com are responded to with the IP address for the Web server in the Seattle datacenter and the rest are equally distributed between the Chicago and Dallas datacenters.

You can use the following Windows PowerShell commands to create a DNS policy that balances application traffic across these three datacenters.

NOTE

In the example command below, the expression –ZoneScope "SeattleZoneScope,2; ChicagoZoneScope,1; DallasZoneScope,1" configures the DNS server with an array that includes the parameter combination <ZoneScope>, <weight>.

```
Add-DnsServerQueryResolutionPolicy -Name "AmericaPolicy" -Action ALLOW - -ZoneScope  
"SeattleZoneScope,2;ChicagoZoneScope,1;DallasZoneScope,1" -ZoneName "contosogiftservices.com"
```

For more information, see [Add-DnsServerQueryResolutionPolicy](#).

You have now successfully created a DNS policy that provides application load balancing across Web servers in three different datacenters.

You can create thousands of DNS policies according to your traffic management requirements, and all new policies are applied dynamically - without restarting the DNS server - on incoming queries.

Use DNS Policy for Application Load Balancing With Geo-Location Awareness

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to configure DNS policy to load balance an application with geo-location awareness.

The previous topic in this guide, [Use DNS Policy for Application Load Balancing](#), uses an example of a fictional company - Contoso Gift Services - which provides online gifting services, and which has a Web site named contosogiftservices.com. Contoso Gift Services load balances their online Web application between servers in North American datacenters located in Seattle, WA, Chicago, IL, and Dallas, TX.

NOTE

It is recommended that you familiarize yourself with the topic [Use DNS Policy for Application Load Balancing](#) before performing the instructions in this scenario.

This topic uses the same fictional company and network infrastructure as a basis for a new example deployment that includes geo-location awareness.

In this example, Contoso Gift Services is successfully expanding their presence across the globe.

Similar to North America, the company now has web servers hosted in European datacenters.

Contoso Gift Services DNS Administrators want to configure application load balancing for European datacenters in a similar manner to the DNS policy implementation in the United States, with application traffic distributed among Web servers that are located in Dublin, Ireland, Amsterdam, Holland, and elsewhere.

DNS Administrators also want all queries from other locations in the world distributed equally between all of their datacenters.

In the next sections you can learn how to achieve similar goals to those of the Contoso DNS Administrators on your own network.

How to Configure Application Load Balancing with Geo-Location Awareness

The following sections show you how to configure DNS policy for application load balancing with geo-location awareness.

IMPORTANT

The following sections include example Windows PowerShell commands that contain example values for many parameters. Ensure that you replace example values in these commands with values that are appropriate for your deployment before you run these commands.

Create the DNS Client Subnets

You must first identify the subnets or IP address space of the North America and Europe regions.

You can obtain this information from Geo-IP maps. Based on these Geo-IP distributions, you must create the DNS Client Subnets.

A DNS Client Subnet is a logical grouping of IPv4 or IPv6 subnets from which queries are sent to a DNS server.

You can use the following Windows PowerShell commands to create DNS Client Subnets.

```
Add-DnsServerClientSubnet -Name "AmericaSubnet" -IPv4Subnet 192.0.0.0/24,182.0.0.0/24  
Add-DnsServerClientSubnet -Name "EuropeSubnet" -IPv4Subnet 141.1.0.0/24,151.1.0.0/24
```

For more information, see [Add-DnsServerClientSubnet](#).

Create the Zone Scopes

After the client subnets are in place, you must partition the zone contosogiftservices.com into different zone scopes, each for a datacenter.

A zone scope is a unique instance of the zone. A DNS zone can have multiple zone scopes, with each zone scope containing its own set of DNS records. The same record can be present in multiple scopes, with different IP addresses or the same IP addresses.

NOTE

By default, a zone scope exists on the DNS zones. This zone scope has the same name as the zone, and legacy DNS operations work on this scope.

The previous scenario on application load balancing demonstrates how to configure three zone scopes for datacenters in North America.

With the commands below, you can create two more zone scopes, one each for the Dublin and Amsterdam datacenters.

You can add these zone scopes without any changes to the three existing North America zone scopes in the same zone. In addition, after you create these zone scopes, you do not need to restart your DNS server.

You can use the following Windows PowerShell commands to create zone scopes.

```
Add-DnsServerZoneScope -ZoneName "contosogiftservices.com" -Name "DublinZoneScope"  
Add-DnsServerZoneScope -ZoneName "contosogiftservices.com" -Name "AmsterdamZoneScope"
```

For more information, see [Add-DnsServerZoneScope](#)

Add Records to the Zone Scopes

Now you must add the records representing the web server host into the zone scopes.

The records for the America datacenters were added in the previous scenario. You can use the following Windows PowerShell commands to add records to the zone scopes for European datacenters.

```
Add-DnsServerResourceRecord -ZoneName "contosogiftservices.com" -A -Name "www" -IPv4Address "151.1.0.1" -  
ZoneScope "DublinZoneScope"  
Add-DnsServerResourceRecord -ZoneName "contosogiftservices.com" -A -Name "www" -IPv4Address "141.1.0.1" -  
ZoneScope "AmsterdamZoneScope"
```

For more information, see [Add-DnsServerResourceRecord](#).

Create the DNS Policies

After you have created the partitions (zone scopes) and you have added records, you must create DNS policies

that distribute the incoming queries across these scopes.

For this example, query distribution across application servers in different datacenters meets the following criteria.

1. When the DNS query is received from a source in a North American client subnet, 50% of the DNS responses point to the Seattle data center, 25% of responses point to the Chicago datacenter, and the remaining 25% of responses point to the Dallas datacenter.
2. When the DNS query is received from a source in a European client subnet, 50% of the DNS responses point to the Dublin datacenter, and 50% of the DNS responses point to the Amsterdam datacenter.
3. When the query comes from anywhere else in the world, the DNS responses are distributed across all five datacenters.

You can use the following Windows PowerShell commands to implement these DNS policies.

```
Add-DnsServerQueryResolutionPolicy -Name "AmericaLBPolicy" -Action ALLOW -ClientSubnet "eq,AmericaSubnet" -ZoneScope "SeattleZoneScope,2;ChicagoZoneScope,1; TexasZoneScope,1" -ZoneName "contosogiftservices.com" -ProcessingOrder 1

Add-DnsServerQueryResolutionPolicy -Name "EuropeLBPolicy" -Action ALLOW -ClientSubnet "eq,EuropeSubnet" -ZoneScope "DublinZoneScope,1;AmsterdamZoneScope,1" -ZoneName "contosogiftservices.com" -ProcessingOrder 2

Add-DnsServerQueryResolutionPolicy -Name "WorldWidePolicy" -Action ALLOW -FQDN "eq,*.contoso.com" -ZoneScope "SeattleZoneScope,1;ChicagoZoneScope,1; TexasZoneScope,1;DublinZoneScope,1;AmsterdamZoneScope,1" -ZoneName "contosogiftservices.com" -ProcessingOrder 3
```

For more information, see [Add-DnsServerQueryResolutionPolicy](#).

You have now successfully created a DNS policy that provides application load balancing across Web servers that are located in five different datacenters on multiple continents.

You can create thousands of DNS policies according to your traffic management requirements, and all new policies are applied dynamically - without restarting the DNS server - on incoming queries.

Dynamic Host Configuration Protocol (DHCP)

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic for a brief overview of DHCP in Windows Server 2016.

NOTE

In addition to this topic, the following DHCP documentation is available.

- [What's New in DHCP](#)
- [Deploy DHCP Using Windows PowerShell](#)

Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway. RFCs 2131 and 2132 define DHCP as an Internet Engineering Task Force (IETF) standard based on Bootstrap Protocol (BOOTP), a protocol with which DHCP shares many implementation details. DHCP allows hosts to obtain required TCP/IP configuration information from a DHCP server.

Windows Server 2016 includes DHCP Server, which is an optional networking server role that you can deploy on your network to lease IP addresses and other information to DHCP clients. All Windows-based client operating systems include the DHCP client as part of TCP/IP, and DHCP client is enabled by default.

Why use DHCP?

Every device on a TCP/IP-based network must have a unique unicast IP address to access the network and its resources. Without DHCP, IP addresses for new computers or computers that are moved from one subnet to another must be configured manually; IP addresses for computers that are removed from the network must be manually reclaimed.

With DHCP, this entire process is automated and managed centrally. The DHCP server maintains a pool of IP addresses and leases an address to any DHCP-enabled client when it starts up on the network. Because the IP addresses are dynamic (leased) rather than static (permanently assigned), addresses no longer in use are automatically returned to the pool for reallocation.

The network administrator establishes DHCP servers that maintain TCP/IP configuration information and provide address configuration to DHCP-enabled clients in the form of a lease offer. The DHCP server stores the configuration information in a database that includes:

- Valid TCP/IP configuration parameters for all clients on the network.
- Valid IP addresses, maintained in a pool for assignment to clients, as well as excluded addresses.
- Reserved IP addresses associated with particular DHCP clients. This allows consistent assignment of a single IP address to a single DHCP client.
- The lease duration, or the length of time for which the IP address can be used before a lease renewal is required.

A DHCP-enabled client, upon accepting a lease offer, receives:

- A valid IP address for the subnet to which it is connecting.

- Requested DHCP options, which are additional parameters that a DHCP server is configured to assign to clients. Some examples of DHCP options are Router (default gateway), DNS Servers, and DNS Domain Name.

Benefits of DHCP

DHCP provides the following benefits.

- **Reliable IP address configuration.** DHCP minimizes configuration errors caused by manual IP address configuration, such as typographical errors, or address conflicts caused by the assignment of an IP address to more than one computer at the same time.
- **Reduced network administration.** DHCP includes the following features to reduce network administration:
 - Centralized and automated TCP/IP configuration.
 - The ability to define TCP/IP configurations from a central location.
 - The ability to assign a full range of additional TCP/IP configuration values by means of DHCP options.
 - The efficient handling of IP address changes for clients that must be updated frequently, such as those for portable devices that move to different locations on a wireless network.
 - The forwarding of initial DHCP messages by using a DHCP relay agent, which eliminates the need for a DHCP server on every subnet.

What's New in DHCP

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

This topic describes the Dynamic Host Configuration Protocol (DHCP) functionality that is new or changed in Windows Server 2016.

DHCP is an Internet Engineering Task Force (IETF) standard that is designed to reduce the administrative burden and complexity of configuring hosts on a TCP/IP-based network, such as a private intranet. By using the DHCP Server service, the process of configuring TCP/IP on DHCP clients is automatic.

The following sections provide information about new features and changes in functionality for DHCP.

DHCP Subnet Selection Options

DHCP now supports options 118 and 82 (sub-option 5). You can use these options to allow DHCP proxy clients and relay agents to request an IP address for a specific subnet, and from a specific IP address range and scope.

If you are using a DHCP relay agent that is configured with DHCP option 82, sub-option 5, the relay agent can request an IP address lease for DHCP clients from a specific IP address range.

For more information, see [DHCP Subnet Selection Options](#).

New Logging Events for DNS Registration Failures by the DHCP Server

DHCP now includes logging events for circumstances in which DHCP server DNS record registrations fail on the DNS server.

For more information, see [DHCP Logging Events for DNS Record Registrations](#).

DHCP NAP Is Not Supported in Windows Server 2016

Network Access Protection (NAP) is deprecated in Windows Server 2012 R2, and in Windows Server 2016 the DHCP Server role no longer supports NAP. For more information, see [Features Removed or Deprecated in Windows Server 2012 R2](#).

NAP support was introduced to the DHCP Server role with Windows Server 2008, and is supported in Windows client and server operating systems prior to Windows 10 and Windows Server 2016. The following table summarizes support for NAP in Windows Server.

OPERATING SYSTEM	NAP SUPPORT
Windows Server 2008	Supported
Windows Server 2008 R2	Supported
Windows Server 2012	Supported
Windows Server 2012 R2	Supported

OPERATING SYSTEM	NAP SUPPORT
Windows Server 2016	Not supported

In a NAP deployment, a DHCP server running an operating system that supports NAP can function as a NAP enforcement point for the NAP DHCP enforcement method. For more information about DHCP in NAP, see [Checklist: Implementing a DHCP Enforcement Design](#).

In Windows Server 2016, DHCP servers do not enforce NAP policies, and DHCP scopes cannot be NAP-enabled. DHCP client computers that are also NAP clients send a statement of health (SoH) with the DHCP request. If the DHCP server is running Windows Server 2016, these requests are processed as if no SoH is present. The DHCP server grants a normal DHCP lease to the client.

If servers that are running Windows Server 2016 are RADIUS proxies that forward authentication requests to a Network Policy Server (NPS) that supports NAP, these NAP clients are evaluated by NPS as non NAP-capable, and NAP processing fails.

See also

- [Dynamic Host Configuration Protocol \(DHCP\)](#)

DHCP Subnet Selection Options

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic for information about new DHCP subnet selection options.

DHCP now supports option 82 (sub-option 5). You can use these options to allow DHCP proxy clients and relay agents to request an IP address for a specific subnet, and from a specific IP address range and scope. For more details, see [Option 82 Sub Option 5: RFC 3527 Link Selection sub-option for the Relay Agent Information Option for DHCPv4](#).

If you are using a DHCP relay agent that is configured with DHCP option 82, sub-option 5, the relay agent can request an IP address lease for DHCP clients from a specific IP address range.

Option 82 Sub Option 5: Link Selection Sub Option

The Relay Agent Link Selection sub-option allows a DHCP Relay Agent to specify an IP subnet from which the DHCP server should assign IP addresses and options.

Typically, DHCP relay agents rely on the Gateway IP Address (GIADDR) field to communicate with DHCP servers. However, GIADDR is limited by its two operational functions:

1. To inform the DHCP server about the subnet upon which the DHCP client that is requesting the IP address lease resides.
2. To inform the DHCP server of the IP address to use to communicate with the relay agent.

In some cases, the IP address that the relay agent uses to communicate with the DHCP server might be different than the IP address range from which the DHCP client IP address needs to be allocated.

The Link Selection Sub option of option 82 is useful in this situation, allowing the relay agent to explicitly state the subnet from which it wants the IP address allocated in the form of DHCP v4 option 82 sub option 5.

NOTE

All relay agent IP addresses (GIADDR) must be part of an active DHCP scope IP address range. Any GIADDR outside of the DHCP scope IP address ranges is considered a rogue relay and Windows DHCP Server will not acknowledge DHCP client requests from those relay agents.

A special scope can be created to "authorize" relay agents. Create a scope with the GIADDR (or multiple if the GIADDR's are sequential IP addresses), exclude the GIADDR address(es) from distribution, and then activate the scope. This will authorize the relay agents while preventing the GIADDR addresses from being assigned.

Use case scenario

In this scenario, an organization network includes both a DHCP server and a Wireless Access Point (AP) for the guest users. Guests client IP addresses are assigned from the organization DHCP server - however, due to firewall policy restrictions, the DHCP server cannot access the guest wireless network or wireless clients with broadcast messages.

To resolve this restriction, the AP is configured with the Link Selection Sub Option 5 to specify the subnet from which it wants the IP address allocated for guest clients, while in the GIADDR also specifying the IP address of the internal interface that leads to the corporate network.

DHCP Logging Events for DNS Registrations

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

DHCP server event logs now provide detailed information about DNS registration failures.

NOTE

In many cases, the reason for DNS record registration failures by DHCP servers is that a DNS Reverse-Lookup Zone is either configured incorrectly or not configured at all.

The following new DHCP events assist you to easily identify when DNS registrations are failing because of a misconfigured or missing DNS Reverse-Lookup Zone.

ID	EVENT	VALUE
20317	DHCPv4.ForwardRecordDNSFailure	Forward record registration for IPv4 address %1 and FQDN %2 failed with error %3. This is likely to be because the forward lookup zone for this record does not exist on the DNS server.
20318	DHCPv4.ForwardRecordDNSTimeout	Forward record registration for IPv4 address %1 and FQDN %2 failed with error %3.
20319	DHCPv4.PTRRecordDNSFailure	PTR record registration for IPv4 address %1 and FQDN %2 failed with error %3. This is likely to be because the reverse lookup zone for this record does not exist on the DNS server.
20320	DHCPv4.PTRRecordDNSTimeout	PTR record registration for IPv4 address %1 and FQDN %2 failed with error %3.
20321	DHCPv6.ForwardRecordDNSFailure	Forward record registration for IPv6 address %1 and FQDN %2 failed with error %3. This is likely to be because the forward lookup zone for this record does not exist on the DNS server.
20322	DHCPv6.ForwardRecordDNSTimeout	Forward record registration for IPv6 address %1 and FQDN %2 failed with error %3.
20323	DHCPv6.PTRRecordDNSFailure	PTR record registration for IPv6 address %1 and FQDN %2 failed with error %3. This is likely to be because the reverse lookup zone for this record does not exist on the DNS server.

ID	EVENT	VALUE
20324	DHCPv6.PTRRecordDNSTimeout	PTR record registration for IPv6 address %1 and FQDN %2 failed with error %3.
20325	DHCPv4.ForwardRecordDNSError	PTR record registration for IPv4 address %1 and FQDN %2 failed with error %3 (%4).
20326	DHCPv6.ForwardRecordDNSError	Forward record registration for IPv6 address %1 and FQDN %2 failed with error %3 (%4)
20327	DHCPv6.PTRRecordDNSError	PTR record registration for IPv6 address %1 and FQDN %2 failed with error %3 (%4).

Deploy DHCP Using Windows PowerShell

9/1/2018 • 22 minutes to read • [Edit Online](#)

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

This guide provides instructions on how to use Windows PowerShell to deploy an Internet Protocol (IP) version 4 Dynamic Host Configuration Protocol (DHCP) server that automatically assigns IP addresses and DHCP options to IPv4 DHCP clients that are connected to one or more subnets on your network.

NOTE

To download this document in Word format from TechNet Gallery, see [Deploy DHCP Using Windows PowerShell in Windows Server 2016](#).

Using DHCP servers to assign IP addresses saves administrative overhead because you do not need to manually configure the TCP/IP v4 settings for every network adapter in every computer on your network. With DHCP, TCP/IP v4 configuration is performed automatically when a computer or other DHCP client is connected to your network.

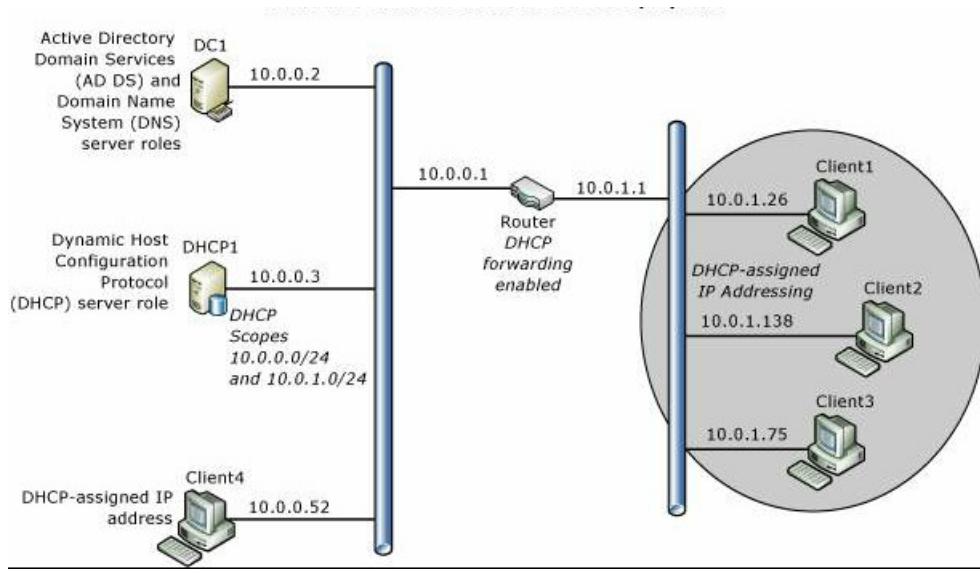
You can deploy your DHCP server in a workgroup as a standalone server, or as part of an Active Directory domain.

This guide contains the following sections.

- [DHCP Deployment Overview](#)
- [Technology Overviews](#)
- [Plan DHCP Deployment](#)
- [Using This Guide in a Test Lab](#)
- [Deploy DHCP](#)
- [Verify Server Functionality](#)
- [Windows PowerShell Commands for DHCP](#)
- [List of Windows PowerShell Commands in this guide](#)

DHCP Deployment Overview

The following illustration depicts the scenario that you can deploy by using this guide. The scenario includes one DHCP server in an Active Directory domain. The server is configured to provide IP addresses to DHCP clients on two different subnets. The subnets are separated by a router that has DHCP Forwarding enabled.



Technology Overviews

The following sections provide brief overviews of DHCP and TCP/IP.

DHCP overview

DHCP is an IP standard for simplifying the management of host IP configuration. The DHCP standard provides for the use of DHCP servers as a way to manage dynamic allocation of IP addresses and other related configuration details for DHCP-enabled clients on your network.

DHCP allows you to use a DHCP server to dynamically assign an IP address to a computer or other device, such as a printer, on your local network, rather than manually configuring every device with a static IP address.

Every computer on a TCP/IP network must have a unique IP address, because the IP address and its related subnet mask identify both the host computer and the subnet to which the computer is attached. By using DHCP, you can ensure that all computers that are configured as DHCP clients receive an IP address that is appropriate for their network location and subnet, and by using DHCP options, such as default gateway and DNS servers, you can automatically provide DHCP clients with the information that they need to function correctly on your network.

For TCP/IP-based networks, DHCP reduces the complexity and amount of administrative work involved in configuring computers.

TCP/IP overview

By default, all versions of Windows Server and Windows Client operating systems have TCP/IP settings for IP version 4 network connections configured to automatically obtain an IP address and other information, called DHCP options, from a DHCP server. Because of this, you do not need to configure TCP/IP settings manually unless the computer is a server computer or other device that requires a manually configured, static IP address.

For example, it is recommended that you manually configure the IP address of the DHCP server, and the IP addresses of DNS servers and domain controllers that are running Active Directory Domain Services (AD DS).

TCP/IP in Windows Server 2016 is the following:

- Networking software based on industry-standard networking protocols.
- A routable enterprise networking protocol that supports the connection of your Windows-based computer to both local area network (LAN) and wide area network (WAN) environments.
- Core technologies and utilities for connecting your Windows-based computer with dissimilar systems for the purpose of sharing information.
- A foundation for gaining access to global Internet services, such as Web and File Transfer Protocol (FTP)

servers.

- A robust, scalable, cross-platform, client/server framework.

TCP/IP provides basic TCP/IP utilities that enable Windows-based computers to connect and share information with other Microsoft and non-Microsoft systems, including:

- Windows Server 2016
- Windows 10
- Windows Server 2012 R2
- Windows 8.1
- Windows Server 2012
- Windows 8
- Windows Server 2008 R2
- Windows 7
- Windows Server 2008
- Windows Vista
- Internet hosts
- Apple Macintosh systems
- IBM mainframes
- UNIX and Linux systems
- Open VMS systems
- Network-ready printers
- Tablets and cellular telephones with wired Ethernet or wireless 802.11 technology enabled

Plan DHCP Deployment

Following are key planning steps before installing the DHCP server role.

Planning DHCP servers and DHCP forwarding

Because DHCP messages are broadcast messages, they are not forwarded between subnets by routers. If you have multiple subnets and want to provide DHCP service for each subnet, you must do one of the following:

- Install a DHCP server on each subnet
- Configure routers to forward DHCP broadcast messages across subnets and configure multiple scopes on the DHCP server, one scope per subnet.

In most cases, configuring routers to forward DHCP broadcast messages is more cost effective than deploying a DHCP server on each physical segment of the network.

Planning IP address ranges

Each subnet must have its own unique IP address range. These ranges are represented on a DHCP server with scopes.

A scope is an administrative grouping of IP addresses for computers on a subnet that use the DHCP service. The administrator first creates a scope for each physical subnet and then uses the scope to define the parameters used

by clients.

A scope has the following properties:

- A range of IP addresses from which to include or exclude addresses used for DHCP service lease offerings.
- A subnet mask, which determines the subnet prefix for a given IP address.
- A scope name assigned when it is created.
- Lease duration values, which are assigned to DHCP clients that receive dynamically allocated IP addresses.
- Any DHCP scope options configured for assignment to DHCP clients, such as DNS server IP address and router/default gateway IP address.
- Reservations are optionally used to ensure that a DHCP client always receives the same IP address.

Before deploying your servers, list your subnets and the IP address range you want to use for each subnet.

Planning subnet masks

Network IDs and host IDs within an IP address are distinguished by using a subnet mask. Each subnet mask is a 32-bit number that uses consecutive bit groups of all ones (1) to identify the network ID and all zeroes (0) to identify the host ID portions of an IP address.

For example, the subnet mask normally used with the IP address 131.107.16.200 is the following 32-bit binary number:

```
11111111 11111111 00000000 00000000
```

This subnet mask number is 16 one-bits followed by 16 zero-bits, indicating that the network ID and host ID sections of this IP address are both 16 bits in length. Normally, this subnet mask is displayed in dotted decimal notation as 255.255.0.0.

The following table displays subnet masks for the Internet address classes.

ADDRESS CLASS	BITS FOR SUBNET MASK	SUBNET MASK
Class A	11111111 00000000 00000000 00000000	255.0.0.0
Class B	11111111 11111111 00000000 00000000	255.255.0.0
Class C	11111111 11111111 11111111 00000000	255.255.255.0

When you create a scope in DHCP and you enter the IP address range for the scope, DHCP provides these default subnet mask values. Typically, default subnet mask values are acceptable for most networks with no special requirements and where each IP network segment corresponds to a single physical network.

In some cases, you can use customized subnet masks to implement IP subnetting. With IP subnetting, you can subdivide the default host ID portion of an IP address to specify subnets, which are subdivisions of the original class-based network ID.

By customizing the subnet mask length, you can reduce the number of bits that are used for the actual host ID.

To prevent addressing and routing problems, you should make sure that all TCP/IP computers on a network segment use the same subnet mask and that each computer or device has an unique IP address.

Planning exclusion ranges

When you create a scope on a DHCP server, you specify an IP address range that includes all of the IP addresses that the DHCP server is allowed to lease to DHCP clients, such as computers and other devices. If you then go and manually configure some servers and other devices with static IP addresses from the same IP address range that the DHCP server is using, you can accidentally create an IP address conflict, where you and the DHCP server have both assigned the same IP address to different devices.

To solve this problem, you can create an exclusion range for the DHCP scope. An exclusion range is a contiguous range of IP addresses within the scope's IP address range that the DHCP server is not allowed to use. If you create an exclusion range, the DHCP server does not assign the addresses in that range, allowing you to manually assign these addresses without creating an IP address conflict.

You can exclude IP addresses from distribution by the DHCP server by creating an exclusion range for each scope. You should use exclusions for all devices that are configured with a static IP address. The excluded addresses should include all IP addresses that you assigned manually to other servers, non-DHCP clients, diskless workstations, or Routing and Remote Access and PPP clients.

It is recommended that you configure your exclusion range with extra addresses to accommodate future network growth. The following table provides an example exclusion range for a scope with an IP address range of 10.0.0.1 - 10.0.0.254 and a subnet mask of 255.255.255.0.

CONFIGURATION ITEMS	EXAMPLE VALUES
Exclusion range Start IP Address	10.0.0.1
Exclusion range End IP Address	10.0.0.25

Planning TCP/IP static configuration

Certain devices, such as routers, DHCP servers, and DNS servers, must be configured with a static IP address. In addition, you might have additional devices, such as printers, that you want to ensure always have the same IP address. List the devices that you want to configure statically for each subnet, and then plan the exclusion range you want to use on the DHCP server to ensure that the DHCP server does not lease the IP address of a statically configured device. An exclusion range is a limited sequence of IP addresses within a scope, excluded from DHCP service offerings. Exclusion ranges assure that any addresses in these ranges are not offered by the server to DHCP clients on your network.

For example, if the IP address range for a subnet is 192.168.0.1 through 192.168.0.254 and you have ten devices that you want to configure with a static IP address, you can create an exclusion range for the 192.168.0.x scope that includes ten or more IP addresses: 192.168.0.1 through 192.168.0.15.

In this example, you use ten of the excluded IP addresses to configure servers and other devices with static IP addresses and five additional IP addresses are left available for static configuration of new devices that you might want to add in the future. With this exclusion range, the DHCP server is left with an address pool of 192.168.0.16 through 192.168.0.254.

Additional example configuration items for AD DS and DNS are provided in the following table.

CONFIGURATION ITEMS	EXAMPLE VALUES
Network Connect Bindings	Ethernet
DNS Server Settings	DC1.corp.contoso.com
Preferred DNS server IP address	10.0.0.2

CONFIGURATION ITEMS	EXAMPLE VALUES
Scope values 1. Scope Name 2. Starting IP Address 3. Ending IP Address 4. Subnet Mask 5. Default Gateway (optional) 6. Lease duration	1. Primary Subnet 2. 10.0.0.1 3. 10.0.0.254 4. 255.255.255.0 5. 10.0.0.1 6. 8 days
IPv6 DHCP Server Operation Mode	Not enabled

Using This Guide in a Test Lab

You can use this guide to deploy DHCP in a test lab before you deploy in a production environment.

NOTE

If you do not want to deploy DHCP in a test lab, you can skip to the section [Deploy DHCP](#).

The requirements for your lab differ depending on whether you are using physical servers or virtual machines (VMs), and whether you are using an Active Directory domain or deploying a standalone DHCP server.

You can use the following information to determine the minimum resources you need to test DHCP deployment using this guide.

Test Lab requirements with VMs

To deploy DHCP in a test lab with VMs, you need the following resources.

For either domain deployment or standalone deployment, you need one server that is configured as a Hyper-V host.

Domain deployment

This deployment requires one physical server, one virtual switch, two virtual servers, and one virtual client:

On your physical server, in Hyper-V Manager, create the following items.

1. One **Internal** virtual switch. Do not create an **External** virtual switch, because if your Hyper-V host is on a subnet that includes a DHCP server, your test VMs will receive an IP address from your DHCP server. In addition, the test DHCP server that you deploy might assign IP addresses to other computers on the subnet where the Hyper-V host is installed.
2. One VM running Windows Server 2016 configured as a domain controller with Active Directory Domain Services that is connected to the Internal virtual switch you created. To match this guide, this server must have a statically configured IP address of 10.0.0.2. For information on deploying AD DS, see the section [Deploying DC1](#) in the Windows Server 2016 [Core Network Guide](#).
3. One VM running Windows Server 2016 that you will configure as a DHCP server by using this guide and that is connected to the Internal virtual switch you created.
4. One VM running a Windows client operating system that is connected to the Internal virtual switch you created and that you will use to verify that your DHCP server is dynamically allocating IP addresses and DHCP options to DHCP clients.

Standalone DHCP server deployment

This deployment requires one physical server, one virtual switch, one virtual server, and one virtual client:

On your physical server, in Hyper-V Manager, create the following items.

1. One **Internal** virtual switch. Do not create an **External** virtual switch, because if your Hyper-V host is on a subnet that includes a DHCP server, your test VMs will receive an IP address from your DHCP server. In addition, the test DHCP server that you deploy might assign IP addresses to other computers on the subnet where the Hyper-V host is installed.
2. One VM running Windows Server 2106 that you will configure as a DHCP server by using this guide and that is connected to the Internal virtual switch you created.
3. One VM running a Windows client operating system that is connected to the Internal virtual switch you created and that you will use to verify that your DHCP server is dynamically allocating IP addresses and DHCP options to DHCP clients.

Test Lab requirements with physical servers

To deploy DHCP in a test lab with physical servers, you need the following resources.

Domain deployment

This deployment requires one hub or switch, two physical servers and one physical client:

1. One Ethernet hub or switch to which you can connect the physical computers with Ethernet cables
2. One physical computer running Windows Server 2106 configured as a domain controller with Active Directory Domain Services. To match this guide, this server must have a statically configured IP address of 10.0.0.2. For information on deploying AD DS, see the section **Deploying DC1** in the Windows Server 2016 [Core Network Guide](#).
3. One physical computer running Windows Server 2106 that you will configure as a DHCP server by using this guide.
4. One physical computer running a Windows client operating system that you will use to verify that your DHCP server is dynamically allocating IP addresses and DHCP options to DHCP clients.

NOTE

If you do not have enough test machines for this deployment, you can use one test machine for both AD DS and DHCP - however this configuration is not recommended for a production environment.

Standalone DHCP server deployment

This deployment requires one hub or switch, one physical server, and one physical client:

1. One Ethernet hub or switch to which you can connect the physical computers with Ethernet cables
2. One physical computer running Windows Server 2106 that you will configure as a DHCP server by using this guide.
3. One physical computer running a Windows client operating system that you will use to verify that your DHCP server is dynamically allocating IP addresses and DHCP options to DHCP clients.

Deploy DHCP

This section provides example Windows PowerShell commands that you can use to deploy DHCP on one server. Before you run these example commands on your server, you must modify the commands to match your network and environment.

For example, before you run the commands, you should replace example values in the commands for the following items:

- Computer names
- IP Address range for each scope you want to configure (1 scope per subnet)

- Subnet mask for each IP address range you want to configure
- Scope name for each scope
- Exclusion range for each scope
- DHCP option values, such as default gateway, domain name, and DNS or WINS servers
- Interface names

IMPORTANT

Examine and modify every command for your environment before you run the command.

Where to Install DHCP - on a physical computer or a VM?

You can install the DHCP server role on a physical computer or on a virtual machine (VM) that is installed on a Hyper-V host. If you are installing DHCP on a VM and you want the DHCP server to provide IP address assignments to computers on the physical network to which the Hyper-V host is connected, you must connect the VM virtual network adapter to a Hyper-V Virtual Switch that is **External**.

For more information, see the section **Create a Virtual Switch with Hyper-V Manager** in the topic [Create a virtual network](#).

Run Windows PowerShell as an Administrator

You can use the following procedure to run Windows PowerShell with Administrator privileges.

1. On a computer running Windows Server 2016, click **Start**, then right-click the Windows PowerShell icon. A menu appears.
2. In the menu, click **More**, and then click **Run as administrator**. If prompted, type the credentials for an account that has Administrator privileges on the computer. If the user account with which you are logged on to the computer is an Administrator level account, you will not receive a credential prompt.
3. Windows PowerShell opens with Administrator privileges.

Rename the DHCP server and configure a static IP address

If you have not already done so, you can use the following Windows PowerShell commands to rename the DHCP server and configure a static IP address for the server.

Configure a static IP address

You can use the following commands to assign a static IP address to the DHCP server, and to configure the DHCP server TCP/IP properties with the correct DNS server IP address. You must also replace interface names and IP addresses in this example with the values that you want to use to configure your computer.

```
New-NetIPAddress -IPAddress 10.0.0.3 -InterfaceAlias "Ethernet" -DefaultGateway 10.0.0.1 -AddressFamily IPv4 -PrefixLength 24
```

```
Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses 10.0.0.2
```

For more information about these commands, see the following topics.

- [New-NetIPAddress](#)
- [Set-DnsClientServerAddress](#)

Rename the computer

You can use the following commands to rename and then restart the computer.

```
Rename-Computer -Name DHCP1
```

```
Restart-Computer
```

For more information about these commands, see the following topics.

- [Rename-Computer](#)
- [Restart-Computer](#)

Join the computer to the domain (Optional)

If you are installing your DHCP server in an Active Directory domain environment, you must join the computer to the domain. Open Windows PowerShell with Administrator privileges, and then run the following command after replacing the domain NetBios name **CORP** with a value that is appropriate for your environment.

```
Add-Computer CORP
```

When prompted, type the credentials for a domain user account that has permission to join a computer to the domain.

```
Restart-Computer
```

For more information about the Add-Computer command, see the following topic.

- [Add-Computer](#)

Install DHCP

After the computer restarts, open Windows PowerShell with Administrator privileges, and then install DHCP by running the following command.

```
Install-WindowsFeature DHCP -IncludeManagementTools
```

For more information about this command, see the following topic.

- [Install-WindowsFeature](#)

Create DHCP security groups

To create security groups, you must run a Network Shell (netsh) command in Windows PowerShell, and then restart the DHCP service so that the new groups become active.

When you run the following netsh command on the DHCP server, the **DHCP Administrators** and **DHCP Users** security groups are created in **Local Users and Groups** on the DHCP server.

```
netsh dhcp add securitygroups
```

The following command restarts the DHCP service on the local computer.

```
Restart-service dhcpserver
```

For more information about these commands, see the following topics.

- [Network Shell \(Netsh\)](#)
- [Restart-Service](#)

Authorize the DHCP server in Active Directory (Optional)

If you are installing DHCP in a domain environment, you must perform the following steps to authorize the DHCP server to operate in the domain.

NOTE

Unauthorized DHCP servers that are installed in Active Directory domains cannot function properly, and do not lease IP addresses to DHCP clients. The automatic disabling of unauthorized DHCP servers is a security feature that prevents unauthorized DHCP servers from assigning incorrect IP addresses to clients on your network.

You can use the following command to add the DHCP server to the list of authorized DHCP servers in Active Directory.

NOTE

If you do not have a domain environment, do not run this command.

```
Add-DhcpServerInDC -DnsName DHCP1.corp.contoso.com -IpAddress 10.0.0.3
```

To verify that the DHCP server is authorized in Active Directory, you can use the following command.

```
Get-DhcpServerInDC
```

Following are example results that are displayed in Windows PowerShell.

IpAddress	DnsName
-----	-----
10.0.0.3	DHCP1.corp.contoso.com

For more information about these commands, see the following topics.

- [Add-DhcpServerInDC](#)
- [Get-DhcpServerInDC](#)

Notify Server Manager that post-install DHCP configuration is complete (Optional)

After you have completed post-installation tasks, such as creating security groups and authorizing the DHCP server in Active Directory, Server Manager might still display an alert in the user interface stating that post-installation steps must be completed by using the DHCP Post Installation Configuration wizard.

You can prevent this now-unnecessary and inaccurate message from appearing in Server Manager by configuring the following registry key using this Windows PowerShell command.

```
Set-ItemProperty -Path registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ServerManager\Roles\12 -Name ConfigurationState -Value 2
```

For more information about this command, see the following topic.

- [Set-ItemProperty](#)

Set server level DNS dynamic update configuration settings (Optional)

If you want the DHCP server to perform DNS dynamic updates for DHCP client computers, you can run the following command to configure this setting. This is a server level setting, not a scope level setting, so it will affect all scopes that you configure on the server. This example command also configures the DHCP server to delete DNS resource records for clients when the client lease expires.

```
Set-DhcpServerv4DnsSetting -ComputerName "DHCP1.corp.contoso.com" -DynamicUpdates "Always" -  
DeleteDnsRRonLeaseExpiry $True
```

You can use the following command to configure the credentials that the DHCP server uses to register or unregister client records on a DNS server. This example saves a credential on a DHCP server. The first command uses **Get-Credential** to create a **PSCredential** object, and then stores the object in the **\$Credential** variable. The command prompts you for user name and password, so ensure that you provide credentials for an account that has permission to update resource records on your DNS server.

```
$Credential = Get-Credential  
Set-DhcpServerDnsCredential -Credential $Credential -ComputerName "DHCP1.corp.contoso.com"
```

For more information about these commands, see the following topics.

- [Set-DhcpServerv4DnsSetting](#)
- [Set-DhcpServerDnsCredential](#)

Configure the Corpnet Scope

After DHCP installation is completed, you can use the following commands to configure and activate the Corpnet scope, create an exclusion range for the scope, and configure the DHCP options default gateway, DNS server IP address, and DNS domain name.

```
Add-DhcpServerv4Scope -name "Corpnet" -StartRange 10.0.0.1 -EndRange 10.0.0.254 -SubnetMask 255.255.255.0 -  
State Active`  
  
Add-DhcpServerv4ExclusionRange -ScopeID 10.0.0.0 -StartRange 10.0.0.1 -EndRange 10.0.0.15`  
  
Set-DhcpServerv4OptionValue -OptionID 3 -Value 10.0.0.1 -ScopeID 10.0.0.0 -ComputerName  
DHCP1.corp.contoso.com`  
  
Set-DhcpServerv4OptionValue -DnsDomain corp.contoso.com -DnsServer 10.0.0.2
```

For more information about these commands, see the following topics.

- [Add-DhcpServerv4Scope](#)
- [Add-DhcpServerv4ExclusionRange](#)
- [Set-DhcpServerv4OptionValue](#)

Configure the Corpnet2 Scope (Optional)

If you have a second subnet that is connected to the first subnet with a router where DHCP forwarding is enabled, you can use the following commands to add a second scope, named Corpnet2 for this example. This example also configures an exclusion range and the IP address for the default gateway (the router IP address on the subnet) of the Corpnet2 subnet.

```
Add-DhcpServerv4Scope -name "Corpnet2" -StartRange 10.0.1.1 -EndRange 10.0.1.254 -SubnetMask 255.255.255.0 -  
State Active`
```

```
Add-DhcpServerv4ExclusionRange -ScopeID 10.0.1.0 -StartRange 10.0.1.1 -EndRange 10.0.1.15`
```

```
Set-DhcpServerv4OptionValue -OptionID 3 -Value 10.0.1.1 -ScopeID 10.0.1.0 -ComputerName DHCP1.corp.contoso.com`
```

If you have additional subnets that are serviced by this DHCP server, you can repeat these commands, using different values for all of the command parameters, to add scopes for each subnet.

IMPORTANT

Ensure that all routers between your DHCP clients and your DHCP server are configured for DHCP message forwarding. See your router documentation for information on how to configure DHCP forwarding.

Verify Server Functionality

To verify that your DHCP server is providing dynamic allocation of IP addresses to DHCP clients, you can connect another computer to a serviced subnet. After you connect the Ethernet cable to the network adapter and power on the computer, it will request an IP address from your DHCP server. You can verify successful configuration by using the **ipconfig /all** command and reviewing the results, or by performing connectivity tests, such as attempting to access Web resources with your browser or file shares with Windows Explorer or other applications.

If the client does not receive an IP address from your DHCP server, perform the following troubleshooting steps.

1. Ensure that the Ethernet cable is plugged into both the computer and the Ethernet switch, hub, or router.
2. If you plugged the client computer into a network segment that is separated from the DHCP server by a router, ensure that the router is configured to forward DHCP messages.
3. Ensure that the DHCP server is authorized in Active Directory by running the following command to retrieve the list of authorized DHCP servers from Active Directory. [Get-DhcpServerInDC](#).
4. Ensure that your scopes are activated by opening the DHCP console (Server Manager, **Tools, DHCP**), expanding the server tree to review scopes, then right-clicking each scope. If the resulting menu includes the selection **Activate**, click **Activate**. (If the scope is already activated, the menu selection reads **Deactivate**.)

Windows PowerShell Commands for DHCP

The following reference provides command descriptions and syntax for all DHCP Server Windows PowerShell commands for Windows Server 2016. The topic lists commands in alphabetical order based on the verb at the beginning of the commands, such as **Get** or **Set**.

NOTE

You can not use Windows Server 2016 commands in Windows Server 2012 R2.

- [DhcpServer Module](#)

The following reference provides command descriptions and syntax for all DHCP Server Windows PowerShell commands for Windows Server 2012 R2. The topic lists commands in alphabetical order based on the verb at the beginning of the commands, such as **Get** or **Set**.

NOTE

You can use Windows Server 2012 R2 commands in Windows Server 2016.

- [DHCP Server Cmdlets in Windows PowerShell](#)

List of Windows PowerShell Commands in this guide

Following is a simple list of commands and example values that are used in this guide.

```
New-NetIPAddress -IPAddress 10.0.0.3 -InterfaceAlias "Ethernet" -DefaultGateway 10.0.0.1 -AddressFamily IPv4 -  
PrefixLength 24  
Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses 10.0.0.2  
Rename-Computer -Name DHCP1  
Restart-Computer  
  
Add-Computer CORP  
Restart-Computer  
  
Install-WindowsFeature DHCP -IncludeManagementTools  
netsh dhcp add securitygroups  
Restart-service dhcpserver  
  
Add-DhcpServerInDC -DnsName DHCP1.corp.contoso.com -IPAddress 10.0.0.3  
Get-DhcpServerInDC  
  
Set-ItemProperty -Path registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ServerManager\Roles\12 -Name  
ConfigurationState -Value 2  
  
Set-DhcpServerv4DnsSetting -ComputerName "DHCP1.corp.contoso.com" -DynamicUpdates "Always" -  
DeleteDnsRRonLeaseExpiry $True  
  
$Credential = Get-Credential  
Set-DhcpServerDnsCredential -Credential $Credential -ComputerName "DHCP1.corp.contoso.com"  
  
rem At prompt, supply credential in form DOMAIN\user, password  
  
rem Configure scope Corpnet  
  
Add-DhcpServerv4Scope -name "Corpnet" -StartRange 10.0.0.1 -EndRange 10.0.0.254 -SubnetMask 255.255.255.0 -  
State Active  
  
Add-DhcpServerv4ExclusionRange -ScopeID 10.0.0.0 -StartRange 10.0.0.1 -EndRange 10.0.0.15  
  
Set-DhcpServerv4OptionValue -OptionID 3 -Value 10.0.0.1 -ScopeID 10.0.0.0 -ComputerName DHCP1.corp.contoso.com  
  
Set-DhcpServerv4OptionValue -DnsDomain corp.contoso.com -DnsServer 10.0.0.2  
  
rem Configure scope Corpnet2  
  
Add-DhcpServerv4Scope -name "Corpnet2" -StartRange 10.0.1.1 -EndRange 10.0.1.254 -SubnetMask 255.255.255.0 -  
State Active  
  
Add-DhcpServerv4ExclusionRange -ScopeID 10.0.1.0 -StartRange 10.0.1.1 -EndRange 10.0.1.15  
  
Set-DhcpServerv4OptionValue -OptionID 3 -Value 10.0.1.1 -ScopeID 10.0.1.0 -ComputerName DHCP1.corp.contoso.com
```

High-performance networking (HPN)

9/21/2018 • 2 minutes to read • [Edit Online](#)

High-performance networks (HPNs) play a role in real-time data processing requirements. For example, activities such as datacenter replication, datacenter disaster recovery, and high-performance distributed computing require high volume data transfer and low network latency. HPNs with dynamic connection capabilities make high-performance network resources more accessible and manageable.

The high-performance networking topics include:

- [Insider preview](#)
 - [Network offload and optimization technologies](#)
 - [Software only \(SO\) features and technologies](#)
 - [Software and hardware \(SH\) integrated features and technologies](#)
 - [Hardware Only \(HO\) features and technologies](#)
 - [NIC advanced properties](#)
 - [RSC in the vSwitch](#)
-

Network offload and optimization technologies

9/21/2018 • 2 minutes to read • [Edit Online](#)

In this topic, we give you an overview of the different network offload and optimization features available in Windows Server 2016 and discuss how they help make networking more efficient. These technologies include Software Only (SO) features and technologies, Software and Hardware (SH) integrated features and technologies, and Hardware Only (HO) features and technologies.

The three categories of networking features available in Windows Server 2016 are:

1. **Software only (SO) features and technologies:** These features are implemented as part of the OS and are independent of the underlying NIC(s). Sometimes these features will require some tuning of the NIC for optimal operation. Examples of these include Hyper-v features such as vmQoS, ACLs, and non-Hyper-V features like NIC Teaming.
2. **Software and Hardware (SH) integrated features and technologies:** These features have both software and hardware components. The software is intimately tied to hardware capabilities that are required for the feature to work. Examples of these include VMMQ, VMQ, Send-side IPv4 Checksum Offload, and RSS.
3. **Hardware Only (HO) features and technologies:** These hardware accelerations improve networking performance in conjunction with the software but are not intimately part of any software feature. Examples of these include Interrupt Moderation, Flow Control, and Receive-side IPv4 Checksum Offload.
4. **NIC advanced properties:** You can manage NICs and all the features via Windows PowerShell using the NetAdapter cmdlet. You can also manage NICs and all the features using Network Control Panel (ncpa.cpl).

TIP

- SO features and technologies are available in all hardware architectures, regardless of NIC speed or NIC capabilities.
- SH and HO features are available only when your network adapter supports the features or technologies.

Software only (SO) features and technologies

9/21/2018 • 4 minutes to read • [Edit Online](#)

Software only features are implemented as part of the OS and are independent of the underlying NIC(s). Sometimes these features require some tuning of the NIC for optimal operation. Examples of these include Hyper-V features such as Virtual Machine Quality of Service (vmQoS), Access Control Lists (ACLs), and non-Hyper-V features like NIC Teaming.

Access Control Lists (ACLs)

A Hyper-V and SDNv1 feature for managing security for a VM. This feature applies to the non-virtualized Hyper-V stack and the HVnv1 stack. You can manage Hyper-V switch ACLs through [Add-VMNetworkAdapterAcl](#) and [Remove-VMNetworkAdapterAcl](#) PowerShell cmdlets.

Extended ACLs

Hyper-V Virtual Switch extended ACLs enable you to configure the Hyper-V Virtual Switch Extended Port ACLs to provide firewall protection and enforce security policies for the tenant VMs in datacenters. Because the port ACLs are configured on the Hyper-V Virtual Switch rather than within the VMs, the administrator can manage security policies for all tenants in a multitenant environment.

You can manage Hyper-V switch extended ACLs through the [Add-VMNetworkAdapterExtendedAcl](#) and [Remove-VMNetworkAdapterExtendedAcl](#) PowerShell cmdlets.

TIP

This feature applies to the HVnv1 stack. For ACLs in the SDN stack, refer to Software Defined Networking SDN) ACLs below.

For more information about Extended Port Access Control Lists in this library, see [Create Security Policies with Extended Port Access Control Lists](#).

NIC Teaming

NIC Teaming, also called NIC bonding, is the aggregation of multiple NIC ports into an entity the host perceives as a single NIC port. NIC Teaming protects against the failure of a single NIC port (or the cable connected to it). It also aggregates network traffic for faster throughput. For more details, see [NIC Teaming](#).

With Windows Server 2016 you have two ways to do teaming:

1. Windows Server 2012 teaming solution
2. Windows Server 2016 Switch Embedded Teaming (SET)

RSC in the vSwitch

Receive Segment Coalescing (RSC) in the vSwitch is a feature that takes packets that are part of the same stream and arrive between network interrupts, and coalesces them into a single packet before delivering them to the operating system. The virtual switch in Windows Server 2019 has this feature. For more details about this feature, see [Receive Segment Coalescing in the vSwitch](#).

Software Defined Networking (SDN) ACLs

The SDN-extension in Windows Server 2016 improved ways to support ACLs. In the Windows Server 2016 SDN v2 stack, SDN ACLs are used instead of ACLs and Extended ACLs. You can use Network Controller to manage SDN ACLs.

SDN Quality of Service (QoS)

The SDN extension in Windows Server 2016 improved ways to provide bandwidth control (egress reservations, egress limits, and ingress limits) on a 5-tuple basis. Typically, these policies get applied at the vNIC or vmNIC level, but you can make them much more specific. In the Windows Server 2016 SDN v2 stack, SDN QoS is used instead of vmQoS. You can use Network Controller to manage SDN QoS.

Switch Embedded Teaming (SET)

SET is an alternative NIC Teaming solution that you can use in environments that include Hyper-V and the Software Defined Networking (SDN) stack in Windows Server 2016. SET integrates some NIC Teaming functionality into the Hyper-V Virtual Switch. For information about Switch Embedded Teaming in this library, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#).

Virtual Receive Side Scaling (vRSS)

Software vRSS is used to spread incoming traffic destined for a VM across multiple logical processors (LPs) of the VM. Software vRSS gives the VM the ability to handle more networking traffic than a single LP would be able to handle. For more information, see [Virtual Receive Side Scaling \(vRSS\)](#).

Virtual Machine Quality of Service (vmQoS)

Virtual Machine Quality of Service is a Hyper-V feature that allows the switch to set limits on traffic generated by each VM. It also enables a VM to reserve an amount of bandwidth on the external network connection so that one VM can't starve another VM for bandwidth. In the Windows Server 2016 SDN v2 stack, SDN QoS replaces vmQoS.

vmQoS can set egress limits and egress reservations. You must determine the egress reservation mode (relative weight or absolute bandwidth) before creating the Hyper-V switch.

- Determine the egress reservation mode with the `-MinimumBandwidthMode` parameter of the `New-VMSwitch` PowerShell cmdlet.
- Set the value of the egress limit with the `-MaximumBandwidth` parameter on the `Set-VMNetworkAdapter` PowerShell cmdlet.
- Set the value for the egress reservation with either of the following parameters of the `Set-VMNetworkAdapter` PowerShell cmdlet:
 - If the `-MinimumBandwidthMode` parameter on the `New-VMSwitch` cmdlet is `Absolute`, then set the `-MinimumBandwidthAbsolute` parameter on the `Set-VMNetworkAdapter` cmdlet.
 - If the `-MinimumBandwidthMode` parameter on the `New-VMSwitch` cmdlet is `Weight`, then set the `-MinimumBandwidthWeight` parameter on the `Set-VMNetworkAdapter` cmdlet.

Because of the limitations in the algorithm used for this feature, we recommend that the highest weight or absolute bandwidth not be more than 20 times the lowest weight or absolute bandwidth. If more control is needed, consider using the SDN stack and the SDN-QoS feature.

Software and hardware (SH) integrated features and technologies

9/21/2018 • 6 minutes to read • [Edit Online](#)

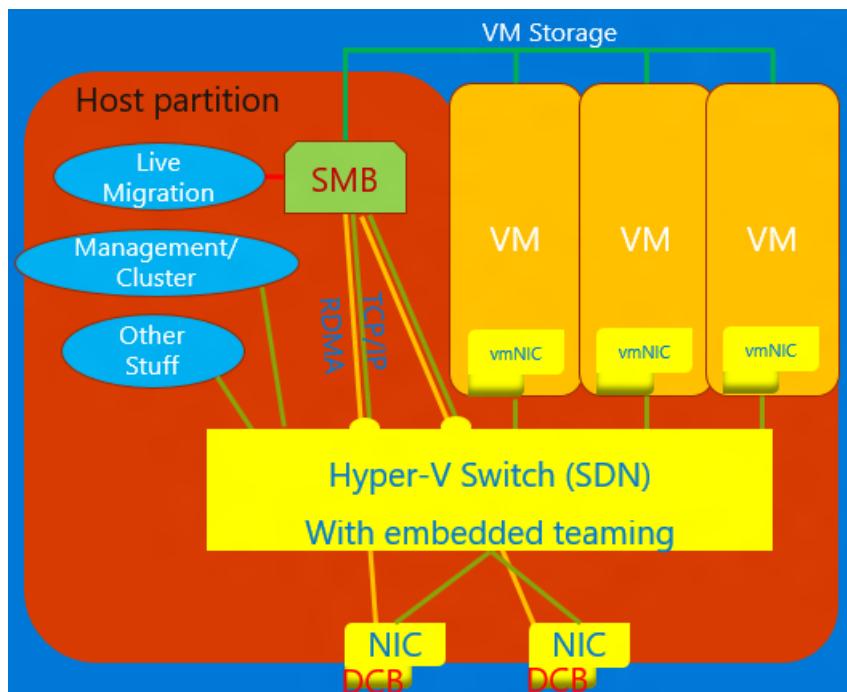
These features have both software and hardware components. The software is intimately tied to hardware capabilities that are required for the feature to work. Examples of these include VMMQ, VMQ, Send-side IPv4 Checksum Offload, and RSS.

TIP

SH and HO features are available if the installed NIC supports it. The feature descriptions below will cover how to tell if your NIC supports the feature.

Converged NIC

Converged NIC is a technology that allows virtual NICs in the Hyper-V host to expose RDMA services to host processes. Windows Server 2016 no longer requires separate NICs for RDMA. The Converged NIC feature allows the Virtual NICs in the Host partition (vNICs) to expose RDMA to the host partition and share the bandwidth of the NICs between the RDMA traffic and the VM and other TCP/UDP traffic in a fair and manageable manner.



You can manage converged NIC operation through VMM or Windows PowerShell. The PowerShell cmdlets are the same cmdlets used for RDMA (see below).

To use the converged NIC capability:

1. Ensure to set the host up for DCB.
2. Ensure to enable RDMA on the NIC, or in the case of a SET team, the NICs are bound to the Hyper-V switch.
3. Ensure to enable RDMA on the vNICs designated for RDMA in the host.

For more details about RDMA and SET, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#).

Data Center Bridging (DCB)

DCB is a suite of Institute of Electrical and Electronics Engineers (IEEE) standards that enable Converged Fabrics in data centers. DCB provides hardware queue-based bandwidth management in a host with cooperation from the adjacent switch. All traffic for storage, data networking, cluster Inter-Process Communication (IPC), and management share the same Ethernet network infrastructure. In Windows Server 2016, DCB can be applied to any NIC individually and to NICs bound to the Hyper-V switch.

For DCB, Windows Server uses Priority-based Flow Control (PFC), standardized in IEEE 802.1Qbb. PFC creates a (nearly) lossless network fabric by preventing overflow within traffic classes. Windows Server also uses Enhanced Transmission Selection (ETS), standardized in IEEE 802.1Qaz. ETS enables the division of the bandwidth into reserved portions for up to eight classes of traffic. Each traffic class has its own transmit queue and, through the use of PFC, can start and stop transmission within a class.

For more information, see [Data Center Bridging \(DCB\)](#).

Hyper-V Network Virtualization

v1 (HNVv1)	Introduced in Windows Server 2012, Hyper-V Network Virtualization (HNV) enables virtualization of customer networks on top of a shared, physical network infrastructure. With minimal changes necessary on the physical network fabric, HNV gives service providers the agility to deploy and migrate tenant workloads anywhere across the three clouds: the service provider cloud, the private cloud, or the Microsoft Azure public cloud.
v2 NVGRE (HNVv2 NVGRE)	In Windows Server 2016 and System Center Virtual Machine Manager, Microsoft provides an end-to-end network virtualization solution that includes RAS Gateway, Software Load Balancing, Network Controller, and more. For more information, see Hyper-V Network Virtualization Overview in Windows Server 2016 .
v2 VxLAN (HNVv2 VxLAN)	In Windows Server 2016, is part of the SDN-extension, which you manage through the Network Controller.

IPsec Task Offload (IPsecTO)

IPsec task offload is a NIC feature that enables the operating system to use the processor on the NIC for the IPsec encryption work.

IMPORTANT

IPsec Task Offload is a legacy technology that is not supported by most network adapters, and where it does exist, it's disabled by default.

Private virtual Local Area Network (PVLAN).

PVLANS allow communication only between virtual machines on the same virtualization server. A private virtual

network is not bound to a physical network adapter. A private virtual network is isolated from all external network traffic on the virtualization server, as well as any network traffic between the management operating system and the external network. This type of network is useful when you need to create an isolated networking environment, such as an isolated test domain. The Hyper-V and SDN stacks support PVLAN Isolated Port mode only.

For details about PVLAN isolation, see [System Center: Virtual Machine Manager Engineering Blog](#).

Remote Direct Memory Access (RDMA)

RDMA is a networking technology that provides high-throughput, low-latency communication that minimizes CPU usage. RDMA supports zero-copy networking by enabling the network adapter to transfer data directly to or from application memory. RDMA-capable means the NIC (physical or virtual) is capable of exposing RDMA to an RDMA client. RDMA-enabled, on the other hand, means an RDMA-capable NIC is exposing the RDMA interface up the stack.

For more details about RDMA, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#).

Receive Side Scaling (RSS)

RSS is a NIC feature that segregates different sets of streams and delivers them to different processors for processing. RSS parallelizes the networking processing, enabling a host to scale to very high data rates.

For more details, see [Receive Side Scaling \(RSS\)](#).

Single Root Input-Output Virtualization (SR-IOV)

SR-IOV allows VM traffic to move directly from the NIC to the VM without passing through the Hyper-V host. SR-IOV is an incredible improvement in performance for a VM but lacks the ability for the host to manage that pipe. Only use SR-IOV when the workload is well-behaved, trusted, and generally the only VM in the host.

Traffic that uses SR-IOV bypasses the Hyper-V switch, which means that any policies, for example, ACLs, or bandwidth management won't be applied. SR-IOV traffic also can't be passed through any network virtualization capability, so NV-GRE or VxLAN encapsulation can't be applied. Only use SR-IOV for well-trusted workloads in specific situations. Additionally, you cannot use the host policies, bandwidth management, and virtualization technologies.

In the future, two technologies would allow SR-IOV: Generic Flow Tables (GFT) and Hardware QoS Offload (bandwidth management in the NIC) – once the NICs in our ecosystem support them. The combination of these two technologies would make SR-IOV useful for all VMs, would allow policies, virtualization, and bandwidth management rules to be applied, and could result in great leaps forward in the general application of SR-IOV.

For more details, see [Overview of Single Root I/O Virtualization \(SR-IOV\)](#).

TCP Chimney Offload

TCP Chimney Offload, also known as TCP Engine Offload (TOE), is a technology that allows the host to offload all TCP processing to the NIC. Because the Windows Server TCP stack is almost always more efficient than the TOE engine, using TCP Chimney Offload is not recommended.

IMPORTANT

TCP Chimney Offload is a deprecated technology. We recommend you do not use TCP Chimney Offload as Microsoft might stop supporting it in the future.

Virtual Local Area Network (VLAN)

VLAN is an extension to the Ethernet frame header to enable partitioning of a LAN into multiple VLANs, each using its own address space. In Windows Server 2016, VLANs are set on ports of the Hyper-V switch or by setting team interfaces on NIC Teaming teams. For more information, see [NIC Teaming and Virtual Local Area Networks \(VLANs\)](#).

Virtual Machine Queue (VMQ)

VMQs is a NIC feature that allocates a queue for each VM. Anytime you have Hyper-V enabled; you must also enable VMQ. In Windows Server 2016, VMQs use NIC Switch vPorts with a single queue assigned to the vPort to provide the same functionality. For more information, see [Virtual Receive Side Scaling \(vRSS\)](#) and [NIC Teaming](#).

Virtual Machine Multi-Queue (VMMQ)

VMMQ is a NIC feature that allows traffic for a VM to spread across multiple queues, each processed by a different physical processor. The traffic is then passed to multiple LPs in the VM as it would be in vRSS, which allows for delivering substantial networking bandwidth to the VM.

Hardware Only (HO) features and technologies

9/21/2018 • 3 minutes to read • [Edit Online](#)

These hardware accelerations improve networking performance in conjunction with the software but are not intimately part of any software feature. Examples of these include Interrupt Moderation, Flow Control, and Receive-side IPv4 Checksum Offload.

TIP

SH and HO features are available if the installed NIC supports it. The feature descriptions below will cover how to tell if your NIC supports the feature.

Address Checksum Offload

Address checksum offloads are a NIC feature that offloads the calculation of address checksums (IP, TCP, UDP) to the NIC hardware for both send and receive.

On the receive path, the checksum offload calculates the checksums in the IP, TCP, and UDP headers (as appropriate) and indicates to the OS whether the checksums passed, failed, or not checked. If the NIC asserts that the checksums are valid, the OS accepts the packet unchallenged. If the NIC asserts the checksums are invalid or not checked, the IP/TCP/UDP stack internally calculates the checksums again. If the computed checksum fails, the packet gets discarded.

On the send path, the checksum offload calculates and inserts the checksums into the IP, TCP, or UDP header as appropriate.

Disabling checksum offloads on the send path does not disable checksum calculation and insertion for packets sent to the miniport driver using the Large Send Offload (LSO) feature. To disable all checksum offload calculations, the user must also disable LSO.

Manage Address Checksum Offloads

In the Advanced Properties there are several distinct properties:

- IPv4 Checksum Offload
- TCP Checksum Offload (IPv4)
- TCP Checksum Offload (IPv6)
- UDP Checksum Offload (IPv4)
- UDP Checksum Offload (IPv6)

By default, these are all always enabled. We recommend always enabling all of these offloads.

The Checksum Offloads can be managed using the `Enable-NetAdapterChecksumOffload` and `Disable-NetAdapterChecksumOffload` cmdlets. For example, the following cmdlet enables the TCP (IPv4) and UDP (IPv4) checksum calculations:

```
Enable-NetAdapterChecksumOffload -Name * -TcpIPv4 -UdpIPv4
```

Tips on using Address Checksum Offloads

Address Checksum Offloads should ALWAYS be enabled no matter what workload or circumstance. This most

basic of all offload technologies always improve your network performance. Checksum offloading is also required for other stateless offloads to work including receive side scaling (RSS), receive segment coalescing (RSC), and large send offload (LSO).

Interrupt Moderation (IM)

IM buffers multiple received packets before interrupting the operating system. When a NIC receives a packet, it starts a timer. When the buffer is full, or the timer expires, whichever comes first, the NIC interrupts the operating system.

Many NICs support more than just on/off for Interrupt Moderation. Most NICs support the concepts of a low, medium, and high rate for IM. The different rates represent shorter or longer timers and appropriate buffer size adjustments to reduce latency (low interrupt moderation) or reduce interrupts (high interrupt moderation).

There is a balance to be struck between reducing interrupts and excessively delaying packet delivery. Generally, packet processing is more efficient with Interrupt Moderation enabled. High performance or low latency applications may need to evaluate the impact of disabling or reducing Interrupt Moderation.

Jumbo frames

Jumbo frames is a NIC and network feature that allows an application to send frames that are much larger than the default 1500 bytes. Typically the limit on jumbo frames is about 9000 bytes but may be smaller.

There were no changes to jumbo frame support in Windows Server 2012 R2.

In Windows Server 2016 there is a new offload: MTU_for_HNV. This new offload works with Jumbo Frame settings to ensure encapsulated traffic doesn't require segmentation between the host and the adjacent switch. This new feature of the SDN stack has the NIC automatically calculate what MTU to advertise and what MTU to use on the wire. These values for MTU are different if any HNV offload is in use. (In the feature compatibility table, Table 1, MTU_for_HNV would have the same interactions as the HNVv2 offloads have since it is directly related to the HNVv2 offloads.)

Large Send Offload (LSO)

LSO allows an application to pass a large block of data to the NIC, and the NIC breaks the data into packets that fit within the Maximum Transfer Unit (MTU) of the network.

Receive Segment Coalescing (RSC)

Receive Segment Coalescing, also known as Large Receive Offload, is a NIC feature that takes packets that are part of the same stream that arrives between network interrupts and coalesces them into a single packet before delivering them to the operating system. RSC is not available on NICs that are bound to the Hyper-V Virtual Switch. For more information, see [Receive Segment Coalescing \(RSC\)](#).

NIC advanced properties

9/21/2018 • 2 minutes to read • [Edit Online](#)

You can manage NICs and all the features via Windows PowerShell using the [NetAdapter](#) cmdlet. You can also manage NICs and all the features using Network Control Panel (ncpa.cpl).

1. In **Windows PowerShell**, run the `Get-NetAdapterAdvancedProperty` cmdlet against two different make/model of NICs.

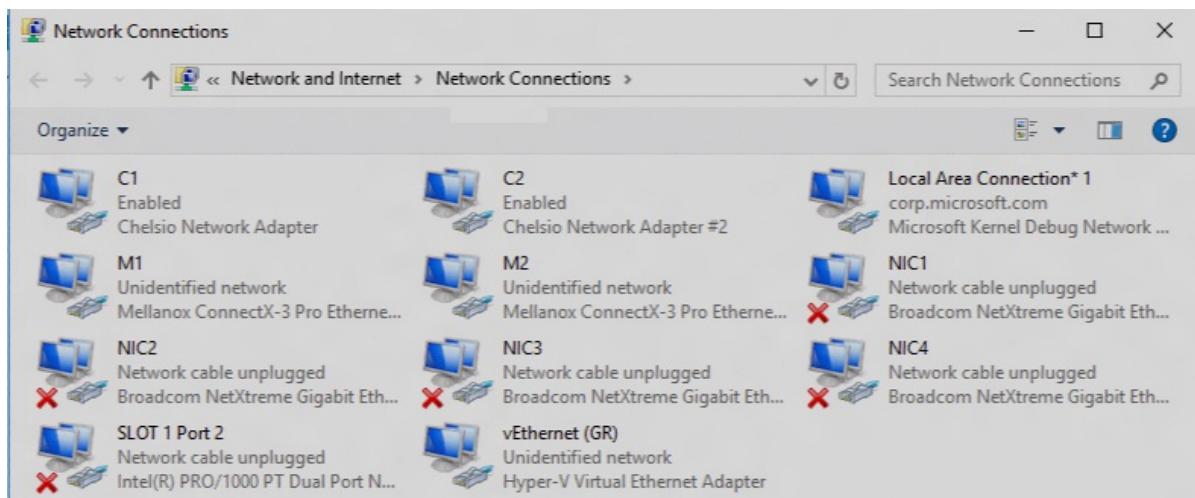
Name	DisplayName	DisplayValue	RegistryKeyword	RegistryValue
M1	Encapsulated Task Offload	Enabled	*Encapsulate...	{1}
M1	Flow Control	Rx & Tx Enabled	*FlowControl	{3}
M1	Header Data Split	Disabled	*HeaderDataS...	{0}
M1	Interrupt Moderation	Enabled	*InterruptMo...	{1}
M1	IPv4 Checksum Offload	Rx & Tx Enabled	*IPChecksumO...	{3}
M1	Jumbo Packet	1514	*JumboPacket	{1514}
M1	Large Send Offload V2 (IPv4)	Enabled	*LsoV2IPv4	{1}
M1	Large Send Offload V2 (IPv6)	Enabled	*LsoV2IPv6	{1}
M1	Maximum number of RSS Proce...	8	*MaxRssProce...	{8}
M1	NetworkDirect Functionality	Enabled	*NetworkDirect	{1}
M1	Preferred NUMA node	Default Settings	*NumaNodeId	{65535}
M1	Maximum Number of RSS Queues	8	*NumRSSQueues	{8}
M1	Priority & Vlan Tag	Priority & VLAN Enabled	*PriorityVLA...	{3}
M1	Quality Of Service	Enabled	*QoS	{1}
M1	Receive Buffers	512	*ReceiveBuffers	{512}
M1	Receive Side Scaling	Enabled	*RSS	{1}
M1	RSS Base Processor Number	0	*RssBaseProc...	{0}
M1	RSS Maximum Processor Number	63	*RssMaxProcN...	{63}
M1	RSS load balancing Profile	ClosestProcessor	*RSSProfile	{1}
M1	SR-IOV	Enabled	*Sriov	{1}
M1	TCP/UDP Checksum Offload (I...	Rx & Tx Enabled	*TCPUDPCheck...	{3}
M1	TCP/UDP Checksum Offload (I...	Rx & Tx Enabled	*TCPUDPCheck...	{3}
M1	Send Buffers	2048	*TransmitBuf...	{2048}
M1	Virtual Machine Queues	Enabled	*VMQ	{1}
M1	VMQ VLAN Filtering	Enabled	*VMQVlanFilt...	{1}
M1	Force NDK to work with Glob...	Disabled	NdkWithGloba...	{0}
M1	Locally Administered Address	--	NetworkAddress	{--}
M1	Transmit Control Blocks	16	NumTcb	{16}
M1	Receive Completion Method	Adaptive	RecvCompleti...	{1}
M1	R/RoCE Max Frame Size	Auto	RoceMaxFrame...	{0}
M1	Rx Interrupt Moderation Type	Adaptive	RxIntModeration	{2}
M1	Rx Interrupt Moderation Pro...	Moderate	RxIntModerat...	{1}
M1	Number of Polls on Receive	10000	ThreadPoll	{10000}
M1	Tx Throughput Port Arbitr...	Best Effort (Default)	TxBwPrecedence	{0}
M1	Tx Interrupt Moderation Pro...	Moderate	TxIntModerat...	{1}
M1	VLAN ID	0	VlanID	{0}

Name	DisplayName	DisplayValue	RegistryKeyword	RegistryValue
C1	Encapsulated Task Offload	Enabled	*Encapsulate...	{1}
C1	Flow Control	Rx & Tx Enabled	*FlowControl	{3}
C1	IPv4 Checksum Offload	Rx & Tx Enabled	*IPChecksumO...	{3}
C1	Jumbo Packet	1500	*JumboPacket	{1500}
C1	Large Send Offload V2 (IPv4)	Enabled	*LsoV2IPv4	{1}
C1	Large Send Offload V2 (IPv6)	Enabled	*LsoV2IPv6	{1}
C1	Maximum Number of RSS proce...	8	*MaxRssProce...	{8}
C1	NetworkDirect Functionality	Enabled	*NetworkDirect	{1}
C1	Maximum Number of RSS Queues	4	*NumRSSQueues	{4}
C1	Recv Segment Coalescing (IPv4)	Enabled	*RSCIPv4	{1}
C1	Recv Segment Coalescing (IPv6)	Enabled	*RSCIPv6	{1}
C1	Receive Side Scaling	Enabled	*RSS	{1}
C1	RSS Base processor	1	*RssBaseProc...	{1}
C1	RSS load balancing profile	NUMA Scaling Static	*RSSProfile	{4}
C1	TCP Checksum Offload (IPv4)	Rx & Tx Enabled	*TCPChecksum...	{3}
C1	TCP Checksum Offload (IPv6)	Rx & Tx Enabled	*TCPChecksum...	{3}
C1	UDP Checksum Offload (IPv4)	Rx & Tx Enabled	*UDPChecksum...	{3}
C1	UDP Checksum Offload (IPv6)	Rx & Tx Enabled	*UDPChecksum...	{3}
C1	Virtual Machine Queues	Enabled	*VMQ	{1}
C1	VMQ LookAhead Split	Enabled	*VMQLookahea...	{1}
C1	VMQ VLAN ID Filtering	Enabled	*VMQVlanFilt...	{1}
C1	NetworkDirect Interrupt Mod...	Enabled	NDKIntrModer...	{1}
C1	Locally Administered Address	--	NetworkAddress	{--}
C1	Rx Ethernet Queue Size	2048	T4NicResQue...	{2048}
C1	Rx Offload Queue Size	512	T4NicRxOffldQ...	{512}
C1	TCP Offload	Disabled	T4NicTCPOffload	{0}
C1	Tx Offload Queue Size	512	T4NicTxOffldQ...	{512}
C1	Tx Ethernet Queue Size	1024	T4NicTxQueSize	{1024}
C1	User Mode NetworkDirect	Enabled	UserModeNetw...	{1}
C1	VLAN Identifier	0	VlanID	{0}

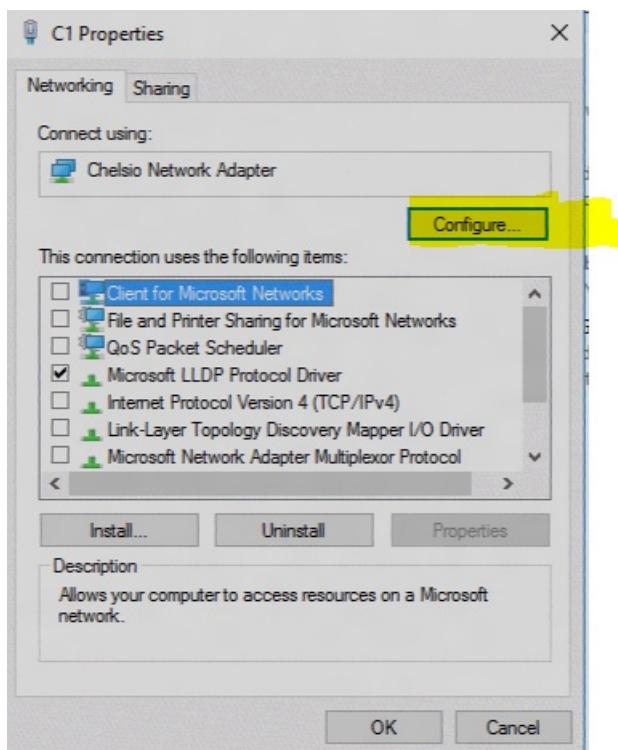
There are similarities and differences in these two NIC Advanced Properties Lists.

2. In the **Network Control Panel** (ncpa.cpl), do the following:

- and right-click the NIC.

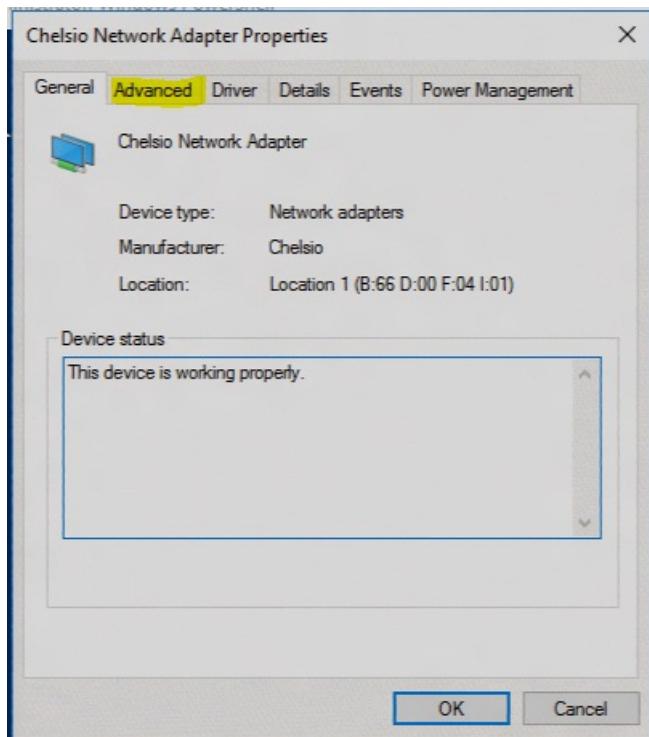


b. In the properties dialog, click **Configure**.



c. Click the **Advanced** tab to view the advanced properties.

The items in this list correlates to the items in the `Get-NetAdapterAdvancedProperties` output.



Insider Preview

9/21/2018 • 2 minutes to read • [Edit Online](#)

Dynamic vRSS and VMMQ

Applies to: Windows Server 2019

In the past, Virtual Machine Queues and Virtual Machine Multi-Queues enabled much higher throughput to individual VMs as network throughputs first reached the 10GbE mark and beyond. Unfortunately, the planning, baselining, tuning, and monitoring required for success became a large undertaking; often more than the IT administrator intended to spend.

Windows Server 2019 improves these optimizations by dynamically spreading and tuning the processing of network workloads as required. Windows Server 2019 ensures peak efficiency and removes the configuration burden for IT administrators.

For more information, see:

- [Announcement blog](#)
- [Validation Guide for the IT Pro](#)

Receive Segment Coalescing (RSC) in the vSwitch

Applies to: Windows Server 2019 and Windows 10, version 1809

Receive Segment Coalescing (RSC) in the vSwitch is an enhancement that coalesces multiple TCP segments into a larger segment before data traversing the vSwitch. The large segment improves networking performance for virtual workloads.

Previously, this was an offload implemented by the NIC. Unfortunately, this was disabled the moment you attached the adapter to a virtual switch. RSC in the vSwitch on Windows Server 2019 and Windows 10 October 2018 Update removes this limitation.

By default, RSC in the vSwitch is enabled on external virtual switches.

For more information, see:

- [Announcement blog](#)
- [Validation Guide for the IT Pro](#)

RSC in the vSwitch

9/21/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server 2019

Receive Segment Coalescing (RSC) in the vSwitch is a feature in Windows Server 2019 and Windows 10 October 2018 Update that helps reduce host CPU utilization and increases throughput for virtual workloads by coalescing multiple TCP segments into fewer, but larger segments. Processing fewer, large segments (coalesced) is more efficient than processing numerous, small segments.

Windows Server 2012 and later included a hardware-only offload version (implemented in the physical network adapter) of the technology also known as Receive Segment Coalescing. This offloaded version of RSC is still available in later versions of Windows. However, it is incompatible with virtual workloads and was disabled once a physical network adapter is attached to a vSwitch. For more information on the hardware-only version of RSC, see [Receive Segment Coalescing \(RSC\)](#).

Scenarios that benefit from RSC in the vSwitch

Workloads whose datapath traverses a virtual switch benefits from this feature.

For example:

- Host Virtual NICs including:
 - Software Defined Networking
 - Hyper-V Host
 - Storage Spaces Direct
- Hyper-V Guest Virtual NICs
- Software Defined Networking GRE Gateways
- Container

Workloads that are not compatible with this feature include:

- Software Defined Networking IPSEC Gateways
- SR-IOV enabled virtual NICs
- SMB Direct

Configure RSC in the vSwitch

By default, on external vSwitches, RSC is enabled.

View the current settings:

```
Get-VMSwitch -Name vSwitchName | Select-Object *RSC*
```

Enable or Disable RSC in the vSwitch

IMPORTANT

Important: RSC in the vSwitch can be enabled and disabled on the fly without impact to existing connections.

Disable RSC in the vSwitch

```
Set-VMSwitch -Name vSwitchName -EnableSoftwareRsc $false
```

Re-enable RSC in the vSwitch

```
Set-VMSwitch -Name vSwitchName -EnableSoftwareRsc $True
```

For more information, see [Set-VMSwitch](#).

Converged Network Interface Card (NIC) configuration guidance

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

Converged network interface card (NIC) allows you to expose RDMA through a host-partition virtual NIC (vNIC) so that the host partition services can access Remote Direct Memory Access (RDMA) on the same NICs that the Hyper-V guests are using for TCP/IP traffic.

Before the Converged NIC feature, management (host partition) services that wanted to use RDMA were required to use dedicated RDMA-capable NICs, even if bandwidth was available on the NICs that were bound to the Hyper-V Virtual Switch.

With Converged NIC, the two workloads (management users of RDMA and Guest traffic) can share the same physical NICs, allowing you to install fewer NICs in your servers.

When you deploy Converged NIC with Windows Server 2016 Hyper-V hosts and Hyper-V Virtual Switches, the vNICs in the Hyper-V hosts expose RDMA services to host processes using RDMA over any Ethernet-based RDMA technology.

NOTE

To use Converged NIC technology, the certified network adapters in your servers must support RDMA.

This guide provides two sets of instructions, one for deployments where your servers have a single network adapter installed, which is a basic deployment of Converged NIC; and another set of instructions where your servers have two or more network adapters installed, which is a deployment of Converged NIC over a Switch Embedded Teaming (SET) team of RDMA-capable network adapters.

Prerequisites

Following are the prerequisites for the Basic and Datacenter deployments of Converged NIC.

NOTE

For the examples provided, we use a Mellanox ConnectX-3 Pro 40 Gbps Ethernet Adapter, but you can use any of the Windows Server certified RDMA-capable network adapters that support this feature. For more information about compatible network adapters, see the Windows Server Catalog topic [LAN Cards](#).

Basic Converged NIC prerequisites

To perform the steps in this guide for basic Converged NIC configuration, you must have the following.

- Two servers that run Windows Server 2016 Datacenter edition or Windows Server 2016 Standard edition.
- One RDMA-capable, certified network adapter installed on each server.
- Hyper-V server role installed on each server.

Datacenter Converged NIC prerequisites

To perform the steps in this guide for datacenter Converged NIC configuration, you must have the following.

- Two servers that run Windows Server 2016 Datacenter edition or Windows Server 2016 Standard edition.
- Two RDMA-capable, certified network adapters installed on each server.
- Hyper-V server role installed on each server.
- You must be familiar with Switch Embedded Teaming (SET), an alternative NIC Teaming solution used in environments that include Hyper-V and the Software Defined Networking (SDN) stack in Windows Server 2016. SET integrates some NIC Teaming functionality into the Hyper-V Virtual Switch. For more information, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#).

Related topics

- [Converged NIC Configuration with a Single Network Adapter](#)
 - [Converged NIC Teamed NIC Configuration](#)
 - [Physical Switch Configuration for Converged NIC](#)
 - [Troubleshooting Converged NIC Configurations](#)
-

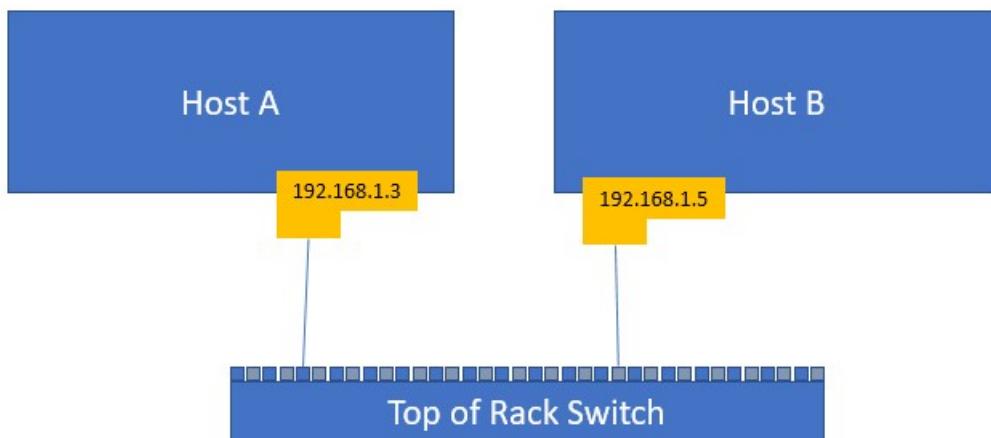
Converged NIC configuration with a single network adapter

9/21/2018 • 11 minutes to read • [Edit Online](#)

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

In this topic, we provide you with the instructions to configure Converged NIC with a single NIC in your Hyper-V host.

The example configuration in this topic describes two Hyper-V hosts, **Hyper-V Host A**, and **Hyper-V Host B**. Both hosts have a single physical NIC (pNIC) installed, and the NICs are connected to a top of rack (ToR) physical switch. In addition, the hosts are located on the same subnet, which is 192.168.1.x/24.



Step 1. Test the connectivity between source and destination

Ensure that the physical NIC can connect to the destination host. This test demonstrates connectivity by using Layer 3 (L3) - or the IP layer - as well as Layer 2 (L2).

1. View the network adapter properties.

```
Get-NetAdapter
```

Results:

NAME	INTERFACEDESCRIPTION	IFINDEX	STATUS	MACADDRESS	LINKSPEED
M1	Mellanox ConnectX-3 Pro ...	4	Up	7C-FE-90-93-8F-A1	40 Gbps

2. View additional adapter properties, including the IP address.

```
Get-NetAdapter M1 | fl *
```

Results:

```
MacAddress      : 7C-FE-90-93-8F-A1
Status         : Up
LinkSpeed: 40 Gbps
MediaType: 802.3
PhysicalMediaType: 802.3
AdminStatus    : Up
MediaConnectionState : Connected
DriverInformation: Driver Date 2016-08-28 Version 5.25.12665.0 NDIS 6.60
DriverFileName  : mlx4eth63.sys
NdisVersion    : 6.60
ifOperStatus   : Up
ifAlias        : M1
InterfaceAlias : M1
ifIndex        : 4
ifDesc          : Mellanox ConnectX-3 Pro Ethernet Adapter
ifName          : ethernet_32773
DriverVersion: 5.25.12665.0
LinkLayerAddress : 7C-FE-90-93-8F-A1
Caption        :
Description     :
ElementName    :
InstanceID     : {39B58B4C-8833-4ED2-A2FD-E105E7146D43}
CommunicationStatus :
DetailedStatus  :
HealthState    :
InstallDate   :
Name : M1
OperatingStatus :
OperationalStatus:
PrimaryStatus:
StatusDescriptions :
AvailableRequestedStates :
EnabledDefault  : 2
EnabledState    : 5
OtherEnabledState:
RequestedState  : 12
TimeOfLastStateChange:
TransitioningToState : 12
AdditionalAvailability  :
Availability   :
CreationClassName: MSFT_NetAdapter
```

Step 2. Ensure that source and destination can communicate

In this step, we use the **Test-NetConnection** Windows PowerShell command, but if you can use the **ping** command if you prefer.

TIP

If you're certain that your hosts can communicate with each other, you can skip this step.

1. Verify bi-directional communication.

```
Test-NetConnection 192.168.1.5
```

Results:

PARAMETER	VALUE
ComputerName	192.168.1.5
RemoteAddress	192.168.1.5
InterfaceAlias	M1
SourceAddress	192.168.1.3
PingSucceeded	True
PingReplyDetails (RTT)	0 ms

In some cases, you might need to disable Windows Firewall with Advanced Security to successfully perform this test. If you disable the firewall, keep security in mind and ensure that your configuration meets your organization's security requirements.

2. Disable all firewall profiles.

```
Set-NetFirewallProfile -All -Enabled False
```

3. After disabling the firewall profiles, test the connection again.

```
Test-NetConnection 192.168.1.5
```

Results:

PARAMETER	VALUE
ComputerName	192.168.1.5
RemoteAddress	192.168.1.5
InterfaceAlias	Test-40G-1
SourceAddress	192.168.1.3
PingSucceeded	False
PingReplyDetails (RTT)	0 ms

Step 3. (Optional) Configure the VLAN IDs for NICs installed in your Hyper-V hosts

Many network configurations make use of VLANs, and if you are planning to use VLANs in your network, you must repeat the previous test with VLANs configured. Also, if you are planning to use RoCE for RDMA services you must enable VLANs.

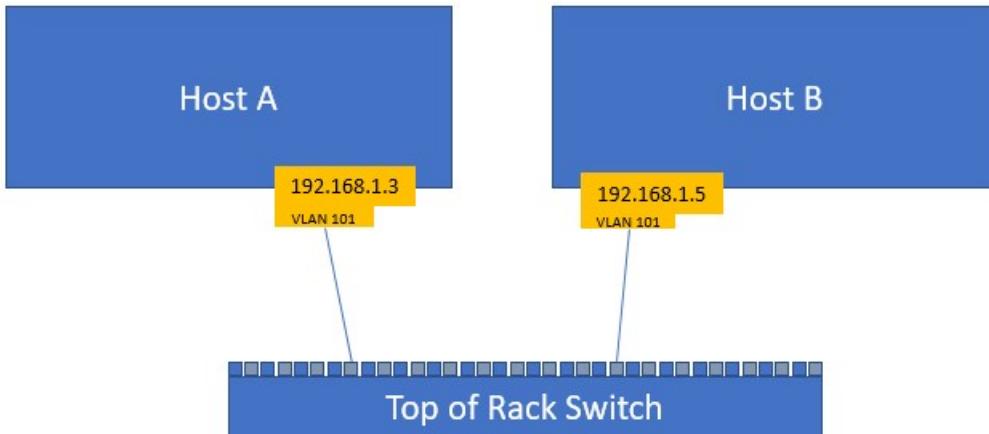
For this step, the NICs are in **ACCESS** mode. However, when you create a Hyper-V Virtual Switch (vSwitch) later in this guide, the VLAN properties are applied at the vSwitch port level.

Because a switch can host multiple VLANs, it is necessary for the Top of Rack (ToR) physical switch to have the port that the host is connected to configured in Trunk mode.

NOTE

Consult your ToR switch documentation for instructions on how to configure Trunk mode on the switch.

The following image shows two Hyper-V hosts, each with one physical network adapter, and each configured to communicate on VLAN 101.



IMPORTANT

Perform this on both the local and destination servers. If the destination server is not configured with the same VLAN ID as the local server, the two cannot communicate.

1. Configure the VLAN ID for NICs installed in your Hyper-V hosts.

IMPORTANT

Do not run this command if you are connected to the host remotely over this interface, because this results in loss of access to the host.

```
Set-NetAdapterAdvancedProperty -Name M1 -RegistryKeyword VlanID -RegistryValue "101"
Get-NetAdapterAdvancedProperty -Name M1 | Where-Object {$_._RegistryKeyword -eq "VlanID"}
```

Results:

NAME	DISPLAYNAME	DISPLAYVALUE	REGISTRYKEYWORD	REGISTRYVALUE
M1	VLAN ID	101	VlanID	{101}

2. Restart the network adapter to apply the VLAN ID.

```
Restart-NetAdapter -Name "M1"
```

3. Ensure the Status is **Up**.

```
Get-NetAdapter -Name "M1"
```

Results:

NAME	INTERFACEDESCRIPTION	IFINDEX	STATUS	MACADDRESS	LINKSPEED
M1	Mellanox ConnectX-3 Pro Ethernet Adapter	4	Up	7C-FE-90-93-8F-A1	40 Gbps

IMPORTANT

It might take several seconds for the device to restart and become available on the network.

4. Verify the connectivity.

If connectivity fails, inspect the switch VLAN configuration or destination participation in the same VLAN.

```
Test-NetConnection 192.168.1.5
```

Step 4. Configure Quality of Service (QoS)

NOTE

You must perform all of the following DCB and QoS configuration steps on all hosts that are intended to communicate with each other.

1. Install Data Center Bridging (DCB) on each of your Hyper-V hosts.

- **Optional** for network configurations that use iWarp for RDMA services.
- **Required** for network configurations that use RoCE (any version) for RDMA services.

```
Install-WindowsFeature Data-Center-Bridging
```

2. Set the QoS policies for SMB-Direct:

- **Optional** for network configurations that use iWarp.
- **Required** for network configurations that use RoCE.

In the example command below, the value "3" is arbitrary. You can use any value between 1 and 7 as long as you consistently use the same value throughout the configuration of QoS policies.

```
New-NetQosPolicy "SMB" -NetDirectPortMatchCondition 445 -PriorityValue8021Action 3
```

Results:

PARAMETER	VALUE
Name	SMB
Owner	Group Policy (Machine)
NetworkProfile	All

PARAMETER	VALUE
Precedence	127
JobObject	
NetDirectPort	445
PriorityValue	3

3. For RoCE deployments, turn on **Priority Flow Control** for SMB traffic, which is not required for iWarp.

```
Enable-NetQosFlowControl -priority 3
Get-NetQosFlowControl
```

Results:

PRIORITY	ENABLED	POLICYSET	IFINDEX	IFALIAS
0	False	Global		
1	False	Global		
2	False	Global		
3	True	Global		
4	False	Global		
5	False	Global		
6	False	Global		
7	False	Global		

4. Enable QoS for the local and destination network adapters.

- **Not needed** for network configurations that use iWarp.
- **Required** for network configurations that use RoCE.

```
Enable-NetAdapterQos -InterfaceAlias "M1"
Get-NetAdapterQos -Name "M1"
```

Results:

Name: M1

Enabled: True

Capabilities:

PARAMETER	HARDWARE	CURRENT
MacSecBypass	NotSupported	NotSupported
DcbxSupport	None	None
NumTCs(Max/ETS/PFC)	8/8/8	8/8/8

OperationalTrafficClasses:

TC	TSA	BANDWIDTH	PRIORITIES
0	ETS	70%	0-2,4-7
1	ETS	30%	3

OperationalFlowControl:

Priority 3 Enabled

OperationalClassifications:

PROTOCOL	PORT/TYPE	PRIORITY
Default		0
NetDirect	445	3

- Reserve a percentage of the bandwidth for SMB Direct (RDMA).

In this example, a 30% bandwidth reservation is used. You should select a value that represents what you expect your storage traffic requires.

```
New-NetQosTrafficClass "SMB" -Priority 3 -BandwidthPercentage 30 -Algorithm ETS
```

Results:

NAME	ALGORITHM	BANDWIDTH(%)	PRIORITY	POLICYSET	IFINDEX	IFALIAS
SMB	ETS	30	3	Global		

- View the bandwidth reservation settings.

```
Get-NetQosTrafficClass
```

Results:

NAME	ALGORITHM	BANDWIDTH(%)	PRIORITY	POLICYSET	IFINDEX	IFALIAS
[Default]	ETS	70	0-2,4-7	Global		

NAME	ALGORITHM	BANDWIDTH(%)	PRIORITY	POLICYSET	IFINDEX	IFALIAS
SMB	ETS	30	3	Global		

Step 5. (Optional) Resolve the Mellanox adapter debugger conflict

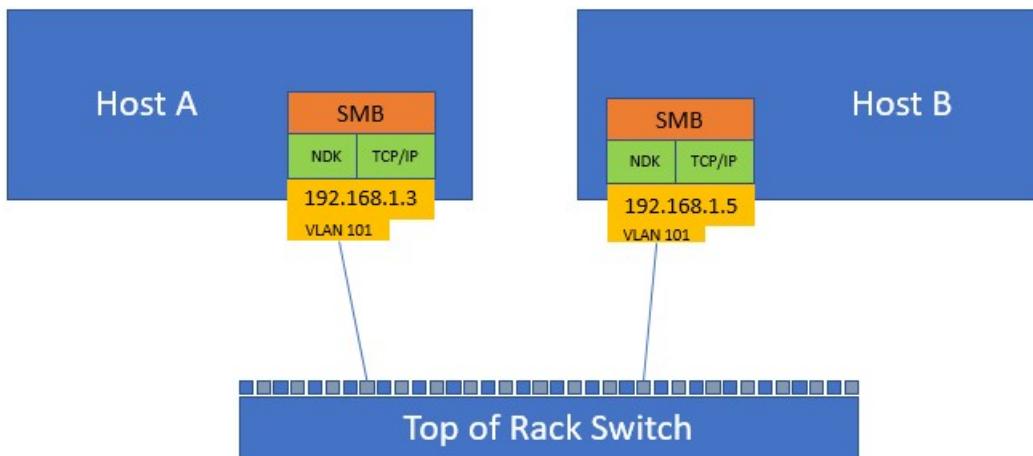
By default, when using a Mellanox adapter, the attached debugger blocks NetQoS, which is a known issue. Therefore, if you are using an adapter from Mellanox and intend to attach a debugger, use the following command to resolve this issue. This step is not required if you do not intend to attach a debugger or if you're not using a Mellanox adapter.

```
Set-ItemProperty HKLM:"\SYSTEM\CurrentControlSet\Services\NDIS\Parameters" AllowFlowControlUnderDebugger -type DWORD -Value 1 -Force
```

Step 6. Verify the RDMA configuration (Native host)

You want to ensure that the fabric is configured correctly prior to creating a vSwitch and transitioning to RDMA (Converged NIC).

The following image shows the current state of the Hyper-V hosts.



1. Verify the RDMA configuration.

```
Get-NetAdapterRdma
```

Results:

NAME	INTERFACEDESCRIPTION	ENABLED
M1	Mellanox ConnectX-3 Pro Ethernet Adapter	True

2. Determine the **ifIndex** value of your target adapter.

You use this value in subsequent steps when you run the script you download.

```
Get-NetIPConfiguration -InterfaceAlias "M*" | ft InterfaceAlias,InterfaceIndex,IPv4Address
```

Results:

INTERFACEALIAS	INTERFACEINDEX	IPV4ADDRESS
M2	14	{192.168.1.5}

3. Download the [DiskSpd.exe utility](#) and extract it into C:\TEST.
4. Download the [Test-RDMA powershell script](#) to a test folder on your local drive, for example, C:\TEST.
5. Run the **Test-Rdma.ps1** PowerShell script to pass the ifIndex value to the script, along with the IP address of the remote adapter on the same VLAN.

In this example, the script passes the **ifIndex** value of 14 on the remote network adapter IP address 192.168.1.5.

```
C:\TEST\Test-RDMA.PS1 -IfIndex 14 -IsRoCE $true -RemoteIpAddress 192.168.1.5 -PathToDiskspd  
C:\TEST\Diskspd-v2.0.17\amd64fre\  
  
VERBOSE: Diskspd.exe found at C:\TEST\Diskspd-v2.0.17\amd64fre\diskspd.exe  
VERBOSE: The adapter M2 is a physical adapter  
VERBOSE: Underlying adapter is RoCE. Checking if QoS/DCB/PFC is configured on each physical adapter(s)  
VERBOSE: QoS/DCB/PFC configuration is correct.  
VERBOSE: RDMA configuration is correct.  
VERBOSE: Checking if remote IP address, 192.168.1.5, is reachable.  
VERBOSE: Remote IP 192.168.1.5 is reachable.  
VERBOSE: Disabling RDMA on adapters that are not part of this test. RDMA will be enabled on them  
later.  
VERBOSE: Testing RDMA traffic now for. Traffic will be sent in a parallel job. Job details:  
VERBOSE: 0 RDMA bytes written per second  
VERBOSE: 0 RDMA bytes sent per second  
VERBOSE: 662979201 RDMA bytes written per second  
VERBOSE: 37561021 RDMA bytes sent per second  
VERBOSE: 1023098948 RDMA bytes written per second  
VERBOSE: 8901349 RDMA bytes sent per second  
VERBOSE: Enabling RDMA on adapters that are not part of this test. RDMA was disabled on them prior to  
sending RDMA traffic.  
VERBOSE: RDMA traffic test SUCCESSFUL: RDMA traffic was sent to 192.168.1.5
```

NOTE

If the RDMA traffic fails, for the RoCE case specifically, consult your ToR Switch configuration for proper PFC/ETS settings that should match the Host settings. Refer to the QoS section in this document for reference values.

Step 7. Remove the Access VLAN setting

In preparation for creating the Hyper-V switch, you must remove the VLAN settings you installed above.

1. Remove the ACCESS VLAN setting from the physical NIC to prevent the NIC from auto-tagging the egress traffic with the incorrect VLAN ID.

Removing this setting also prevents it from filtering ingress traffic that doesn't match the ACCESS VLAN ID.

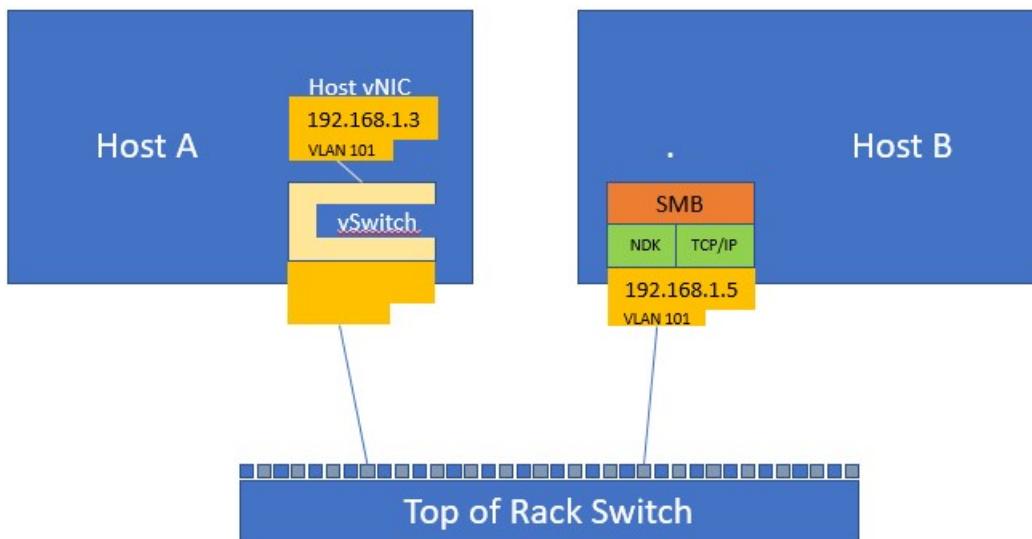
```
Set-NetAdapterAdvancedProperty -Name M1 -RegistryKeyword VlanID -RegistryValue "0"
```

2. Confirm that the **VlanID setting** shows the VLAN ID value as zero.

```
Get-NetAdapterAdvancedProperty -name m1 | Where-Object {$_._RegistryKeyword -eq 'VlanID'}
```

Step 8. Create a Hyper-V vSwitch on your Hyper-V hosts

The following image depicts Hyper-V Host 1 with a vSwitch.



1. Create an external Hyper-V vSwitch in Hyper-V on Hyper-V Host A.

In this example, the switch is named VMTEST. Also, the parameter **AllowManagementOS** creates a host vNIC that inherits the MAC and IP addresses of the physical NIC.

```
New-VMSwitch -Name VMTEST -NetAdapterName "M1" -AllowManagementOS $true
```

Results:

NAME	SWITCHTYPE	NETADAPTERINTERFACEDESCRIPTION
VMTEST	External	Mellanox ConnectX-3 Pro Ethernet Adapter

2. View the network adapter properties.

```
Get-NetAdapter | ft -AutoSize
```

Results:

NAME	INTERFACEDESCRIPTION	IFINDEX	STATUS	MACADDRESS	LINKSPEED
vEthernet (VMTEST)	Hyper-V Virtual Ethernet Adapter #2	27	Up	E4-1D-2D-07-40-71	40 Gbps

3. Manage the host vNIC in one of two ways.

- **NetAdapter** view operates based upon the "vEthernet (VMTEST)" name. Not all network adapter properties display in this view.
- **VMNetworkAdapter** view drops the "vEthernet" prefix and simply uses the vmswitch name.

(Recommended)

```
Get-VMNetworkAdapter -ManagementOS | ft -AutoSize
```

Results:

NAME	ISMANAGEMEN TOS	VMNAME	SWITCHNAME	MACADDRESS	STATUS	IPADDRESSES
CORP-External-Switch	True	CORP-External-Switch	001B785768AA	{Ok}		
VMSTEST	True	VMSTEST	E41D2D074071	{Ok}		

4. Test the connection.

```
Test-NetConnection 192.168.1.5
```

Results:

```
ComputerName    : 192.168.1.5
RemoteAddress   : 192.168.1.5
InterfaceAlias  : vEthernet (CORP-External-Switch)
SourceAddress   : 192.168.1.3
PingSucceeded   : True
PingReplyDetails (RTT) : 0 ms
```

5. Assign and view the network adapter VLAN settings.

```
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "VMSTEST" -VlanId "101" -Access -ManagementOS
Get-VMNetworkAdapterVlan -ManagementOS -VMNetworkAdapterName "VMSTEST"
```

Results:

VMNAME	VMNETWORKADAPTERNAME	MODE	VLANLIST
	VMSTEST	Access	101

6. Test the connection.

It may take a few seconds to complete before you can successfully ping the other adapter.

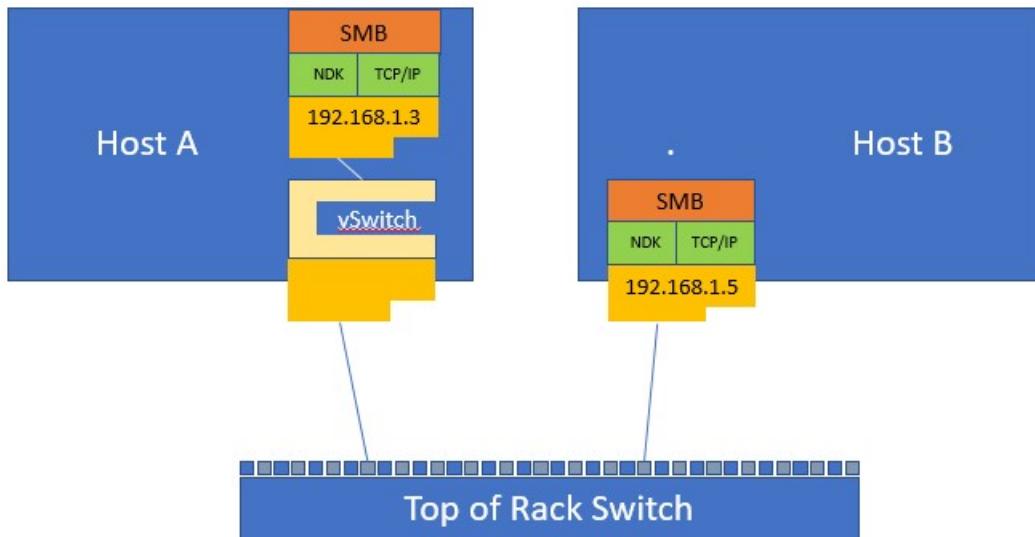
```
Test-NetConnection 192.168.1.5
```

Results:

```
ComputerName      : 192.168.1.5
RemoteAddress    : 192.168.1.5
InterfaceAlias   : vEthernet (VMSTEST)
SourceAddress    : 192.168.1.3
PingSucceeded    : True
PingReplyDetails (RTT) : 0 ms
```

Step 9. Test Hyper-V Virtual Switch RDMA (Mode 2)

The following image depicts the current state of your Hyper-V hosts, including the vSwitch on Hyper-V Host 1.



1. Set the priority tagging on the host vNIC.

```
Set-VMNetworkAdapter -ManagementOS -Name "VMSTEST" -IeeePriorityTag on
Get-VMNetworkAdapter -ManagementOS -Name "VMSTEST" | fl Name,IeeePriorityTag
```

Results:

Name: VMSTEST IeeePriorityTag : On

2. View the network adapter RDMA information.

```
Get-NetAdapterRdma
```

Results:

NAME	INTERFACEDESCRIPTION	ENABLED
vEthernet (VMSTEST)	Hyper-V Virtual Ethernet Adapter #2	False

NOTE

If the parameter **Enabled** has the value **False**, it means that RDMA is not enabled.

3. View the network adapter information.

```
Get-NetAdapter
```

Results:

NAME	INTERFACEDESCRIPTION	IFINDEX	STATUS	MACADDRESS	LINKSPEED
vEthernet (VMTEST)	Hyper-V Virtual Ethernet Adapter #2	27	Up	E4-1D-2D-07-40-71	40 Gbps

4. Enable RDMA on the host vNIC.

```
Enable-NetAdapterRdma -Name "vEthernet (VMTEST)"
Get-NetAdapterRdma -Name "vEthernet (VMTEST)"
```

Results:

NAME	INTERFACEDESCRIPTION	ENABLED
vEthernet (VMTEST)	Hyper-V Virtual Ethernet Adapter #2	True

NOTE

If the parameter **Enabled** has the value **True**, it means that RDMA is enabled.

5. Perform RDMA traffic test.

```
C:\TEST\Test-RDMA.ps1 -IfIndex 27 -IsRoCE $true -RemoteIpAddress 192.168.1.5 -PathToDiskspd
C:\TEST\Diskspd-v2.0.17\amd64fre\
VERBOSE: Diskspd.exe found at C:\TEST\Diskspd-v2.0.17\amd64fre\diskspd.exe
VERBOSE: The adapter vEthernet (VMTEST) is a virtual adapter
VERBOSE: Retrieving vSwitch bound to the virtual adapter
VERBOSE: Found vSwitch: VMTEST
VERBOSE: Found the following physical adapter(s) bound to vSwitch: TEST-40G-1
VERBOSE: Underlying adapter is RoCE. Checking if QoS/DCB/PFC is configured on each physical adapter(s)
VERBOSE: QoS/DCB/PFC configuration is correct.
VERBOSE: RDMA configuration is correct.
VERBOSE: Checking if remote IP address, 192.168.1.5, is reachable.
VERBOSE: Remote IP 192.168.1.5 is reachable.
VERBOSE: Disabling RDMA on adapters that are not part of this test. RDMA will be enabled on them
later.
VERBOSE: Testing RDMA traffic now for. Traffic will be sent in a parallel job. Job details:
VERBOSE: 0 RDMA bytes written per second
VERBOSE: 0 RDMA bytes sent per second
VERBOSE: 0 RDMA bytes written per second
VERBOSE: 0 RDMA bytes sent per second
VERBOSE: 0 RDMA bytes written per second
VERBOSE: 0 RDMA bytes sent per second
VERBOSE: 0 RDMA bytes written per second
VERBOSE: 0 RDMA bytes sent per second
VERBOSE: 9162492 RDMA bytes sent per second
VERBOSE: 938797258 RDMA bytes written per second
VERBOSE: 34621865 RDMA bytes sent per second
VERBOSE: 933572610 RDMA bytes written per second
VERBOSE: 35035861 RDMA bytes sent per second
VERBOSE: Enabling RDMA on adapters that are not part of this test. RDMA was disabled on them prior to
sending RDMA traffic.
VERBOSE: RDMA traffic test SUCCESSFUL: RDMA traffic was sent to 192.168.1.5
```

The final line in this output, "RDMA traffic test SUCCESSFUL: RDMA traffic was sent to 192.168.1.5," shows that

you have successfully configured Converged NIC on your adapter.

Related topics

- [Converged NIC Teamed NIC Configuration](#)
- [Physical Switch Configuration for Converged NIC](#)
- [Troubleshooting Converged NIC Configurations](#)

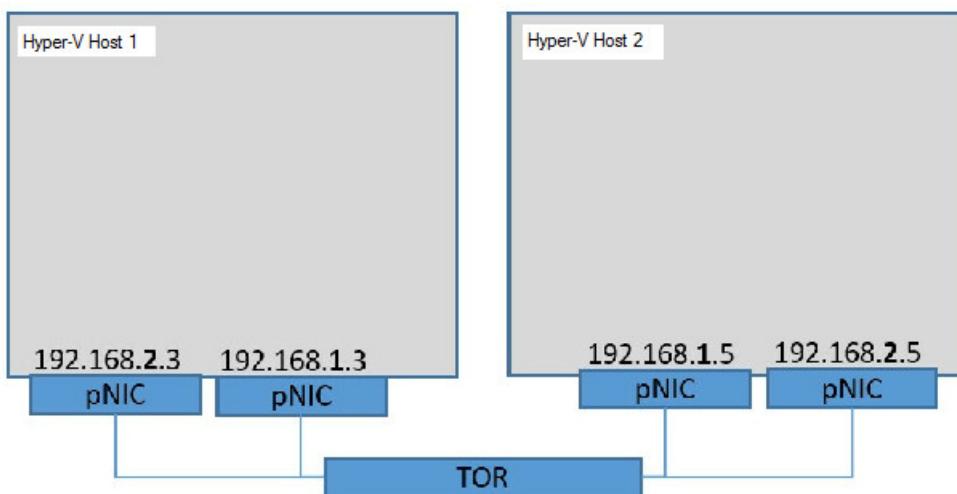
Converged NIC in a Teamed NIC configuration (datacenter)

9/18/2018 • 23 minutes to read • [Edit Online](#)

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

In this topic, we provide you with instructions to deploy Converged NIC in a Teamed NIC configuration with Switch Embedded Teaming (SET).

The example configuration in this topic describes two Hyper-V hosts, **Hyper-V Host 1** and **Hyper-V Host 2**. Both hosts have two network adapters. On each host, one adapter is connected to the 192.168.1.x/24 subnet, and one adapter is connected to the 192.168.2.x/24 subnet.



Step 1. Test the connectivity between source and destination

Ensure that the physical NIC can connect to the destination host. This test demonstrates connectivity by using Layer 3 (L3) - or the IP layer - as well as the Layer 2 (L2) virtual local area networks (VLAN).

1. View the first network adapter properties.

```
Get-NetAdapter -Name "Test-40G-1" | ft -AutoSize
```

Results:

NAME	INTERFACEDESCRIPTION	IFINDEX	STATUS	MACADDRESS	LINKSPEED
Test-40G-1	Mellanox ConnectX-3 Pro Ethernet Adapter	11	Up	E4-1D-2D-07-43-D0	40 Gbps

2. View additional properties for the first adapter, including the IP address.

```
Get-NetIPAddress -InterfaceAlias "Test-40G-1"
Get-NetIPAddress -InterfaceAlias "TEST-40G-1" | Where-Object {$_._AddressFamily -eq "IPv4"} | fl
InterfaceAlias,IPAddress
```

Results:

PARAMETER	VALUE
IPAddress	192.168.1.3
InterfaceIndex	11
InterfaceAlias	Test-40G-1
AddressFamily	IPv4
Type	Unicast
PrefixLength	24

3. View the second network adapter properties.

```
Get-NetAdapter -Name "Test-40G-2" | ft -AutoSize
```

Results:

NAME	INTERFACEDESCRIPTION	IFINDEX	STATUS	MACADDRESS	LINKSPEED
TEST-40G-2	Mellanox ConnectX-3 Pro Ethernet A...#2	13	Up	E4-1D-2D-07-40-70	40 Gbps

4. View additional properties for the second adapter, including the IP address.

```
Get-NetIPAddress -InterfaceAlias "Test-40G-2"
Get-NetIPAddress -InterfaceAlias "Test-40G-2" | Where-Object {$_._AddressFamily -eq "IPv4"} | fl
InterfaceAlias,IPAddress
```

Results:

PARAMETER	VALUE
IPAddress	192.168.2.3
InterfaceIndex	13
InterfaceAlias	TEST-40G-2
AddressFamily	IPv4
Type	Unicast

PARAMETER	VALUE
PrefixLength	24

- Verify that other NIC Team or SET member pNICs has a valid IP address.

Use a separate subnet, (xxx.xxx.**2**.xxx vs xxx.xxx.**1**.xxx), to facilitate sending from this adapter to the destination. Otherwise, if you locate both pNICs on the same subnet, the Windows TCP/IP stack load balances among the interfaces and simple validation becomes more complicated.

Step 2. Ensure that source and destination can communicate

In this step, we use the **Test-NetConnection** Windows PowerShell command, but if you can use the **ping** command if you prefer.

- Verify bi-directional communication.

```
Test-NetConnection 192.168.1.5
```

Results:

PARAMETER	VALUE
ComputerName	192.168.1.5
RemoteAddress	192.168.1.5
InterfaceAlias	Test-40G-1
SourceAddress	192.168.1.3
PingSucceeded	False
PingReplyDetails (RTT)	0 ms

In some cases, you might need to disable Windows Firewall with Advanced Security to successfully perform this test. If you disable the firewall, keep security in mind and ensure that your configuration meets your organization's security requirements.

- Disable all firewall profiles.

```
Set-NetFirewallProfile -All -Enabled False
```

- After disabling the firewall profiles, test the connection again.

```
Test-NetConnection 192.168.1.5
```

Results:

PARAMETER	VALUE
ComputerName	192.168.1.5
RemoteAddress	192.168.1.5
InterfaceAlias	Test-40G-1
SourceAddress	192.168.1.3
PingSucceeded	False
PingReplyDetails (RTT)	0 ms

4. Verify the connectivity for additional NICs. Repeat the previous steps for all subsequent pNICs included in the NIC or SET team.

```
Test-NetConnection 192.168.2.5
```

Results:

PARAMETER	VALUE
ComputerName	192.168.2.5
RemoteAddress	192.168.2.5
InterfaceAlias	Test-40G-2
SourceAddress	192.168.2.3
PingSucceeded	False
PingReplyDetails (RTT)	0 ms

Step 3. Configure the VLAN IDs for NICs installed in your Hyper-V hosts

Many network configurations make use of VLANs, and if you are planning to use VLANs in your network, you must repeat the previous test with VLANs configured.

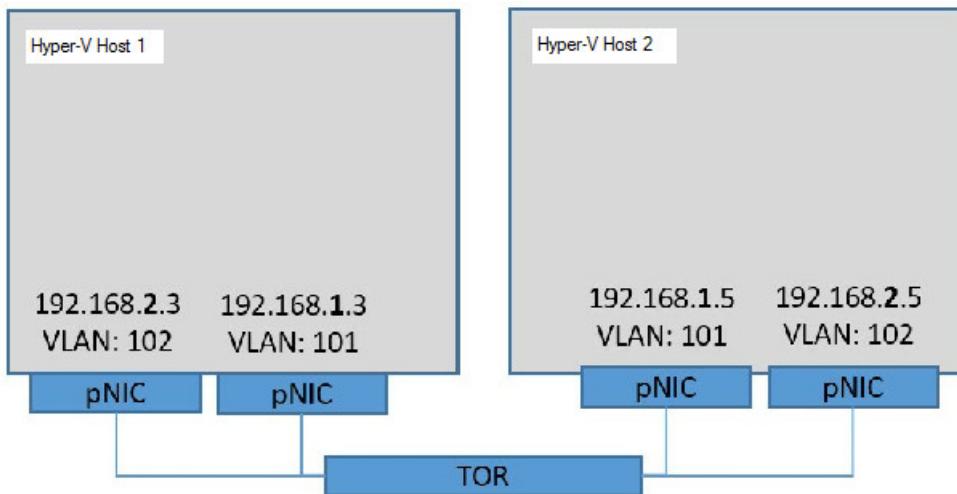
For this step, the NICs are in **ACCESS** mode. However, when you create a Hyper-V Virtual Switch (vSwitch) later in this guide, the VLAN properties are applied at the vSwitch port level.

Because a switch can host multiple VLANs, it is necessary for the Top of Rack (ToR) physical switch to have the port that the host is connected to configured in Trunk mode.

NOTE

Consult your ToR switch documentation for instructions on how to configure Trunk mode on the switch.

The following image shows two Hyper-V hosts with two network adapters each that have VLAN 101 and VLAN 102 configured in network adapter properties.



TIP

According to the Institute of Electrical and Electronics Engineers (IEEE) networking standards, the Quality of Service (QoS) properties in the physical NIC act on the 802.1p header that is embedded within the 802.1Q (VLAN) header when you configure the VLAN ID.

1. Configure the VLAN ID on the first NIC, Test-40G-1.

```
Set-NetAdapterAdvancedProperty -Name "Test-40G-1" -RegistryKeyword VlanID -RegistryValue "101"
Get-NetAdapterAdvancedProperty -Name "Test-40G-1" | Where-Object {$_.RegistryKeyword -eq "VlanID"} | ft
-AutoSize
```

Results:

NAME	DISPLAYNAME	DISPLAYVALUE	REGISTRYKEYWORD	REGISTRYVALUE
TEST-40G-1	VLAN ID	101	VlanID	{101}

2. Restart the network adapter to apply the VLAN ID.

```
Restart-NetAdapter -Name "Test-40G-1"
```

3. Ensure the Status is **Up**.

```
Get-NetAdapter -Name "Test-40G-1" | ft -AutoSize
```

Results:

NAME	INTERFACEDESCRIPTION	IFINDEX	STATUS	MACADDRESS	LINKSPEED
Test-40G-1	Mellanox ConnectX-3 Pro Ethernet Ada...	11	Up	E4-1D-2D-07-43-D0	40 Gbps

4. Configure the VLAN ID on the second NIC, Test-40G-2.

```
Set-NetAdapterAdvancedProperty -Name "Test-40G-2" -RegistryKeyword VlanID -RegistryValue "102"
Get-NetAdapterAdvancedProperty -Name "Test-40G-2" | Where-Object {$_._RegistryKeyword -eq "VlanID"} | ft
-AutoSize
```

Results:

NAME	DISPLAYNAME	DISPLAYVALUE	REGISTRYKEYWORD	REGISTRYVALUE
TEST-40G-2	VLAN ID	102	VlanID	{102}

5. Restart the network adapter to apply the VLAN ID.

```
Restart-NetAdapter -Name "Test-40G-2"
```

6. Ensure the Status is **Up**.

```
Get-NetAdapter -Name "Test-40G-1" | ft -AutoSize
```

Results:

NAME	INTERFACEDESCRIPTION	IFINDEX	STATUS	MACADDRESS	LINKSPEED
Test-40G-2	Mellanox ConnectX-3 Pro Ethernet Ada...	11	Up	E4-1D-2D-07-43-D1	40 Gbps

IMPORTANT

It might take several seconds for the device to restart and become available on the network.

7. Verify the connectivity for the first NIC, Test-40G-1.

If connectivity fails, inspect the switch VLAN configuration or destination participation in the same VLAN.

```
Test-NetConnection 192.168.1.5
```

Results:

PARAMETER	VALUE
ComputerName	192.168.1.5
RemoteAddress	192.168.1.5
InterfaceAlias	Test-40G-1
SourceAddress	192.168.1.5

PARAMETER	VALUE
PingSucceeded	True
PingReplyDetails (RTT)	0 ms

8. Verify the connectivity for the first NIC, Test-40G-2.

If connectivity fails, inspect the switch VLAN configuration or destination participation in the same VLAN.

```
Test-NetConnection 192.168.2.5
```

Results:

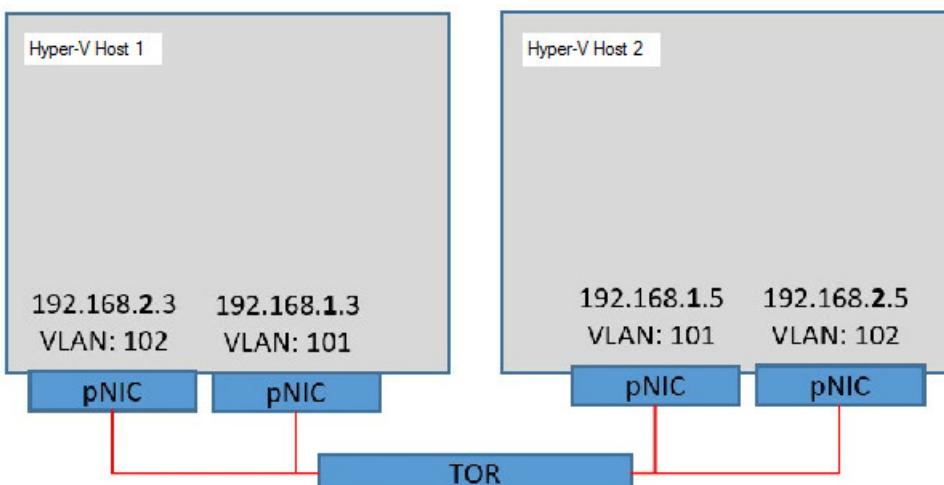
PARAMETER	VALUE
ComputerName	192.168.2.5
RemoteAddress	192.168.2.5
InterfaceAlias	Test-40G-2
SourceAddress	192.168.2.3
PingSucceeded	True
PingReplyDetails (RTT)	0 ms

IMPORTANT

It's not uncommon for a **Test-NetConnection** or ping failure to occur immediately after you perform **Restart-NetAdapter**. So wait for the network adapter to fully initialize, and then try again.

If the VLAN 101 connections work, but the VLAN 102 connections don't, the problem might be that the switch needs to be configured to allow port traffic on the desired VLAN. You can check for this by temporarily setting the failing adapters to VLAN 101, and repeating the connectivity test.

The following image shows your Hyper-V hosts after successfully configuring VLANs.



Step 4. Configure Quality of Service (QoS)

NOTE

You must perform all of the following DCB and QoS configuration steps on all hosts that are intended to communicate with each other.

1. Install Data Center Bridging (DCB) on each of your Hyper-V hosts.

- **Optional** for network configurations that use iWarp.
- **Required** for network configurations that use RoCE (any version) for RDMA services.

```
Install-WindowsFeature Data-Center-Bridging
```

Results:

SUCCESS	RESTART NEEDED	EXIT CODE	FEATURE RESULT
True	No	Success	{Data Center Bridging}

2. Set the QoS policies for SMB-Direct:

- **Optional** for network configurations that use iWarp.
- **Required** for network configurations that use RoCE (any version) for RDMA services.

In the example command below, the value "3" is arbitrary. You can use any value between 1 and 7 as long as you consistently use the same value throughout the configuration of QoS policies.

```
New-NetQosPolicy "SMB" -NetDirectPortMatchCondition 445 -PriorityValue8021Action 3
```

Results:

PARAMETER	VALUE
Name	SMB
Owner	Group Policy (Machine)
NetworkProfile	All
Precedence	127
JobObject	
NetDirectPort	445
PriorityValue	3

3. Set additional QoS policies for other traffic on the interface.

```
New-NetQosPolicy "DEFAULT" -Default -PriorityValue8021Action 0
```

Results:

PARAMETER	VALUE
Name	DEFAULT
Owner	Group Policy (Machine)
NetworkProfile	All
Precedence	127
Template	Default
JobObject	
PriorityValue	0

4. Turn on **Priority Flow Control** for SMB traffic, which is not required for iWarp.

```
Enable-NetQosFlowControl -priority 3  
Get-NetQosFlowControl
```

Results:

PRIORITY	ENABLED	POLICYSET	IFINDEX	IFALIAS
0	False	Global		
1	False	Global		
2	False	Global		
3	True	Global		
4	False	Global		
5	False	Global		
6	False	Global		
7	False	Global		

IMPORTANT If your results do not match these results because items other than 3 have an Enabled value of True, you must disable **FlowControl** for these classes.

```
Disable-NetQosFlowControl -priority 0,1,2,4,5,6,7  
Get-NetQosFlowControl
```

Under more complex configurations, the other traffic classes might require flow control, however these scenarios are outside the scope of this guide.

5. Enable QoS for the first NIC, Test-40G-1.

```
Enable-NetAdapterQos -InterfaceAlias "Test-40G-1"
Get-NetAdapterQos -Name "Test-40G-1"

Name: TEST-40G-1
Enabled: True
```

Capabilities:

PARAMETER	HARDWARE	CURRENT
MacSecBypass	NotSupported	NotSupported
DcbxSupport	None	None
NumTCs(Max/ETS/PFC)	8/8/8	8/8/8

OperationalTrafficClasses:

TC	TSA	BANDWIDTH	PRIORITIES
0	Strict		0-7

OperationalFlowControl:

Priority 3 Enabled

OperationalClassifications:

PROTOCOL	PORT/TYPE	PRIORITY
Default		0
NetDirect	445	3

6. Enable QoS for the second NIC, Test-40G-2.

```
Enable-NetAdapterQos -InterfaceAlias "Test-40G-2"
Get-NetAdapterQos -Name "Test-40G-2"

Name: TEST-40G-2
Enabled: True
```

Capabilities:

PARAMETER	HARDWARE	CURRENT
MacSecBypass	NotSupported	NotSupported
DcbxSupport	None	None
NumTCs(Max/ETS/PFC)	8/8/8	8/8/8

OperationalTrafficClasses:

TC	TSA	BANDWIDTH	PRIORITIES
0	Strict		0-7

OperationalFlowControl:

Priority 3 Enabled

OperationalClassifications:

PROTOCOL	PORT/TYPE	PRIORITY
Default		0
NetDirect	445	3

7. Reserve half the bandwidth to SMB Direct (RDMA)

```
New-NetQosTrafficClass "SMB" -priority 3 -bandwidthpercentage 50 -algorithm ETS
```

Results:

NAME	ALGORITHM	BANDWIDTH(%)	PRIORITY	POLICYSET	IFINDEX	IFALIAS
SMB	ETS	50	3	Global		

8. View the bandwidth reservation settings:

```
Get-NetQosTrafficClass | ft -AutoSize
```

Results:

NAME	ALGORITHM	BANDWIDTH(%)	PRIORITY	POLICYSET	IFINDEX	IFALIAS
[Default]	ETS	50	0-2,4-7	Global		
SMB	ETS	50	3	Global		

9. (Optional) Create two additional traffic classes for tenant IP traffic.

TIP

You can omit the "IP1" and "IP2" values.

```
New-NetQosTrafficClass "IP1" -Priority 1 -bandwidthpercentage 10 -algorithm ETS
```

Results:

NAME	ALGORITHM	BANDWIDTH(%)	PRIORITY	POLICYSET	IFINDEX	IFALIAS
IP1	ETS	10	1	Global		

```
New-NetQosTrafficClass "IP2" -Priority 2 -bandwidthpercentage 10 -algorithm ETS
```

Results:

NAME	ALGORITHM	BANDWIDTH(%)	PRIORITY	POLICYSET	IFINDEX	IFALIAS
IP2	ETS	10	2	Global		

10. View the QoS traffic classes.

```
Get-NetQosTrafficClass | ft -AutoSize
```

Results:

NAME	ALGORITHM	BANDWIDTH(%)	PRIORITY	POLICYSET	IFINDEX	IFALIAS
[Default]	ETS	30	0,4-7	Global		
SMB	ETS	50	3	Global		
IP1	ETS	10	1	Global		
IP2	ETS	10	2	Global		

11. (Optional) Override the debugger.

By default, the attached debugger blocks NetQos.

```
Set-ItemProperty HKLM:"\SYSTEM\CurrentControlSet\Services\NDIS\Parameters"
AllowFlowControlUnderDebugger -type DWORD -Value 1 -Force
Get-ItemProperty HKLM:"\SYSTEM\CurrentControlSet\Services\NDIS\Parameters" | ft
AllowFlowControlUnderDebugger
```

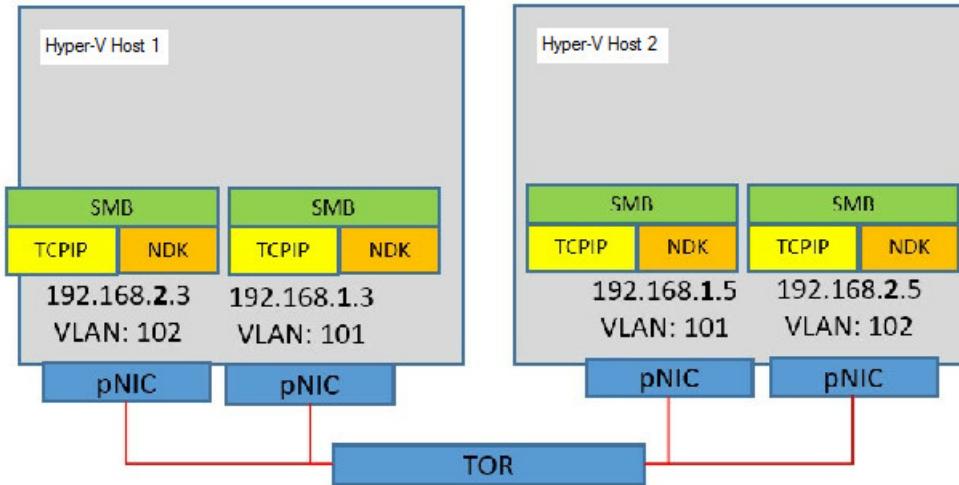
Results:

```
AllowFlowControlUnderDebugger
-----
1
```

Step 5. Verify the RDMA configuration (Mode 1)

You want to ensure that the fabric is configured correctly prior to creating a vSwitch and transitioning to RDMA (Mode 2).

The following image shows the current state of the Hyper-V hosts.



1. Verify the RDMA configuration.

```
Get-NetAdapterRdma | ft -AutoSize
```

Results:

NAME	INTERFACEDESCRIPTION	ENABLED
TEST-40G-1	Mellanox ConnectX-4 VPI Adapter #2	True
TEST-40G-2	Mellanox ConnectX-4 VPI Adapter	True

2. Determine the **ifIndex** value of your target adapters.

You use this value in subsequent steps when you run the script you download.

```
Get-NetIPConfiguration -InterfaceAlias "TEST*" | ft InterfaceAlias,InterfaceIndex,IPv4Address
```

Results:

INTERFACEALIAS	INTERFACEINDEX	IPV4ADDRESS
TEST-40G-1	14	{192.168.1.3}
TEST-40G-2	13	{192.168.2.3}

3. Download the [DiskSpd.exe](#) utility and extract it into C:\TEST.
4. Download the [Test-RDMA PowerShell script](#) to a test folder on your local drive, for example, C:\TEST.
5. Run the **Test-Rdma.ps1** PowerShell script to pass the ifIndex value to the script, along with the IP address of the first remote adapter on the same VLAN.

In this example, the script passes the **ifIndex** value of 14 on the remote network adapter IP address 192.168.1.5.

```
C:\TEST\Test-RDMA.ps1 -IfIndex 14 -IsRoCE $true -RemoteIpAddress 192.168.1.5 -PathToDiskspd
C:\TEST\Diskspd-v2.0.17\amd64fre\
```

Results:

```
VERBOSE: Diskspd.exe found at C:\TEST\Diskspd-v2.0.17\amd64fre\diskspd.exe
VERBOSE: The adapter M2 is a physical adapter
VERBOSE: Underlying adapter is RoCE. Checking if QoS/DCB/PFC is configured on each physical adapter(s)
VERBOSE: QoS/DCB/PFC configuration is correct.
VERBOSE: RDMA configuration is correct.
VERBOSE: Checking if remote IP address, 192.168.1.5, is reachable.
VERBOSE: Remote IP 192.168.1.5 is reachable.
VERBOSE: Disabling RDMA on adapters that are not part of this test. RDMA will be enabled on them later.
VERBOSE: Testing RDMA traffic now for. Traffic will be sent in a parallel job. Job details:
VERBOSE: 0 RDMA bytes written per second
VERBOSE: 0 RDMA bytes sent per second
VERBOSE: 662979201 RDMA bytes written per second
VERBOSE: 37561021 RDMA bytes sent per second
VERBOSE: 1023098948 RDMA bytes written per second
VERBOSE: 8901349 RDMA bytes sent per second
VERBOSE: Enabling RDMA on adapters that are not part of this test. RDMA was disabled on them prior to sending RDMA traffic.
VERBOSE: RDMA traffic test SUCCESSFUL: RDMA traffic was sent to 192.168.1.5
```

NOTE

If the RDMA traffic fails, for the RoCE case specifically, consult your ToR Switch configuration for proper PFC/ETS settings that should match the Host settings. Refer to the QoS section in this document for reference values.

- Run the **Test-Rdma.ps1** PowerShell script to pass the **ifIndex** value to the script, along with the IP address of the second remote adapter on the same VLAN.

In this example, the script passes the **ifIndex** value of 13 on the remote network adapter IP address 192.168.2.5.

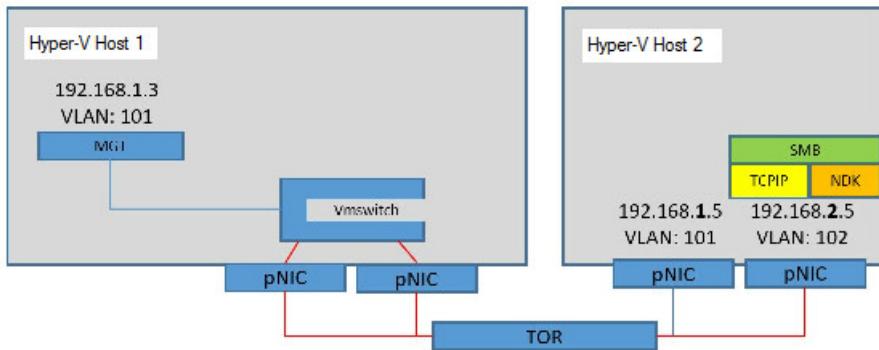
```
C:\TEST\Test-RDMA.ps1 -IfIndex 13 -IsRoCE $true -RemoteIpAddress 192.168.2.5 -PathToDiskspd
C:\TEST\Diskspd-v2.0.17\amd64fre\
```

Results:

```
VERBOSE: Diskspd.exe found at C:\TEST\Diskspd-v2.0.17\amd64fre\diskspd.exe
VERBOSE: The adapter TEST-40G-2 is a physical adapter
VERBOSE: Underlying adapter is RoCE. Checking if QoS/DCB/PFC is configured on each physical adapter(s)
VERBOSE: QoS/DCB/PFC configuration is correct.
VERBOSE: RDMA configuration is correct.
VERBOSE: Checking if remote IP address, 192.168.2.5, is reachable.
VERBOSE: Remote IP 192.168.2.5 is reachable.
VERBOSE: Disabling RDMA on adapters that are not part of this test. RDMA will be enabled on them later.
VERBOSE: Testing RDMA traffic now for. Traffic will be sent in a parallel job. Job details:
VERBOSE: 0 RDMA bytes written per second
VERBOSE: 0 RDMA bytes sent per second
VERBOSE: 541185606 RDMA bytes written per second
VERBOSE: 34821478 RDMA bytes sent per second
VERBOSE: 954717307 RDMA bytes written per second
VERBOSE: 35040816 RDMA bytes sent per second
VERBOSE: Enabling RDMA on adapters that are not part of this test. RDMA was disabled on them prior to sending RDMA traffic.
VERBOSE: RDMA traffic test SUCCESSFUL: RDMA traffic was sent to 192.168.2.5
```

Step 6. Create a Hyper-V vSwitch on your Hyper-V hosts

The following image shows Hyper-V Host 1 with a vSwitch.



1. Create a vSwitch in SET mode on Hyper-V host 1.

```
New-VMSwitch -Name "VMSTEST" -NetAdapterName "TEST-40G-1","TEST-40G-2" -EnableEmbeddedTeaming $true -AllowManagementOS $true
```

Result:

NAME	SWITCHTYPE	NETADAPTERINTERFACEDESCRIPTION
VMSTEST	External	Teamed-Interface

2. View the physical adapter team in SET.

```
Get-VMSwitchTeam -Name "VMSTEST" | fl
```

Results:

```
Name: VMSTEST
Id: ad9bb542-dda2-4450-a00e-f96d44bdfbec
NetAdapterInterfaceDescription: {Mellanox ConnectX-3 Pro Ethernet Adapter, Mellanox ConnectX-3 Pro Ethernet Adapter #2}
TeamingMode: SwitchIndependent
LoadBalancingAlgorithm: Dynamic
```

3. Display two views of the host vNIC

```
Get-NetAdapter
```

Results:

NAME	INTERFACEDESCRIPTION	IFINDEX	STATUS	MACADDRESS	LINKSPEED
vEthernet (VMSTEST)	Hyper-V Virtual Ethernet Adapter #2	28	Up	E4-1D-2D-07-40-71	80 Gbps

4. View additional properties of the host vNIC.

```
Get-VMNetworkAdapter -ManagementOS
```

Results:

NAME	ISMANAGEMEN TOS	VMNAME	SWITCHNAME	MACADDRESS	STATUS	IPADDRESSES
VMSTEST	True	VMSTEST	E41D2D074 071	{Ok}		

5. Test the network connection to the remote VLAN 101 adapter.

```
Test-NetConnection 192.168.1.5
```

Results:

```
WARNING: Ping to 192.168.1.5 failed -- Status: DestinationHostUnreachable

ComputerName    : 192.168.1.5
RemoteAddress   : 192.168.1.5
InterfaceAlias  : vEthernet (CORP-External-Switch)
SourceAddress   : 10.199.48.170
PingSucceeded   : False
PingReplyDetails (RTT) : 0 ms
```

Step 7. Remove the Access VLAN setting

In this step, you remove the ACCESS VLAN setting from the physical NIC and to set the VLANID using the vSwitch.

You must remove the ACCESS VLAN setting to prevent both auto-tagging the egress traffic with the incorrect VLAN ID and from filtering ingress traffic which doesn't match the ACCESS VLAN ID.

1. Remove the setting.

```
Set-NetAdapterAdvancedProperty -Name "Test-40G-1" -RegistryKeyword VlanID -RegistryValue "0"
Set-NetAdapterAdvancedProperty -Name "Test-40G-2" -RegistryKeyword VlanID -RegistryValue "0"
```

2. Set the VLAN ID.

```
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "VMSTEST" -VlanId "101" -Access -ManagementOS
Get-VMNetworkAdapterVlan -ManagementOS -VMNetworkAdapterName "VMSTEST"
```

Results:

VMName	VMNetworkAdapterName	Mode	VlanList
VMSTEST		Access	101

3. Test the network connection.

```
Test-NetConnection 192.168.1.5
```

Results:

```
ComputerName : 192.168.1.5
RemoteAddress : 192.168.1.5
InterfaceAlias : vEthernet (VMTEST)
SourceAddress : 192.168.1.3
PingSucceeded : True
PingReplyDetails (RTT) : 0 ms
```

IMPORTANT If your results are not similar to the example results and ping fails with the message "WARNING: Ping to 192.168.1.5 failed -- Status: DestinationHostUnreachable," confirm that the "vEthernet (VMTEST)" has the proper IP address.

```
Get-NetIPAddress -InterfaceAlias "vEthernet (VMTEST)"
```

If the IP address is not set, correct the issue.

```
New-NetIPAddress -InterfaceAlias "vEthernet (VMTEST)" -IPAddress 192.168.1.3 -PrefixLength 24

IPAddress : 192.168.1.3
InterfaceIndex: 37
InterfaceAlias: vEthernet (VMTEST)
AddressFamily : IPv4
Type : Unicast
PrefixLength : 24
PrefixOrigin : Manual
SuffixOrigin : Manual
AddressState : Tentative
ValidLifetime : Infinite ([TimeSpan]::.MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::.MaxValue)
SkipAsSource : False
PolicyStore : ActiveStore
```

4. Rename the Management NIC.

```
Rename-VMNetworkAdapter -ManagementOS -Name "VMTEST" -NewName "MGT"
Get-VMNetworkAdapter -ManagementOS
```

Results:

NAME	ISMANAGEMEN TOS	VMNAME	SWITCHNAME	MACADDRESS	STATUS	IPADDRESSES
CORP-External-Switch	True		CORP-External-Switch	001B785768AA	{Ok}	
MGT	True		VMTEST	E41D2D074071	{Ok}	

5. View additional NIC properties.

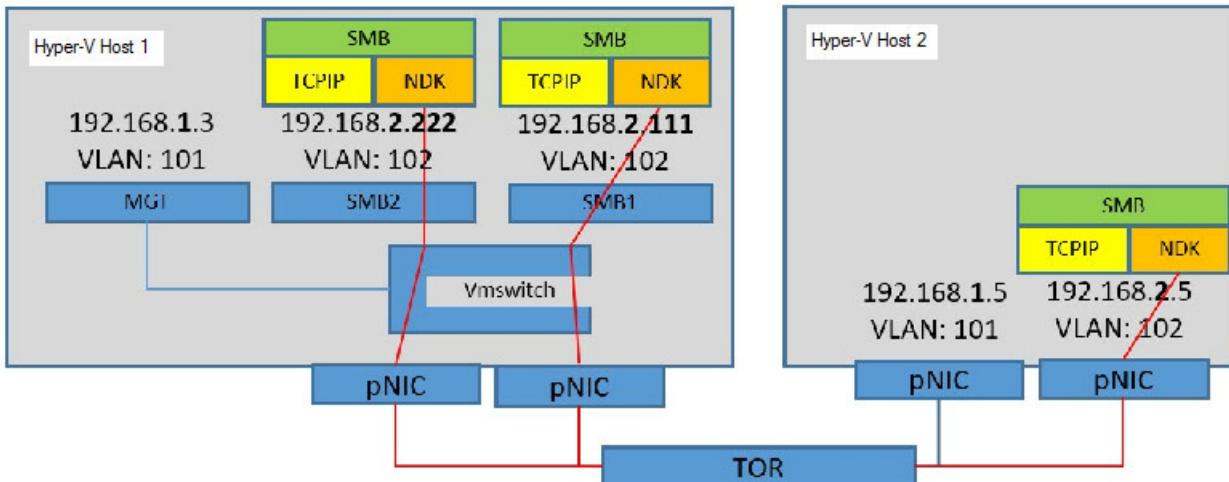
```
Get-NetAdapter
```

Results:

Name	Interface Description	IfIndex	Status	MacAddress	LinkSpeed
vEthernet (MGT)	Hyper-V Virtual Ethernet Adapter #2	28	Up	E4-1D-2D-07-40-71	80 Gbps

Step 8. Test Hyper-V vSwitch RDMA

The following image shows the current state of your Hyper-V hosts, including the vSwitch on Hyper-V Host 1.



- Set the priority tagging on the Host vNIC to complement the previous VLAN settings.

```
Set-VMNetworkAdapter -ManagementOS -Name "MGT" -IeeePriorityTag on
Get-VMNetworkAdapter -ManagementOS -Name "MGT" | fl Name,IeeePriorityTag
```

Results:

Name : MGT

IeeePriorityTag : On

- Create two host vNICs for RDMA and connect them to the vSwitch VMTEST.

```
Add-VMNetworkAdapter -SwitchName "VMTEST" -Name SMB1 -ManagementOS
Add-VMNetworkAdapter -SwitchName "VMTEST" -Name SMB2 -ManagementOS
```

- View the Management NIC properties.

```
Get-VMNetworkAdapter -ManagementOS
```

Results:

Name	IsManagementOS	VmName	SwitchName	MacAddress	Status	IPAddresses
CORP-External-Switch	True	CORP-External-Switch	001B785768AA	{Ok}		

NAME	ISMANAGEMEN NTOS	VMNAME	SWITCHNAME	MACADDRESS	STATUS	IPADDRESSES
Mgt	True	VMTEST	E41D2D074 071	{Ok}		
SMB1	True	VMTEST	00155D30A A00	{Ok}		
SMB2	True	VMTEST	00155D30A A01	{Ok}		

Step 9. Assign an IP address to the SMB Host vNICs vEthernet (SMB1) and vEthernet (SMB2)

The TEST-40G-1 and TEST-40G-2 physical adapters still have an ACCESS VLAN of 101 and 102 configured. Because of this, the adapters tag the traffic - and ping succeeds. Previously, you configured both pNIC VLAN IDs to zero, then set the VMTEST vSwitch to VLAN 101. After that, you were still able to ping the remote VLAN 101 adapter by using the MGT vNIC, but there are currently no VLAN 102 members.

1. Remove the ACCESS VLAN setting from the physical NIC to prevent it from both auto-tagging the egress traffic with the incorrect VLAN ID and to prevent it from filtering ingress traffic that doesn't match the ACCESS VLAN ID.

```
New-NetIPAddress -InterfaceAlias "vEthernet (SMB1)" -IPAddress 192.168.2.111 -PrefixLength 24
```

Results:

```

IPAddress : 192.168.2.111
InterfaceIndex: 40
InterfaceAlias: vEthernet (SMB1)
AddressFamily : IPv4
Type : Unicast
PrefixLength : 24
PrefixOrigin : Manual
SuffixOrigin : Manual
AddressState : Invalid
ValidLifetime : Infinite ([TimeSpan]::.MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::.MaxValue)
SkipAsSource : False
PolicyStore : PersistentStore

```

2. Test the remote VLAN 102 adapter.

```
Test-NetConnection 192.168.2.5
```

Results:

```

ComputerName : 192.168.2.5
RemoteAddress : 192.168.2.5
InterfaceAlias : vEthernet (SMB1)
SourceAddress : 192.168.2.111
PingSucceeded : True
PingReplyDetails (RTT) : 0 ms

```

3. Add a new IP address for interface vEthernet (SMB2).

```
New-NetIPAddress -InterfaceAlias "vEthernet (SMB2)" -IPAddress 192.168.2.222 -PrefixLength 24
```

Results:

```
IPAddress : 192.168.2.222
InterfaceIndex: 44
InterfaceAlias: vEthernet (SMB2)
AddressFamily : IPv4
Type : Unicast
PrefixLength : 24
PrefixOrigin : Manual
SuffixOrigin : Manual
AddressState : Invalid
ValidLifetime : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource : False
PolicyStore : PersistentStore
```

4. Test the connection again.
5. Place the RDMA Host vNICs on the pre-existing VLAN 102.

```
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "SMB1" -VlanId "102" -Access -ManagementOS
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "SMB2" -VlanId "102" -Access -ManagementOS

Get-VMNetworkAdapterVlan -ManagementOS
```

Results:

VMName	VMNetworkAdapterName	Mode	VlanList
SMB1	Access	102	
Mgt	Access	101	
SMB2	Access	102	
CORP-External-Switch	Untagged		

6. Inspect the mapping of SMB1 and SMB2 to the underlying physical NICs under the vSwitch SET Team.

The association of Host vNIC to Physical NICs is random and subject to rebalancing during creation and destruction. In this circumstance, you can use an indirect mechanism to check the current association. The MAC addresses of SMB1 and SMB2 are associated with the NIC Team member TEST-40G-2. This is not ideal because Test-40G-1 does not have an associated SMB Host vNIC, and will not allow for utilization of RDMA traffic over the link until an SMB Host vNIC is mapped to it.

```
Get-NetAdapterVPort (Preferred)

Get-NetAdapterVmqQueue
```

Results:

```

Name QueueID MacAddressVlanID Processor VmFriendlyName
-----
TEST-40G-1 1 E4-1D-2D-07-40-71 1010:17
TEST-40G-2 1 00-15-5D-30-AA-00 1020:17
TEST-40G-2 2 00-15-5D-30-AA-01 1020:17

```

7. View the VM network adapter properties.

```
Get-VMNetworkAdapter -ManagementOS
```

Results:

Name	IsManagementOs	VMName	SwitchName	MacAddress	Status	IPAddresses
CORP-External-Switch	True	CORP-External-Switch		001B785768AA	{Ok}	
Mgt	True	VMSTEST	E41D2D074071		{Ok}	
SMB1	True	VMSTEST	00155D30AA00		{Ok}	
SMB2	True	VMSTEST	00155D30AA01		{Ok}	

8. View the network adapter team mapping.

The results should not return information because you have not yet performed mapping.

```

Get-VMNetworkAdapterTeamMapping -ManagementOS -SwitchName VMSTEST -VMNetworkAdapterName SMB1
Get-VMNetworkAdapterTeamMapping -ManagementOS -SwitchName VMSTEST -VMNetworkAdapterName SMB2

```

9. Map SMB1 and SMB2 to separate physical NIC team members, and to view the results of your actions.

IMPORTANT

Ensure that you complete this step before proceeding, or your implementation fails.

```

Set-VMNetworkAdapterTeamMapping -ManagementOS -SwitchName VMSTEST -VMNetworkAdapterName "SMB1" -
PhysicalNetAdapterName "Test-40G-1"
Set-VMNetworkAdapterTeamMapping -ManagementOS -SwitchName VMSTEST -VMNetworkAdapterName "SMB2" -
PhysicalNetAdapterName "Test-40G-2"

Get-VMNetworkAdapterTeamMapping -ManagementOS -SwitchName VMSTEST

```

Results:

```
NetAdapterName : Test-40G-1
NetAdapterDeviceId : {BAA9A00F-A844-4740-AA93-6BD838F8CFBA}
ParentAdapter : VMInternalNetworkAdapter, Name = 'SMB1'
IsTemplate : False
CimSession : CimSession: .
ComputerName : 27-3145G0803
IsDeleted : False

NetAdapterName : Test-40G-2
NetAdapterDeviceId : {B7AB5BB3-8ACB-444B-8B7E-BC882935EBC8}
ParentAdapter : VMInternalNetworkAdapter, Name = 'SMB2'
IsTemplate : False
CimSession : CimSession: .
ComputerName : 27-3145G0803
IsDeleted : False
```

10. Confirm the MAC associations created previously.

```
Get-NetAdapterVmqQueue
```

Results:

Name	QueueID	MacAddressVlanID	Processor	VmFriendlyName
TEST-40G-1	1	E4-1D-2D-07-40-71	1010:17	
TEST-40G-1	2	00-15-5D-30-AA-00	1020:17	
TEST-40G-2	1	00-15-5D-30-AA-01	1020:17	

11. Test the connection from the remote system because both Host vNICs reside on the same subnet and have the same VLAN ID (102).

```
Test-NetConnection 192.168.2.111
```

Results:

```
ComputerName : 192.168.2.111
RemoteAddress : 192.168.2.111
InterfaceAlias : Test-40G-2
SourceAddress : 192.168.2.5
PingSucceeded : True
PingReplyDetails (RTT) : 0 ms
```

```
Test-NetConnection 192.168.2.222
```

Results:

```
ComputerName : 192.168.2.222
RemoteAddress : 192.168.2.222
InterfaceAlias : Test-40G-2
SourceAddress : 192.168.2.5
PingSucceeded : True
PingReplyDetails (RTT) : 0 ms
```

12. Set the name, switch name, and priority tags.

```
Set-VMNetworkAdapter -ManagementOS -Name "SMB1" -IeeePriorityTag on
Set-VMNetworkAdapter -ManagementOS -Name "SMB2" -IeeePriorityTag on
Get-VMNetworkAdapter -ManagementOS -Name "SMB*" | fl Name,SwitchName,IeeePriorityTag,Status
```

Results:

```
Name: SMB1
SwitchName : VMSTEST
IeeePriorityTag : On
Status : {Ok}
```

```
Name: SMB2
SwitchName : VMSTEST
IeeePriorityTag : On
Status : {Ok}
```

13. View the vEthernet network adapter properties.

```
Get-NetAdapterRdma -Name "vEthernet*" | sort Name | ft -AutoSize
```

Results:

Name	InterfaceDescription	Enabled
vEthernet (SMB2)	Hyper-V Virtual Ethernet Adapter #4	False
vEthernet (SMB1)	Hyper-V Virtual Ethernet Adapter #3	False
vEthernet (MGT)	Hyper-V Virtual Ethernet Adapter #2	False

14. Enable the vEthernet network adapters.

```
Enable-NetAdapterRdma -Name "vEthernet (SMB1)"
Enable-NetAdapterRdma -Name "vEthernet (SMB2)"
Get-NetAdapterRdma -Name "vEthernet*" | sort Name | fl *
```

Results:

Name	InterfaceDescription	Enabled
vEthernet (SMB2)	Hyper-V Virtual Ethernet Adapter #4	True
vEthernet (SMB1)	Hyper-V Virtual Ethernet Adapter #3	True
vEthernet (MGT)	Hyper-V Virtual Ethernet Adapter #2	False

Step 10. Validate the RDMA functionality.

You want to validate the RDMA functionality from the remote system to the local system, which has a vSwitch, to both members of the vSwitch SET team.

Because both Host vNICs (SMB1 and SMB2) are assigned to VLAN 102, you can select the VLAN 102 adapter on the remote system.

In this example, the NIC Test-40G-2 does RDMA to SMB1 (192.168.2.111) and SMB2 (192.168.2.222).

TIP

You might need to disable the Firewall on this system. Consult your fabric policy for details.

```
Set-NetFirewallProfile -All -Enabled False  
Get-NetAdapterAdvancedProperty -Name "Test-40G-2"
```

Results:

Name	DisplayName	DisplayValue	RegistryKeyword	RegistryValue
Test-40G-2VLAN	ID102VlanID	{102}		

1. View the network adapter properties.

```
Get-NetAdapter
```

Results:

Name	InterfaceDescription	ifIndex	Status	MacAddress	LinkSpeed
Test-40G-2Mellanox ConnectX-3 Pro Ethernet A...#3	Mellanox ConnectX-3 Pro Ethernet Adapter	3	Up	E4-1D-2D-07-43-D140	Gbps

2. View the network adapter RDMA information.

```
Get-NetAdapterRdma
```

Results:

Name	InterfaceDescription	Enabled
Test-40G-2Mellanox ConnectX-3 Pro Ethernet Adapter	Mellanox ConnectX-3 Pro Ethernet Adapter	True

3. Perform the RDMA traffic test on the first physical adapter.

```
C:\TEST\Test-RDMA.ps1 -IfIndex 3 -IsRoCE $true -RemoteIpAddress 192.168.2.111 -PathToDiskspd  
C:\TEST\Diskspd-v2.0.17\amd64fre\
```

Results:

```
VERBOSE: Diskspd.exe found at C:\TEST\Diskspd-v2.0.17\amd64fre\diskspd.exe
VERBOSE: The adapter Test-40G-2 is a physical adapter
VERBOSE: Underlying adapter is RoCE. Checking if QoS/DCB/PFC is configured on each physical adapter(s)
VERBOSE: QoS/DCB/PFC configuration is correct.
VERBOSE: RDMA configuration is correct.
VERBOSE: Checking if remote IP address, 192.168.2.111, is reachable.
VERBOSE: Remote IP 192.168.2.111 is reachable.
VERBOSE: Disabling RDMA on adapters that are not part of this test. RDMA will be enabled on them later.
VERBOSE: Testing RDMA traffic now for. Traffic will be sent in a parallel job. Job details:
VERBOSE: 34251744 RDMA bytes sent per second
VERBOSE: 967346308 RDMA bytes written per second
VERBOSE: 35698177 RDMA bytes sent per second
VERBOSE: 976601842 RDMA bytes written per second
VERBOSE: Enabling RDMA on adapters that are not part of this test. RDMA was disabled on them prior to
sending RDMA traffic.
VERBOSE: RDMA traffic test SUCCESSFUL: RDMA traffic was sent to 192.168.2.111
```

4. Perform the RDMA traffic test on the second physical adapter.

```
C:\TEST\Test-RDMA.ps1 -IfIndex 3 -IsRoCE $true -RemoteIpAddress 192.168.2.222 -PathToDiskspd
C:\TEST\Diskspd-v2.0.17\amd64fre\
```

Results:

```
VERBOSE: Diskspd.exe found at C:\TEST\Diskspd-v2.0.17\amd64fre\diskspd.exe
VERBOSE: The adapter Test-40G-2 is a physical adapter
VERBOSE: Underlying adapter is RoCE. Checking if QoS/DCB/PFC is configured on each physical adapter(s)
VERBOSE: QoS/DCB/PFC configuration is correct.
VERBOSE: RDMA configuration is correct.
VERBOSE: Checking if remote IP address, 192.168.2.222, is reachable.
VERBOSE: Remote IP 192.168.2.222 is reachable.
VERBOSE: Disabling RDMA on adapters that are not part of this test. RDMA will be enabled on them later.
VERBOSE: Testing RDMA traffic now for. Traffic will be sent in a parallel job. Job details:
VERBOSE: 0 RDMA bytes written per second
VERBOSE: 0 RDMA bytes sent per second
VERBOSE: 485137693 RDMA bytes written per second
VERBOSE: 35200268 RDMA bytes sent per second
VERBOSE: 939044611 RDMA bytes written per second
VERBOSE: 34880901 RDMA bytes sent per second
VERBOSE: Enabling RDMA on adapters that are not part of this test. RDMA was disabled on them prior to
sending RDMA traffic.
VERBOSE: RDMA traffic test SUCCESSFUL: RDMA traffic was sent to 192.168.2.222
```

5. Test for RDMA traffic from the local to the remote computer.

```
Get-NetAdapter | ft -AutoSize
```

Results:

Name	InterfaceDescription	ifIndex	Status	MacAddress	LinkSpeed
vEthernet (SMB2)	Hyper-V Virtual Ethernet Adapter	#4	Up	00-15-5D-30-AA-0380	Gbps
vEthernet (SMB1)	Hyper-V Virtual Ethernet Adapter	#3	Up	00-15-5D-30-AA-0280	Gbps

6. Perform the RDMA traffic test on the first virtual adapter.

```
C:\TEST\Test-RDMA.ps1 -IfIndex 41 -IsRoCE $true -RemoteIpAddress 192.168.2.5 -PathToDiskspd  
C:\TEST\Diskspd-v2.0.17\amd64fre\
```

Results:

```
VERBOSE: Diskspd.exe found at C:\TEST\Diskspd-v2.0.17\amd64fre\diskspd.exe  
VERBOSE: The adapter vEthernet (SMB1) is a virtual adapter  
VERBOSE: Retrieving vSwitch bound to the virtual adapter  
VERBOSE: Found vSwitch: VMSTEST  
VERBOSE: Found the following physical adapter(s) bound to vSwitch: TEST-40G-1, TEST-40G-2  
VERBOSE: Underlying adapter is RoCE. Checking if QoS/DCB/PFC is configured on each physical adapter(s)  
VERBOSE: QoS/DCB/PFC configuration is correct.  
VERBOSE: RDMA configuration is correct.  
VERBOSE: Checking if remote IP address, 192.168.2.5, is reachable.  
VERBOSE: Remote IP 192.168.2.5 is reachable.  
VERBOSE: Disabling RDMA on adapters that are not part of this test. RDMA will be enabled on them later.  
VERBOSE: Testing RDMA traffic now for. Traffic will be sent in a parallel job. Job details:  
VERBOSE: 0 RDMA bytes written per second  
VERBOSE: 0 RDMA bytes sent per second  
VERBOSE: 0 RDMA bytes written per second  
VERBOSE: 15250197 RDMA bytes sent per second  
VERBOSE: 896320913 RDMA bytes written per second  
VERBOSE: 33947559 RDMA bytes sent per second  
VERBOSE: 912160540 RDMA bytes written per second  
VERBOSE: 34091930 RDMA bytes sent per second  
VERBOSE: Enabling RDMA on adapters that are not part of this test. RDMA was disabled on them prior to sending RDMA traffic.  
VERBOSE: RDMA traffic test SUCCESSFUL: RDMA traffic was sent to 192.168.2.5
```

7. Perform the RDMA traffic test on the second virtual adapter.

```
C:\TEST\Test-RDMA.ps1 -IfIndex 45 -IsRoCE $true -RemoteIpAddress 192.168.2.5 -PathToDiskspd  
C:\TEST\Diskspd-v2.0.17\amd64fre\
```

Results:

```
VERBOSE: Diskspd.exe found at C:\TEST\Diskspd-v2.0.17\amd64fre\diskspd.exe  
VERBOSE: The adapter vEthernet (SMB2) is a virtual adapter  
VERBOSE: Retrieving vSwitch bound to the virtual adapter  
VERBOSE: Found vSwitch: VMSTEST  
VERBOSE: Found the following physical adapter(s) bound to vSwitch: TEST-40G-1, TEST-40G-2  
VERBOSE: Underlying adapter is RoCE. Checking if QoS/DCB/PFC is configured on each physical adapter(s)  
VERBOSE: QoS/DCB/PFC configuration is correct.  
VERBOSE: RDMA configuration is correct.  
VERBOSE: Checking if remote IP address, 192.168.2.5, is reachable.  
VERBOSE: Remote IP 192.168.2.5 is reachable.  
VERBOSE: Disabling RDMA on adapters that are not part of this test. RDMA will be enabled on them later.  
VERBOSE: Testing RDMA traffic now for. Traffic will be sent in a parallel job. Job details:  
VERBOSE: 0 RDMA bytes written per second  
VERBOSE: 0 RDMA bytes sent per second  
VERBOSE: 385169487 RDMA bytes written per second  
VERBOSE: 33902277 RDMA bytes sent per second  
VERBOSE: 907354685 RDMA bytes written per second  
VERBOSE: 33923662 RDMA bytes sent per second  
VERBOSE: Enabling RDMA on adapters that are not part of this test. RDMA was disabled on them prior to sending RDMA traffic.  
VERBOSE: RDMA traffic test SUCCESSFUL: RDMA traffic was sent to 192.168.2.5
```

The final line in this output, "RDMA traffic test SUCCESSFUL: RDMA traffic was sent to 192.168.2.5," shows that you have successfully configured Converged NIC on your adapter.

Related topics

- [Converged NIC Configuration with a Single Network Adapter](#)
- [Physical Switch Configuration for Converged NIC](#)
- [Troubleshooting Converged NIC Configurations](#)

Physical switch configuration for Converged NIC

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

In this topic, we provide you with guidelines for configuring your physical switches.

These are only commands and their uses; you must determine the ports to which the NICs are connected in your environment.

IMPORTANT

Ensure that the VLAN and no-drop policy is set for the priority over which SMB is configured.

Arista switch (dcs-7050s-64, EOS-4.13.7M)

1. en (go to admin mode, usually asks for a password)
2. config (to enter into configuration mode)
3. show run (shows current running configuration)
4. find out switch ports to which your NICs are connected to. In these example, they are 14/1,15/1,16/1,17/1.
5. int eth 14/1,15/1,16/1,17/1 (enter into config mode for these ports)
6. dcbx mode ieee
7. priority-flow-control mode on
8. switchport trunk native vlan 225
9. switchport trunk allowed vlan 100-225
10. switchport mode trunk
11. priority-flow-control priority 3 no-drop
12. qos trust cos
13. show run (verify that configuration is setup correctly on the ports)
14. wr (to make the settings persists across switch reboot)

Tips:

1. No #command# negates a command
2. How to add a new VLAN: int vlan 100 (If storage network is on VLAN 100)
3. How to check existing VLANs : show vlan
4. For more information on configuring Arista Switch, search online for: Arista EOS Manual
5. Use this command to verify PFC settings: show priority-flow-control counters detail

Dell switch (S4810, FTOS 9.9 (0.0))

```

!
dcb enable
! put pfc control on qos class 3
configure
dcb-map dcb-smb
priority group 0 bandwidth 90 pfc on
priority group 1 bandwidth 10 pfc off
priority-pgid 1 1 1 0 1 1 1 1
exit
! apply map to ports 0-31
configure
interface range ten 0/0-31
dcb-map dcb-smb
exit

```

Cisco switch (Nexus 3132, version 6.0(2)U6(1))

Global

```

class-map type qos match-all RDMA
match cos 3
class-map type queuing RDMA
match qos-group 3
policy-map type qos QOS_MARKING
class RDMA
set qos-group 3
class class-default
policy-map type queuing QOS_QUEUEING
class type queuing RDMA
bandwidth percent 50
class type queuing class-default
bandwidth percent 50
class-map type network-qos RDMA
match qos-group 3
policy-map type network-qos QOS_NETWORK
class type network-qos RDMA
mtu 2240
pause no-drop
class type network-qos class-default
mtu 9216
system qos
service-policy type qos input QOS_MARKING
service-policy type queuing output QOS_QUEUEING
service-policy type network-qos QOS_NETWORK

```

Port specific

```

switchport mode trunk
switchport trunk native vlan 99
switchport trunk allowed vlan 99,2000,2050    cause VLANs that already exists
spanning-tree port type edge
flowcontrol receive on (not supported with PFC in Cisco NX-OS)
flowcontrol send on (not supported with PFC in Cisco NX-OS)
no shutdown
priority-flow-control mode on

```

Related topics

- [Converged NIC Configuration with a Single Network Adapter](#)

- Converged NIC Teamed NIC Configuration
 - Troubleshooting Converged NIC Configurations
-

Troubleshooting Converged NIC Configurations

9/18/2018 • 3 minutes to read • [Edit Online](#)

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

You can use the following script to verify whether the RDMA configuration is correct on the Hyper-V host.

- [Download script Test-Rdma.ps1](#)

You can also use the following Windows PowerShell commands to troubleshoot and verify the configuration of your converged NICs.

Get-NetAdapterRdma

To verify your network adapter RDMA configuration, run the following Windows PowerShell command on the Hyper-V server.

```
Get-NetAdapterRdma | fl *
```

You can use the following expected and unexpected results to identify and resolve issues after you run this command on the Hyper-V host.

Get-NetAdapterRdma expected results

Host vNIC and the physical NIC show non-zero RDMA capabilities.

```

PS C:\Windows\system32> Get-NetAdapterRdma | fl *

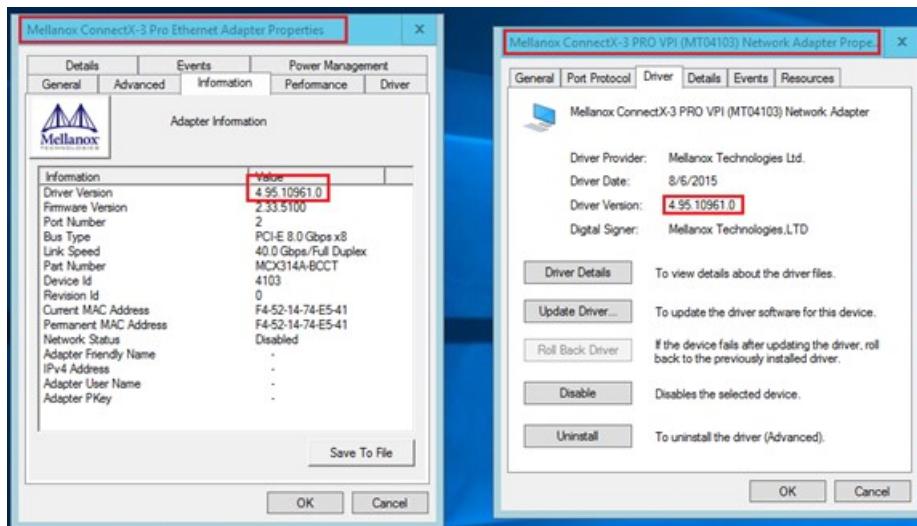
```

ifAlias	:	vEthernet (MlnxSwitch)
InterfaceAlias	:	vEthernet (MlnxSwitch)
ifDesc	:	Hyper-V Virtual Ethernet Adapter #2
Caption	:	MSFT_NetAdapterRdmaSettingData 'Hyper-V Virtual Ethernet Adapter #2'
Description	:	Hyper-V Virtual Ethernet Adapter #2
ElementName	:	Hyper-V Virtual Ethernet Adapter #2
InstanceID	:	{960FA0DF-9CBF-405F-A0DA-CDC67443D369}
InterfaceDescription	:	Hyper-V Virtual Ethernet Adapter #2
Name	:	vEthernet (MlnxSwitch)
Source	:	2
SystemName	:	27-3145G0511.redmond.corp.microsoft.com
Enabled	:	True
MaxCompletionQueueCount	:	65408
MaxInboundReadLimit	:	32504832
MaxMemoryRegionCount	:	524032
MaxMemoryWindowCount	:	0
MaxOutboundReadLimit	:	260038656
MaxProtectionDomainCount	:	32764
MaxQueuePairCount	:	2031552
MaxSharedReceiveQueueCount	:	65472
RdmaAdapterInfo	:	MSFT_NetAdapter_RdmaAdapterInfo
RdmaMissingCounterInfo	:	MSFT_NetAdapter_RdmaMissingCounterInfo
PSComputerName	:	
CimClass	:	ROOT/StandardCimv2:MSFT_NetAdapterRdmaSettingData
CimInstanceProperties	:	{Caption, Description, ElementName, InstanceID...}
CimSystemProperties	:	Microsoft.Management.Infrastructure.CimSystemProperties
ifAlias	:	Slot 4
InterfaceAlias	:	Slot 4
ifDesc	:	Mellanox ConnectX-3 Pro Ethernet Adapter
Caption	:	MSFT_NetAdapterRdmaSettingData 'Mellanox ConnectX-3 Pro Ethernet Adapter'
Description	:	Mellanox ConnectX-3 Pro Ethernet Adapter
ElementName	:	Mellanox ConnectX-3 Pro Ethernet Adapter
InstanceID	:	{11AC55A5-D705-4F54-92D6-A4050B4E7970}
InterfaceDescription	:	Mellanox ConnectX-3 Pro Ethernet Adapter
Name	:	Slot 4
Source	:	2
SystemName	:	27-3145G0511.redmond.corp.microsoft.com
Enabled	:	True
MaxCompletionQueueCount	:	65408
MaxInboundReadLimit	:	32504832
MaxMemoryRegionCount	:	524032
MaxMemoryWindowCount	:	0
MaxOutboundReadLimit	:	260038656
MaxProtectionDomainCount	:	32764
MaxQueuePairCount	:	2031552
MaxSharedReceiveQueueCount	:	65472
RdmaAdapterInfo	:	MSFT_NetAdapter_RdmaAdapterInfo
RdmaMissingCounterInfo	:	MSFT_NetAdapter_RdmaMissingCounterInfo
PSComputerName	:	
CimClass	:	ROOT/StandardCimv2:MSFT_NetAdapterRdmaSettingData
CimInstanceProperties	:	{Caption, Description, ElementName, InstanceID...}
CimSystemProperties	:	Microsoft.Management.Infrastructure.CimSystemProperties

Get-NetAdapterRdma unexpected results

Perform the following steps if you receive unexpected results when you run the **Get-NetAdapterRdma** command.

1. Make sure the Mlnx miniport and Mlnx bus drivers are latest. For Mellanox, use at least drop 42.
2. Verify that Mlnx miniport and bus drivers match by checking the driver version through Device Manager. The bus driver can be found in System Devices. The name should start with Mellanox Connect-X 3 PRO VPI, as illustrated in the following screen shot of network adapter properties.



1. Make sure Network Direct (RDMA) is enabled on both the physical NIC and host vNIC.

2. Make sure vSwitch is created over the right physical adapter by checking its RDMA capabilities.
3. Check EventViewer System log and filter by source "Hyper-V-VmSwitch".

Get SmbClientNetworkInterface

As an additional step to verify your RDMA configuration, run the following Windows PowerShell command on the Hyper-V server.

```
Get SmbClientNetworkInterface
```

Get SmbClientNetworkInterface expected results

The host vNIC should appear as RDMA capable from SMB's perspective as well.

Interface Index	RSS Capable	RDMA Capable	Speed	IpAddresses	Friendly Name
5	False	False	0 bps	[fe80::457b:82ad:feb8:4a87]	NIC3
7	False	False	0 bps	[fe80::89b3:f4d1:e43d:4e6c]	NIC4
4	False	False	0 bps	[fe80::bc08:28b0:9cb2:94aa]	NIC1
6	False	False	0 bps	[fe80::2421:6122:2e5e:5e60]	NIC2
16	True	True	40 Gbps	{12.0.0.3}	vEthernet (MlnxSwitch)
9	False	False	100 Gbps	[fe80::1:1ff:fe10:195.58.182]	[isatap_{38000000-0000-0000-0000-000000000000}]
8	False	False	100 kbps	[fe80::5:fffe:10:195.58.182]	isatap.corp.microsoft.com
2	False	False	1 Gbps	{2001:4898:1b:1046:54ca:10c4:1cc1:ef6, fe80::54ca:10c4:1cc1:ef6, 10.195.58.182}	Local Area Connection^ 1

Get SmbClientNetworkInterface unexpected results

1. Make sure the Mlnx miniport and Mlnx bus drivers are latest. For Mellanox, use at least drop 42.
2. Verify that Mlnx miniport and bus drivers match by checking the driver version through Device Manager. The bus driver can be found in System Devices. The name should start with Mellanox Connect-X 3 PRO VPI, as illustrated in the following screen shot of network adapter properties.
3. Make sure Network Direct (RDMA) is enabled on both the physical NIC and host vNIC.
4. Make sure the Hyper-V Virtual Switch is created over the right physical adapter by checking its RDMA capabilities.
5. Check EventViewer logs for "SMB Client" in **Application And Services | Microsoft | Windows**.

Get-NetAdapterQos

You can view the network adapter quality of service (QoS) configuration by running the following Windows PowerShell command.

```
Get-NetAdapterQos
```

Get-NetAdapterQos expected results

Priorities and traffic classes should be displayed according to the first configuration step that you performed using this guide.

Name	:	Slot 4
Enabled	:	True
Capabilities	:	Hardware Current
		----- -----
MacSecBypass	:	NotSupported NotSupported
DcbxSupport	:	None None
NumTCs (Max/ETS/PFC)	:	8/8/8 8/8/8
OperationalTrafficClasses	:	TC TSA Bandwidth Priorities
		--- --- -----
0 ETS	:	20% 0-2,4-7
1 ETS	:	80% 3
OperationalFlowControl	:	Priority 3 Enabled
OperationalClassifications	:	Protocol Port/Type Priority

NetDirect	:	445 3

Get-NetAdapterQos unexpected results

If your results are unexpected, perform the following steps.

1. Ensure that the physical network adapter supports Data Center Bridging (DCB) and QoS
2. Ensure that the network adapter drivers are up to date.

Get-SmbMultiChannelConnection

You can use the following Windows PowerShell command to verify that the remote node's IP address is RDMA-capable.

```
Get-SmbMultiChannelConnection
```

Get-SmbMultiChannelConnection expected results

Remote node's IP address is shown as RDMA capable.

PS C:\Windows\system32> Get-SmbMultiChannelConnection						
Server Name	Selected Client IP	Server IP	Client Interface Index	Server Interface Index	RSS Capable	Client RDMA Capable
27-3145G0509	True	12.0.0.3	12.0.0.1	16	4	False

Get-SmbMultiChannelConnection unexpected results

If your results are unexpected, perform the following steps.

1. Make sure ping works both ways.
2. Make sure the firewall is not blocking SMB connection initiation. Specifically, enable the firewall rule for SMB Direct port 5445 for iWARP and 445 for ROCE.

Get-SmbClientNetworkInterface

You can use the following command to verify that the virtual NIC you enabled for RDMA is reported as RDMA-capable by SMB.

```
Get-SmbClientNetworkInterface
```

Get-SmbClientNetworkInterface expected results

Virtual NIC that was enabled for RDMA must be seen as RDMA capable by SMB.

PS C:\Windows\system32> Get-SmbClientNetworkInterface					
Interface Index	RSS Capable	RDMA Capable	Speed	IpAddresses	Friendly Name
7	False	False	0 bps	{fe80::7cde:94cf:20ef:c051}	NIC1
5	False	False	0 bps	{fe80::3994:161e:b2c7:b957}	NIC2
3	False	False	0 bps	{fe80::618e:4643:9047:29bc}	NIC3
6	False	False	0 bps	{fe80::9ef:92df:685:2979}	NIC4
20	True	True	10 Gbps	{10.10.1.1, 10.10.2.1, 10.10.1.1, 10.10.2.1, 10.254.7.181}	Local Area Connection 2
25	True	False	20 Gbps	{2001:4898:d8:a1d9:f10e:9a6a:6530:66ce, fe80::f10e:9a6a:6530:66ce, 10.248.138.227}	vEthernet (NS-00)
45	False	False	100 Kbps	{fe80::5eff:10.10.2.11}	isatap_{69A700B9-98B0-4DB0-B6E8-0ED7082BC062}
36	True	True	20 Gbps	{10.10.1.1}	vEthernet (vNIC2)
41	True	True	20 Gbps	{10.10.2.1}	vEthernet (vNIC4)

Get-SmbClientNetworkInterface unexpected results

If your results are unexpected, perform the following steps.

1. Make sure ping works both ways.
2. Make sure firewall is not blocking SMB connection initiation.

vstat (Mellanox specific)

If you are using Mellanox network adapters, you can use the **vstat** command to verify the RDMA over Converged

Ethernet (RoCE) version on Hyper-V nodes.

vstat expected results

The RoCE version on both nodes must be the same. This is also a good way to verify that the firmware version on both nodes is latest.

```
PS C:\Windows\system32> vstat
hca_idx=0
uplink={BUS=PCI_E Gen3, SPEED=8.0 Gbps, WIDTH=x8, CAPS=8.0*x8}
MSI-X={ENABLED=1, SUPPORTED=128, GRANTED=42, ALL_MASKED=N}
vendor_id=0x02c9
vendor_part_id=4103
hw_ver=0x0
Fw_ver=2.33.5100
PSID-MI-1030111023
node_guid=f452:1403:0074:e540
num_phys_ports=2
port=1
port_guid=f652:14ff:fe74:e540
port_state=PORT_DOWN (1)
link_speed=NA
link_width=NA
rate=NA
port_phys_state=DISABLED (3)
active_speed=1.00 Gbps
sm_lid=0x0000
port_lid=0x0000
port_lmc=0x0
transport=RoCE v1.25
max_mtu=2048 (4)
active_mtu=256 (1)

port=2
port_guid=f652:14ff:fe74:e541
port_state=PORT_ACTIVE (4)
link_speed=NA
link_width=NA
rate=40.00 Gbps
port_phys_state=LINK_UP (5)
active_speed=40.00 Gbps
sm_lid=0x0000
port_lid=0x0000
port_lmc=0x0
transport=RoCE v1.25
max_mtu=2048 (4)
active_mtu=1024 (3)
GID[0]=0000:0000:0000:0000:0000:ffff:0c00:0003
```

vstat unexpected results

If your results are unexpected, perform the following steps.

1. Set correct RoCE version using Set-MlnxDriverCoreSetting
2. Install the latest firmware from Mellanox website.

Perfmon Counters

You can review counters in Performance Monitor to verify the RDMA activity of your configuration.

RDMA Activity	Hyper-V Virtual Ethernet Adapter #2
RDMA Accepted Connections	0.000
RDMA Active Connections	2.000
RDMA Completion Queue Errors	0.000
RDMA Connection Errors	0.000
RDMA Failed Connection Attempts	0.000
RDMA Inbound Bytes/sec	1,837,340,425
RDMA Inbound Frames/sec	1,769,090,412
RDMA Initiated Connections	2.000
RDMA Outbound Bytes/sec	7,444,269,125
RDMA Outbound Frames/sec	115,440,896

Related topics

- [Converged NIC Configuration with a Single Network Adapter](#)
- [Converged NIC Teamed NIC Configuration](#)
- [Physical Switch Configuration for Converged NIC](#)

Data Center Bridging (DCB)

9/1/2018 • 5 minutes to read • [Edit Online](#)

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

You can use this topic for introductory information about Data Center Bridging (DCB).

DCB is a suite of Institute of Electrical and Electronics Engineers (IEEE) standards that enable Converged Fabrics in the data center, where storage, data networking, cluster Inter-Process Communication (IPC), and management traffic all share the same Ethernet network infrastructure.

NOTE

In addition to this topic, the following DCB documentation is available

- [Install DCB in Windows Server 2016 or Windows 10](#)
- [Manage Data Center Bridging \(DCB\)](#)

DCB provides hardware-based bandwidth allocation to a specific type of network traffic, and enhances Ethernet transport reliability with the use of priority-based flow control.

Hardware-based bandwidth allocation is essential if traffic bypasses the operating system and is offloaded to a converged network adapter, which might support Internet Small Computer System Interface (iSCSI), Remote Direct Memory Access (RDMA) over Ethernet, or Fiber Channel over Ethernet (FCoE).

Priority-based flow control is essential if the upper layer protocol, such as Fiber Channel, assumes a lossless underlying transport.

DCB Protocols and Management Options

DCB consists of the following set of protocols.

- Enhanced Transmission Service (ETS) – IEEE 802.1Qaz, which builds on the 802.1P and 802.1Q standards
- Priority Flow Control (PFS), IEEE 802.1Qbb
- DCB Exchange Protocol (DCBX), IEEE 802.1AB, as extended in the 802.1Qaz standard.

The DCBX protocol allows you to configure DCB on a switch, which can then automatically configure an end device, such as a computer running Windows Server 2016.

If you prefer to manage DCB from a switch, you don't need to install DCB as a feature in Windows Server 2016, however this approach includes some limitations.

Because DCBX can only inform the host network adapter of ETS class sizes and PFC enablement, however, Windows Server hosts usually require that DCB is installed so that traffic is mapped to ETS classes.

Windows applications are usually not designed to participate in DCBX exchanges. Because of this, the host must be configured separately from the network switches, but with identical settings.

If you do choose to manage DCB from a switch, you can still view the configurations in Windows Server 2016 by using Windows PowerShell commands.

Important DCB functionality

Following is a list that summarizes the functionality that is provided by DCB.

1. Provides interoperability between DCB-capable network adapters and DCB-capable switches.
2. Provides a lossless Ethernet transport between a computer running Windows Server 2016 and its neighbor switch by turning on priority-based flow control on the network adapter.
3. Provides the ability to allocate bandwidth to a Traffic Control (TC) by percentage, where the TC might consist of one or more classes of traffic that are differentiated by 802.1p traffic class (priority) indicators.
4. Enables server administrators or network administrators to assign an application to a particular traffic class or priority based on well-known protocols, well-known TCP/UDP port, or NetworkDirect port used by that application.
5. Provides DCB management through Windows Server 2016 Windows Management Instrumentation (WMI) and Windows PowerShell. For more information, see the section [Windows PowerShell Commands for DCB](#) later in this topic, in addition to the following topics.
 - [System-Provided DCB Components](#)
 - [NDIS QoS Requirements for Data Center Bridging](#)
6. Provides DCB management through Windows Server 2016 Group Policy.
7. Supports the coexistence of Windows Server 2016 Quality of Service (QoS) solutions.

NOTE

Before using any RDMA over Converged Ethernet (RoCE) version of RDMA, you must enable DCB. While not required for Internet Wide Area RDMA Protocol (iWARP) networks, testing has determined that all Ethernet-based RDMA technologies work better with DCB. Because of this, you should consider using DCB for iWARP RDMA deployments. For more information, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#).

Practical applications of DCB

Many organizations have large Fiber Channel (FC) storage area network (SAN) installations for storage service. FC SAN requires special network adapters on servers and FC switches in the network. These organizations typically also use Ethernet network adapters and switches.

In most cases, FC hardware is significantly more expensive to deploy than Ethernet hardware, which results in large capital expenditures. In addition, the requirement for separate Ethernet and FC SAN network adapter and switch hardware requires additional space, power, and cooling capacity in a datacenter, which results in additional, ongoing operational expenditures.

From a cost perspective, it is advantageous for many organizations to merge their FC technology with their Ethernet-based hardware solution to provide both storage and data networking services.

Using DCB for an Ethernet-based converged fabric for storage and data networking

For organizations that already have a large FC SAN but want to migrate away from additional investment in the FC technology, DCB allows you to build an Ethernet based converged fabric for both storage and data networking. A DCB converged fabric can reduce the future total cost of ownership (TCO) and simplify management.

For hosters who have already adopted, or who plan to adopt, iSCSI as their storage solution, DCB can provide hardware-assisted bandwidth reservation for iSCSI traffic to ensure performance isolation. And unlike other proprietary solutions, DCB is standards-based - and therefore relatively easy to deploy and manage in a heterogeneous network environment.

A Windows Server 2016-based implementation of DCB alleviates many of the issues that can occur when converged fabric solutions are provided by multiple original equipment manufacturers (OEMs).

Implementations of proprietary solutions provided by multiple OEMs might not interoperate with one another, might be difficult to manage, and will typically have different software maintenance schedules.

By contrast, Windows Server 2016 DCB is standards-based, and you can deploy and manage DCB in a heterogeneous network.

Windows PowerShell Commands for DCB

There are DCB Windows PowerShell commands for both Windows Server 2016 and Windows Server 2012 R2. You can use all of the commands for Windows Server 2012 R2 in Windows Server 2016.

Windows Server 2016 Windows PowerShell Commands for DCB

The following topic for Windows Server 2016 provides Windows PowerShell cmdlet descriptions and syntax for all Data Center Bridging (DCB) Quality of Service (QoS)-specific cmdlets. It lists the cmdlets in alphabetical order based on the verb at the beginning of the cmdlet.

- [DcbQoS Module](#)

Windows Server 2012 R2 Windows PowerShell Commands for DCB

The following topic for Windows Server 2012 R2 provides Windows PowerShell cmdlet descriptions and syntax for all Data Center Bridging (DCB) Quality of Service (QoS)-specific cmdlets. It lists the cmdlets in alphabetical order based on the verb at the beginning of the cmdlet.

- [Data Center Bridging \(DCB\) Quality of Service \(QoS\) Cmdlets in Windows PowerShell](#)

Install Data Center Bridging (DCB) in Windows Server 2016 or Windows 10

9/1/2018 • 3 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to install DCB in Windows Server 2016 or Windows 10.

Prerequisites for using DCB

Following are the prerequisites for configuring and managing DCB.

Install a compatible operating system

You can use the DCB commands from this guide in the following operating systems.

- Windows Server (Semi-Annual Channel)
- Windows Server 2016
- Windows 10 (all versions)

The following operating systems include previous versions of DCB that are not compatible with the commands that are used in DCB documentation for Windows Server 2016 and Windows 10.

- Windows Server 2012 R2
- Windows Server 2012

Hardware requirements

Following is a list of hardware requirements for DCB.

- DCB-capable Ethernet network adapter(s) must be installed in computers that are providing Windows Server 2016 DCB.
- DCB-capable hardware switches must be deployed on your network.

Install DCB in Windows Server 2016

You can use the following sections to install DCB on a computer running Windows Server 2016.

Administrative Credentials

To perform these procedures, you must be a member of **Administrators**.

Install DCB Using Windows PowerShell

You can use the following procedure to install DCB by using Windows PowerShell.

1. On a computer running Windows Server 2016, click **Start**, then right-click the Windows PowerShell icon. A menu appears. In the menu, click **More**, and then click **Run as administrator**. If prompted, type the credentials for an account that has Administrator privileges on the computer. Windows PowerShell opens with Administrator privileges.
2. Type the following command, and then press ENTER.

```
Install-WindowsFeature -Name Data-Center-Bridging -IncludeManagementTools
```

Install DCB Using Server Manager

You can use the following procedure to install DCB by using Server Manager.

NOTE

After you perform the first step in this procedure, the **Before You Begin** page of the Add Roles and Features Wizard is not displayed if you have previously selected **Skip this page by default** when the Add Roles and Features Wizard was run. If the **Before You Begin** page is not displayed, skip from step 1 to step 3.

1. On DC1, in Server Manager, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features Wizard opens.
2. In **Before You Begin**, click **Next**.
3. In **Select Installation Type**, ensure that **Role-Based or feature-based installation** is selected, and then click **Next**.
4. In **Select destination server**, ensure that **Select a server from the server pool** is selected. In **Server Pool**, ensure that the local computer is selected. Click **Next**.
5. In **Select server roles**, click **Next**.
6. In **Select features**, in **Features**, click **Data Center Bridging**. A dialog box opens to ask if you want to add DCB required features. Click **Add Features**.
7. In **Select features**, click **Next**.
8. In **Confirm installation selections**, click **Install**. The **Installation progress** page displays status during the installation process. After the message appears stating that installation succeeded, click **Close**.

Configure the kernel debugger to allow QoS (Optional)

By default, kernel debuggers block NetQoS. Regardless of the method that you used to install DCB, if you have a kernel debugger installed in the computer, you must configure the debugger to allow QoS to be enabled and configured by running the following command.

```
Set-ItemProperty HKLM:\SYSTEM\CurrentControlSet\Services\NDIS\Parameters AllowFlowControlUnderDebugger -type DWORD -Value 1 -Force
```

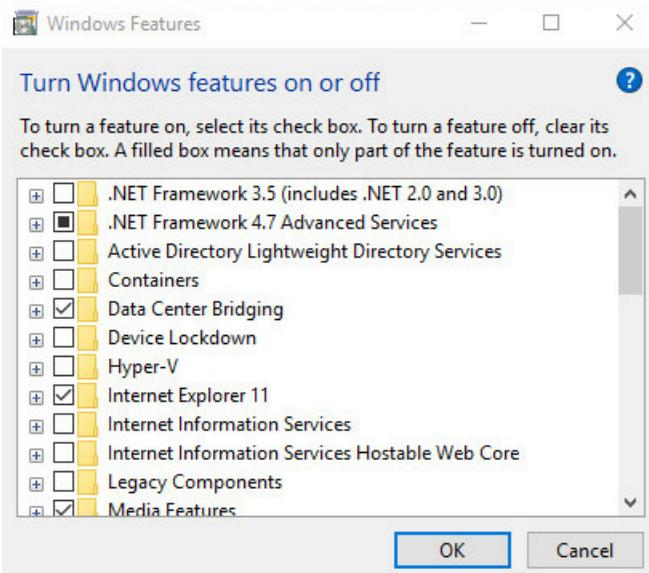
Install DCB in Windows 10

You can perform the following procedure on a Windows 10 computer.

To perform this procedure, you must be a member of **Administrators**.

Install DCB

1. Click **Start**, then scroll down to and click **Windows System**.
2. Click **Control Panel**. The **Control Panel** dialog box opens.
3. In **Control Panel**, click **View by**, and then click either **Large icons** or **Small icons**.
4. Click **Programs and Features**. The Programs and Features dialog box opens.
5. In **Programs and Features**, in the left pane, click **Turn Windows features on or off**. The **Windows Features** dialog box opens.
6. In **Windows Features**, click **Data Center Bridging**, and then click **OK**.



Manage Data Center Bridging (DCB)

9/1/2018 • 10 minutes to read • [Edit Online](#)

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

This topic provides you with instructions on how to use Windows PowerShell commands to configure Data Center Bridging (DCB) on a DCB-compatible network adapter that is installed in a computer that is running either Windows Server 2016 or Windows 10.

Install DCB in Windows Server 2016 or Windows 10

For information on prerequisites for using and how to install DCB, see [Install Data Center Bridging \(DCB\) in Windows Server 2016 or Windows 10](#).

DCB configurations

Prior to Windows Server 2016, all DCB configuration was applied universally to all network adapters that supported DCB.

In Windows Server 2016, you can apply DCB configurations either to the Global Policy Store or to individual Policy Store(s). When Individual Policies are applied they override all Global Policy settings.

The configurations of traffic class, PFC and application priority assignment at the system level is not applied on network adapters until you do the following.

1. Turn the DCBX Willing bit to false
2. Enable DCB on the network adapters. See [Enable and Display DCB Settings on Network Adapters](#).

NOTE

If you want to configure DCB from a switch through DCBX, see [DCBX settings](#)

The DCBX Willing bit is described in the DCB specification. If the Willing bit on a device is set to true, the device is willing to accept configurations from a remote device through DCBX. If the Willing bit on a device is set to false, the device will reject all configuration attempts from remote devices and enforce only the local configurations.

If DCB is not installed in Windows Server 2016 the value of the Willing bit is irrelevant as far as the operating system is concerned because the operating system has no local settings apply to network adapters. After DCB is installed, the default value of the Willing bit is true. This design allows network adapters to keep whatever configurations they may have received from their remote peers.

If a network adapter doesn't support DCBX, it will never receive configurations from a remote device. It does receive configurations from the operating system, but only after the DCBX Willing bit is set to false.

Set the Willing bit

To enforce operating system configurations of traffic class, PFC, and application priority assignment on network adapters, or to simply override the configurations from remote devices — if there are any — you can run the following command.

NOTE

DCB Windows PowerShell command names include "QoS" instead of "DCB" in the name string. This is because QoS and DCB are integrated in Windows Server 2016 to provide a seamless QoS management experience.

```
Set-NetQosDcbxSetting -Willing $FALSE

Confirm
Are you sure you want to perform this action?
Set-NetQosDcbxSetting -Willing $false
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
```

To display the state of the Willing bit setting, you can use the following command:

```
Get-NetQosDcbxSetting

Willing PolicySetIfIndex IfAlias
----- -----
False Global
```

DCB Configuration on Network Adapters

Enabling DCB on a network adapter allows you to see the configuration propagated from a switch to the network adapter.

DCB configurations include the following steps.

1. Configure DCB settings at the system level, which includes:
 - a. Traffic Class Management
 - b. Priority Flow Control (PFC) Settings
 - c. Application Priority Assignment
 - d. DCBX settings
2. Configure DCB on the network adapter.

DCB Traffic Class management

Following are example Windows PowerShell commands for Traffic Class management.

Create a Traffic Class

You can use the **New-NetQosTrafficClass** command to create a traffic class.

```
New-NetQosTrafficClass -Name SMB -Priority 4 -BandwidthPercentage 30 -Algorithm ETS

Name Algorithm Bandwidth(%) Priority PolicySetIfIndex IfAlias
----- -----
SMB ETS 30 4Global
```

By default, all 802.1p values are mapped to a default traffic class, which has 100% of the bandwidth of the physical link. The **New-NetQosTrafficClass** command creates a new traffic class, to which any packet that is tagged with 802.1p priority value 4 is mapped. The Transmission Selection Algorithm (TSA) is ETS and has 30% of the bandwidth.

You can create up to 7 new traffic classes. Including the default traffic class, there can be at most 8 traffic classes in the system. However, a DCB capable network adapter might not support that many traffic classes in the hardware. If you create more traffic classes than can be accommodated on a network adapter and you enable DCB on that network adapter, the miniport driver reports an error to the operating system. The error is logged in the Event log.

The sum of the bandwidth reservations for all created traffic classes may not exceed 99% of the bandwidth. The default traffic class always has at least 1% of the bandwidth reserved for itself.

Display Traffic Classes

You can use the **Get-NetQosTrafficClass** command to view traffic classes.

```
Get-NetQosTrafficClass

NameAlgorithm Bandwidth(%) Priority PolicySetIfIndex IfAlias
-----
[Default]   ETS    70  0-3,5-7 Global
SMB ETS    30    4Global
```

Modify a Traffic Class

You can use the **Set-NetQosTrafficClass** command to create a traffic class.

```
Set-NetQosTrafficClass -Name SMB -BandwidthPercentage 50
```

You can then use the **Get-NetQosTrafficClass** command to view settings.

```
Get-NetQosTrafficClass

NameAlgorithm Bandwidth(%) Priority PolicySetIfIndex IfAlias
-----
[Default]   ETS    50  0-3,5-7 Global
SMB ETS    50    4Global
```

After you create a traffic class, you can change its settings independently. The settings you can change include:

1. Bandwidth allocation (-BandwidthPercentage)
2. TSA (-Algorithm)
3. Priority mapping (-Priority)

Remove a Traffic Class

You can use the **Remove-NetQosTrafficClass** command to delete a traffic class.

IMPORTANT

You cannot remove the default traffic class.

```
Remove-NetQosTrafficClass -Name SMB
```

You can then use the **Get-NetQosTrafficClass** command to view settings.

```
Get-NetQosTrafficClass

NameAlgorithm Bandwidth(%) Priority PolicySetIfIndex IfAlias
----- -----
[Default] ETS 100 0-7 Global
```

After you remove a traffic class, the 802.1p value mapped to that traffic class is remapped to the default traffic class. Any bandwidth that was reserved for a traffic class is returned to the default traffic class allocation when the traffic class is removed.

Per-Network Interface Policies

All of the above examples set Global policies. Following are examples of how you can set and get per-NIC policies.

The "PolicySet" field changes from Global to AdapterSpecific. When AdapterSpecific policies are shown, the Interface Index (ifIndex) and Interface Name (ifAlias) are also displayed.

```
PS C:\> Get-NetQosTrafficClass

Name Algorithm Bandwidth(%) Priority PolicySet IfIndex IfAlias
---- ----- ----- -----
[Default] ETS 100 0-7 Global

PS C:\> Get-NetQosTrafficClass -InterfaceAlias M1

Name Algorithm Bandwidth(%) Priority PolicySet IfIndex IfAlias
---- ----- ----- -----
[Default] ETS 100 0-7 AdapterSpecific 4 M1

PS C:\> New-NetQosTrafficClass -Name SMBGlobal -BandwidthPercentage 30 -Priority 4 -Algorithm ETS

Name Algorithm Bandwidth(%) Priority PolicySet IfIndex IfAlias
---- ----- ----- -----
SMBGlobal ETS 30 4 Global

PS C:\> New-NetQosTrafficClass -Name SMBforM1 -BandwidthPercentage 30 -Priority 4 -Algorithm ETS -InterfaceAlias M1

Name Algorithm Bandwidth(%) Priority PolicySet IfIndex IfAlias
---- ----- ----- -----
SMBforM1 ETS 30 4 AdapterSpecific 4 M1

PS C:\> Get-NetQosTrafficClass

Name Algorithm Bandwidth(%) Priority PolicySet IfIndex IfAlias
---- ----- ----- -----
[Default] ETS 70 0-3,5-7 Global
SMBGlobal ETS 30 4 Global

PS C:\> Get-NetQosTrafficClass -InterfaceAlias M1

Name Algorithm Bandwidth(%) Priority PolicySet IfIndex IfAlias
---- ----- ----- -----
[Default] ETS 70 0-3,5-7 AdapterSpecific 4 M1
SMBforM1 ETS 30 4 AdapterSpecific 4 M1
```

Priority Flow Control settings:

Following are command examples for Priority Flow Control settings. These settings can also be specified for individual adapters.

Enable and Display Priority Flow Control for Global and Interface Specific use cases

```
PS C:\> Enable-NetQosFlowControl -Priority 4  
PS C:\> Enable-NetQosFlowControl -Priority 3 -InterfaceAlias M1  
PS C:\> Get-NetQosFlowControl
```

Priority	Enabled	PolicySet	IfIndex	IfAlias
0	False	Global		
1	False	Global		
2	False	Global		
3	False	Global		
4	True	Global		
5	False	Global		
6	False	Global		
7	False	Global		

```
PS C:\> Get-NetQosFlowControl -InterfaceAlias M1
```

Priority	Enabled	PolicySet	IfIndex	IfAlias
0	False	AdapterSpecific	4	M1
1	False	AdapterSpecific	4	M1
2	False	AdapterSpecific	4	M1
3	True	AdapterSpecific	4	M1
4	False	AdapterSpecific	4	M1
5	False	AdapterSpecific	4	M1
6	False	AdapterSpecific	4	M1
7	False	AdapterSpecific	4	M1

Disable Priority Flow Control (Global and Interface Specific)

```

PS C:\> Disable-NetQosFlowControl -Priority 4
PS C:\> Disable-NetQosFlowControl -Priority 3 -InterfaceAlias m1
PS C:\> Get-NetQosFlowControl

Priority Enabled PolicySet IfIndex IfAlias
----- ----- -----
0 False Global
1 False Global
2 False Global
3 False Global
4 False Global
5 False Global
6 False Global
7 False Global

PS C:\> Get-NetQosFlowControl -InterfaceAlias M1

Priority Enabled PolicySet IfIndex IfAlias
----- ----- -----
0 False AdapterSpecific 4 M1
1 False AdapterSpecific 4 M1
2 False AdapterSpecific 4 M1
3 False AdapterSpecific 4 M1
4 False AdapterSpecific 4 M1
5 False AdapterSpecific 4 M1
6 False AdapterSpecific 4 M1
7 False AdapterSpecific 4 M1

```

Application Priority assignment

Following are examples of Priority assignment.

Create QoS Policy

```

PS C:\> New-NetQosPolicy -Name "SMB Policy" -PriorityValue8021Action 4

Name      : SMB Policy
Owner     : Group Policy (Machine)
NetworkProfile : All
Precedence   : 127
JobObject    :
PriorityValue : 4

```

The previous command creates a new policy for SMB. –SMB is an inbox filter that matches TCP port 445 (reserved for SMB). If a packet is sent to TCP port 445 it will be tagged by the operating system with 802.1p value of 4 before the packet is passed to a network miniport driver.

In addition to –SMB, other default filters include –iSCSI (matching TCP port 3260), -NFS (matching TCP port 2049), -LiveMigration (matching TCP port 6600), -FCOE (matching EtherType 0x8906) and –NetworkDirect.

NetworkDirect is an abstract layer we create on top of any RDMA implementation on a network adapter. – NetworkDirect must be followed by a Network Direct port.

In addition to the default filters, you can classify traffic by application's executable name (as in the first example below), or by IP address, port, or protocol (as shown in the second example):

By executable name

```
PS C:\> New-NetQosPolicy -Name background -AppPathNameMatchCondition "C:\Program files (x86)\backup.exe" -PriorityValue8021Action 1

Name      : background
Owner     : Group Policy (Machine)
NetworkProfile : All
Precedence   : 127
AppPathName  : C:\Program files (x86)\backup.exe
JobObject    :
PriorityValue : 1
```

By IP address port or protocol

```
PS C:\> New-NetQosPolicy -Name "Network Management" -IPDstPrefixMatchCondition 10.240.1.0/24 -IPProtocolMatchCondition both -NetworkProfile all -PriorityValue8021Action 7

Name      : Network Management
Owner     : Group Policy (Machine)
NetworkProfile : All
Precedence   : 127
JobObject    :
IPProtocol   : Both
IPDstPrefix  : 10.240.1.0/24
PriorityValue : 7
```

Display QoS Policy

```
PS C:\> Get-NetQosPolicy

Name      : background
Owner     : Group Policy (Machine)
NetworkProfile : All
Precedence   : 127
AppPathName  : C:\Program files (x86)\backup.exe
JobObject    :
PriorityValue : 1

Name      : Network Management
Owner     : Group Policy (Machine)
NetworkProfile : All
Precedence   : 127
JobObject    :
IPProtocol   : Both
IPDstPrefix  : 10.240.1.0/24
PriorityValue : 7

Name      : SMB Policy
Owner     : Group Policy (Machine)
NetworkProfile : All
Precedence   : 127
JobObject    :
PriorityValue : 4
```

Modify QoS Policy

You can modify QoS policies as shown below.

```

PS C:\> Set-NetQosPolicy -Name "Network Management" -IPSrcPrefixMatchCondition 10.235.2.0/24 -
IPProtocolMatchCondition both -PriorityValue8021Action 7
PS C:\> Get-NetQosPolicy

Name      : Network Management
Owner     : Group Policy (Machine)
NetworkProfile : All
Precedence   : 127
JobObject    :
IPProtocol   : Both
IPSrcPrefix  : 10.235.2.0/24
IPDstPrefix  : 10.240.1.0/24
PriorityValue : 7

```

Remove QoS Policy

```

PS C:\> Remove-NetQosPolicy -Name "Network Management"

Confirm
Are you sure you want to perform this action?
Remove-NetQosPolicy -Name "Network Management" -Store GPO:localhost
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"): y

```

DCB configuration on network adapters

DCB configuration on network adapters is independent of DCB configuration at the system level described above.

Regardless of whether DCB is installed in Windows Server 2016, you can always run the following commands.

If you configure DCB from a switch and rely on DCBX to propagate the configurations to network adapters, you can examine what configurations are received and enforced on the network adapters from the operating system side after you enable DCB on the network adapters.

Enable and Display DCB Settings on Network Adapters

```

PS C:\> Enable-NetAdapterQos M1
PS C:\> Get-NetAdapterQos

Name      : M1
Enabled   : True
Capabilities      : Hardware Current
                  -----
                  MacSecBypass : NotSupported NotSupported
                  DcbxSupport  : None      None
                  NumTCs(Max/ETS/PFC) : 8/8/8  8/8/8

OperationalTrafficClasses : TC TSA Bandwidth Priorities
                           -----
                           0 ETS  70%  0-3,5-7
                           1 ETS  30%  4

OperationalFlowControl   : All Priorities Disabled
OperationalClassifications : Protocol Port/Type Priority
                           -----
                           Default 1

```

Disable DCB on Network Adapters

```
PS C:\> Disable-NetAdapterQos M1
PS C:\> Get-NetAdapterQos M1

Name      : M1
Enabled   : False
Capabilities :          Hardware    Current
                -----
MacSecBypass     : NotSupported NotSupported
DcbxSupport      : None        None
NumTCs(Max/ETS/PFC) : 8/8/8    0/0/0
```

Windows PowerShell Commands for DCB

There are DCB Windows PowerShell commands for both Windows Server 2016 and Windows Server 2012 R2. You can use all of the commands for Windows Server 2012 R2 in Windows Server 2016.

Windows Server 2016 Windows PowerShell Commands for DCB

The following topic for Windows Server 2016 provides Windows PowerShell cmdlet descriptions and syntax for all Data Center Bridging (DCB) Quality of Service (QoS)-specific cmdlets. It lists the cmdlets in alphabetical order based on the verb at the beginning of the cmdlet.

- [DcbQoS Module](#)

Windows Server 2012 R2 Windows PowerShell Commands for DCB

The following topic for Windows Server 2012 R2 provides Windows PowerShell cmdlet descriptions and syntax for all Data Center Bridging (DCB) Quality of Service (QoS)-specific cmdlets. It lists the cmdlets in alphabetical order based on the verb at the beginning of the cmdlet.

- [Data Center Bridging \(DCB\) Quality of Service \(QoS\) Cmdlets in Windows PowerShell](#)

Virtual Receive Side Scaling (vRSS)

9/21/2018 • 3 minutes to read • [Edit Online](#)

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

In this topic, you learn about Virtual Receive Side Scaling (vRSS) and how to configure a virtual network adapter to load balance incoming network traffic across multiple logical processor cores in a VM. You can also use vRSS to configure multiple physical cores for a host virtual Network Interface Card (vNIC).

This configuration allows the load from a virtual network adapter to be distributed across multiple virtual processors in a virtual machine (VM), allowing the VM to process more network traffic more rapidly than it can with a single logical processor.

TIP

You can use vRSS in VMs on Hyper-V hosts that have multiple processors, a single multiple core processor, or more than one multiple core processors installed and configured for VM use.

vRSS is compatible with all other Hyper-V networking technologies. vRSS is dependent on Virtual Machine Queue (VMQ) in the Hyper-V host and RSS in the VM or on the host vNIC.

By default, Windows Server enables vRSS, but you can disable it in a VM by using Windows PowerShell. For more information, see [Manage vRSS](#) and [Windows PowerShell Commands for RSS and vRSS](#).

Operating System Compatibility

You can use RSS on any multiprocessor or multicore computer - or vRSS on any multiprocessor or multicore VM - that is running Windows Server 2016.

Multiprocessor or multicore VMs that are running the following Microsoft operating systems also support vRSS.

- Windows Server 2016
- Windows 10 Pro or Enterprise
- Windows Server 2012 R2
- Windows 8.1 Pro or Enterprise
- Windows Server 2012 with the Windows Server 2012 R2 integration components installed.
- Windows 8 with the Windows Server 2012 R2 integration components installed.

For information about vRSS support for VMs running FreeBSD or Linux as a guest operating system on Hyper-V, see [Supported Linux and FreeBSD virtual machines for Hyper-V on Windows](#).

Hardware requirements

Following are the hardware requirements for vRSS.

- Physical network adapters must support Virtual Machine Queue (VMQ). If VMQ is disabled or not supported, then vRSS is disabled for the Hyper-V host and any VMs configured on the host.
- Network adapters must have a link speed of 10 Gbps or more.
- Hyper-V hosts must be configured with multiple processors or at least one multi-core processor to use vRSS.
- Virtual machines (VMs) must be configured to use two or more logical processors.

Use Case Scenarios

The following two use case scenarios depict common usage of vRSS for processor load balancing and software load balancing.

Processor load balancing

Anthony, a network administrator, is setting up a new Hyper-V host with two network adapter that supports Single Root Input-Output Virtualization (SR-IOV). He deploys Windows Server 2016 to host a VM file server.

After installing the hardware and software, Anthony configures a VM to use eight virtual processors and 4096 MB of memory. Unfortunately, Anthony does not have the option of turning on SR-IOV because his VMs rely on policy enforcement through the virtual switch he created with Hyper-V Virtual Switch manager. Because of this, Anthony decides to use vRSS instead of SR-IOV.

Initially, Anthony assigns four virtual processors by using Windows PowerShell to be available for use with vRSS. The use of the file server after a week appeared to be quite popular, so Anthony checks the performance of the VM. He discovers full utilization of the four virtual processors.

Because of this, Anthony decides to add processors to the VM for use by vRSS. He assigns two more virtual processors to the VM, which are automatically available to vRSS to help handle the large network load. His efforts result in better performance for the VM file server, with the six processors efficiently handling the network traffic load.

Software load balancing

Sandra, a network administrator, is setting up a single high-performance VM on one of her systems to act as a software load balancer. She has assigned all available logical processors to this single VM.

After installing Windows Server, she uses vRSS to get parallel network traffic processing on multiple logical processors in the VM. Because Windows Server enables vRSS, Sandra doesn't have to make any configuration changes.

Related topics

- [Plan the Use of vRSS](#)
 - [Enable vRSS on a Virtual Network Adapter](#)
 - [Manage vRSS](#)
 - [vRSS Frequently Asked Questions](#)
 - [Windows PowerShell Commands for RSS and vRSS](#)
-

Plan the Use of vRSS

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

In Windows Server 2016, vRSS is enabled by default, however you must prepare your environment to allow vRSS to function correctly in a virtual machine (VM) or on a host virtual adapter (vNIC). In Windows Server 2012 R2, vRSS was disabled by default.

When you plan and prepare the use of vRSS, ensure that:

- The physical network adapter is compatible with Virtual Machine Queue (VMQ) and has a link speed of 10 Gbps or more.
- VMQ is enabled on the physical NIC and on the Hyper-V Virtual Switch port
- There is no Single Root Input-Output Virtualization (SR-IOV) configured for the VM.
- NIC Teaming is configured correctly.
- The VM has multiple logical processors (LPs).

NOTE

vRSS is also enabled by default for any host vNICs that have RSS enabled.

Following is additional information you need to complete these preparation steps.

1. **Network Adapter Capacity.** Verify that the network adapter is compatible with Virtual Machine Queue (VMQ) and has a link speed of 10 Gbps or more. If the link speed is less than 10 Gbps, the Hyper-V Virtual Switch disables VMQ by default, even though it still shows VMQ as enabled in the results of the Windows PowerShell command **Get-NetAdapterVmq**. One method you can use to verify that VMQ is enabled or disabled is to use the command **Get-NetAdapterVmqQueue**. If VMQ is disabled, the results of this command show that there is no QueueID assigned to the VM or host virtual network adapter.
2. **Enable VMQ.** Verify that VMQ is enabled on the host machine. vRSS does not work if the host does not support VMQ. You can verify that VMQ is enabled by running **Get-VMSwitch** and finding the adapter that the virtual switch is using. Next, run **Get-NetAdapterVmq** and ensure that the adapter is shown in the results and has VMQ enabled.
3. **Absence of SR-IOV.** Verify that a Single Root Input-Output Virtualization (SR-IOV) Virtual Function (VF) driver is not attached to the VM network interface. You can verify this by using the **Get-NetAdapterSriov** command. If a VF driver is loaded, RSS uses the scaling settings from this driver instead of those configured by vRSS. If the VF driver does not support RSS, then vRSS is disabled.
4. **NIC Teaming Configuration.** If you are using NIC Teaming, it is important that you properly configure VMQ to work with the NIC Teaming settings. For detailed information about NIC Teaming deployment and management, see [NIC Teaming](#).
5. **Number of LPs.** Verify that the VM has more than one logical processor (LP). vRSS relies on RSS in the VM or on the Hyper-V host to load balance received traffic to multiple LPs for parallel processing. You can observe how many LPs your VM has by running the Windows PowerShell command **Get-VMProcessor** in the host. After you run the command, you can observe the Count column entry for the number of LPs.

The host vNIC always has access to all of the physical processors; to configure the host vNIC to use a specific

number of processors, use the settings **-BaseProcessorNumber** and **-MaxProcessors** when you run the **Set-NetAdapterRss** Windows PowerShell command.

Enable vRSS on a Virtual Network Adapter

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

Virtual RSS (vRSS) requires Virtual Machine Queue (VMQ) support from the physical adapter. If VMQ is disabled or is not supported then Virtual Receive-side scaling is disabled.

For more information, see [Plan the Use of vRSS](#).

Enable vRSS on a VM

Use the following procedures to enable vRSS by using either Windows PowerShell or Device Manager.

- Device Manager
- Windows PowerShell

Device Manager

You can use this procedure to enable vRSS by using Device Manager.

NOTE

The first step in this procedure is specific to VMs that are running Windows 10 or Windows Server 2016. If your VM is running a different operating system, you can open Device Manager by first opening Control Panel, then locating and opening Device Manager.

1. On the VM taskbar, in **Type here to search**, type **device**.
2. In the search results, click **Device Manager**.
3. In Device Manager, click to expand **Network adapters**.
4. Right-click the network adapter you want to configure, and then click **Properties**.

The network adapter **Properties** dialog box opens.

5. In network adapter **Properties**, click the **Advanced** tab.
6. In **Property**, scroll down to and click **Receive-side scaling**.
7. Ensure that the selection in **Value** is **Enabled**.
8. Click **OK**.

NOTE

On the **Advanced** tab, some network adapters also display the number of RSS queues that are supported by the adapter.

Windows PowerShell

Use the following procedure to enable vRSS by using Windows PowerShell.

1. On the virtual machine, open **Windows PowerShell**.

2. Type the following command, ensuring that you replace the *AdapterName* value for the **-Name** parameter with the name of the network adapter that you want to configure, and then press ENTER.

```
Enable-NetAdapterRSS -Name "AdapterName"
```

TIP

Alternately, you can use the following command to enable vRSS.

```
Set-NetAdapterRSS -Name "AdapterName" -Enabled $True
```

For more information, see [Windows PowerShell Commands for RSS and vRSS](#).

Manage vRSS

9/21/2018 • 2 minutes to read • [Edit Online](#)

In this topic, you use the Windows PowerShell commands to manage vRSS in virtual machines (VMs) and on Hyper-V hosts.

NOTE

For more information about the commands mentioned in this topic, see [Windows PowerShell Commands for RSS and vRSS](#).

VMQ on Hyper-V Hosts

On the Hyper-V host, you must use the keywords that control the VMQ processors.

View the current settings:

```
Get-NetAdapterVmq
```

Configure the VMQ settings:

```
Set-NetAdapterVmq
```

vRSS on Hyper-V switch ports

On the Hyper-V host, you must also enable vRSS on the Hyper-V Virtual Switch port.

View the current settings:

```
Get-VMNetworkAdapter <vm-name> | fl  
Get-VMNetworkAdapter -ManagementOS | fl
```

Both of the following settings should be **True**.

- VrssEnabledRequested: True
- VrssEnabled: True

IMPORTANT

Under some resource limitation conditions, a Hyper-V Virtual Switch port might be unable to have this feature enabled. This is a temporary condition, and the feature may become available at a subsequent time.

If **VrssEnabled** is **True**, then the feature is enabled for this Hyper-V Virtual Switch port—that is, for this VM or vNIC.

Configure the switch port vRSS settings:

```
Set-VMNetworkAdapter <vm-name> -VrssEnabled $TRUE  
Set-VMNetworkAdapter -ManagementOS -VrssEnabled $TRUE
```

vRSS in VMs and host vNICs

You can use the same commands used for native RSS to configure vRSS settings in VMs and host vNICs, which is also the way to enable RSS on host vNICs.

View the current settings:

```
Get-NetAdapterRSS
```

Configure vRSS settings:

```
Set-NetAdapterRss
```

NOTE

Setting the profile inside the VM does not impact the scheduling of the work. Hyper-V makes all the scheduling decisions and ignores the profile inside the VM.

Disable vRSS

You can disable vRSS to disable any of the previously mentioned settings.

- Disable VMQ for the physical NIC or the VM.

Caution

Disabling VMQ on the physical NIC severely impacts the ability of your Hyper-V host to handle incoming packets.

- Disable vRSS for a VM on the Hyper-V Virtual Switch port on the Hyper-V host.

```
Set-VMNetworkAdapter <vm-name> -VrssEnabled $FALSE
```

- Disable vRSS for a host vNIC on the Hyper-V Virtual Switch port on the Hyper-V host.

```
Set-VMNetworkAdapter -ManagementOS -VrssEnabled $FALSE
```

- Disable RSS in the VM (or host vNIC) inside the VM (or on the host)

```
Disable-NetAdapterRSS *
```

vRSS Frequently Asked Questions

9/21/2018 • 2 minutes to read • [Edit Online](#)

In this topic, you find some commonly asked questions and answers about using vRSS.

What are the requirements for the physical network adapters that I use with vRSS?

Network adapters must be compatible with Virtual Machine Queue (VMQ) and must have a link speed of 10 Gbps or more.

For more information, see [Plan the Use of vRSS](#).

Does vRSS work with hyper-threaded processor cores?

No. Both vRSS and VMQ ignore hyper-threaded processor cores.

Does vRSS work for host virtual NICs (vNICs)?

Yes. Use the **-ManagementOS** parameter instead of the virtual machine (VM) name on the **Set-VMNetworkAdapter** Windows PowerShell command, and **Enable-NetAdapterRss** on the host vNIC.

For more information, see [Windows PowerShell Commands for RSS and vRSS](#).

How many logical processors does a VM need to use vRSS?

VMs need two or more logical processors (LPs) to be able to use vRSS.

For more information, see [Plan the Use of vRSS](#).

Is vRSS compatible with NIC Teaming?

Yes. If you are using NIC Teaming, it is important that you properly configure VMQ to work with the NIC Teaming settings. For detailed information about NIC Teaming deployment and management, see [NIC Teaming](#).

vRSS is enabled, but how do I know if it is working?

You'll be able to tell vRSS is working by opening the task manager in your VM and viewing the virtual processor utilization. If there are multiple connections established to the VM, you can see more than one core above 0% utilization.

Because a single TCP session cannot be load balanced across multiple logical processor cores, your VM must be receiving multiple TCP sessions before you can observe whether or not vRSS is working.

If the VM is receiving multiple TCP sessions, but you do not see more than one LP core above 0% utilization, ensure that you have completed all of the preparation steps in the topic [Plan the Use of vRSS](#).

I'm looking at the host and not all of the processors are being used. It looks like every other one is being skipped.

Check to see if hyper threading is enabled. Both VMQ and vRSS are designed to skip hyper-threaded cores.

Are there different Windows PowerShell commands for RSS and vRSS?

Yes and no. While you use the same commands for both RSS in native hosts and RSS in VMs, vRSS also requires VMQ to be enabled on the physical NIC - and for the VM and vRSS to be enabled on the switch port.

For more information, see [Windows PowerShell Commands for RSS and vRSS](#).

Windows PowerShell Commands for RSS and vRSS

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

In this topic, you learn how to quickly locate technical reference information about Windows PowerShell commands for Receive Side Scaling (RSS) and virtual RSS (vRSS).

Use the following RSS commands to configure RSS on a physical computer with multiple processors or multiple cores. You can use the same commands to configure vRSS on a virtual machine (VM) that is running a supported operating system. For more information, see [Network Adapter Cmdlets in Windows PowerShell](#).

Configure VMQ

vRSS requires that VMQ is enabled and configured. You can use the following Windows PowerShell commands to manage VMQ settings.

- [Disable-NetAdapterVmq](#)
- [Enable-NetAdapterVmq](#)
- [Get-NetAdapterVmq](#)
- [Set-NetAdapterVmq](#)

Enable and configure RSS on a native host

Use the following PowerShell commands to configure RSS on a native host as well as manage RSS in a VM or on a host virtual NIC (vNIC). Some of the parameters of these commands might also affect Virtual Machine Queue (VMQ) in the Hyper-V host.

IMPORTANT

Enabling RSS in a VM or on a host vNIC is a prerequisite for enabling and using vRSS.

- [Disable-NetAdapterRss](#)
- [Enable-NetAdapterRss](#)
- [Get-NetAdapterRss](#)
- [Set-NetAdapterRss](#)

Enable vRSS on the Hyper-V Virtual Switch port

In addition to enabling RSS in the VM, vRSS requires that you enable vRSS on the Hyper-V Virtual Switch port.

Determine the present settings for vRSS and enable or disable the feature for a VM.

View the current settings:

```
Get-VMNetworkAdapter <vm-name> | fl
```

Enabled the feature:

```
Set-VMNetworkAdapter <vm-name> -VrssEnabled [$True|$False]
```

Enable or disable vRSS on a host vNIC

Determine the present settings for vRSS, and enable or disable the feature for a host vNIC.

View the current settings:

```
Get-VMNetworkAdapter -ManagementOS | fl
```

Enable or disable the feature:

```
Set-VMNetworkAdapter -ManagementOS -VrssEnabled [$True|$False]
```

Configure the scheduling mode on the Hyper-V virtual switch port

Applies to: Windows Server 2019

In Windows Server 2019, vRSS can update the logical processors used to process network traffic dynamically. Devices with supported drivers have this scheduling mode enabled by default.

Determine the present scheduling mode on a system, or modify the scheduling mode for a VM.

View the current settings:

```
Get-VMNetworkAdapter <vm-name> | Select 'VRSSQueue'
```

Set or modify the scheduling mode:

```
Set-VMNetworkAdapter <vm-name> -VrssQueueSchedulingMode [Dynamic|$StaticVrss|StaticVMQ]
```

Configure the scheduling mode on a host vNIC

Applies to: Windows Server 2019

To determine the present scheduling mode or to modify the scheduling mode for a host vNIC, use the following Windows PowerShell commands:

View the current settings:

```
Get-VMNetworkAdapter -ManagementOS | Select 'VRSSQueue'
```

Set or modify the scheduling mode:

```
Set-VMNetworkAdapter -ManagementOS -VrssQueueSchedulingMode -VrssQueueSchedulingMode  
[Dynamic|$StaticVrss|StaticVMQ]
```

Related topics

For more information, see the following reference topics.

- [Get-VMNetworkAdapter](#)
- [Set-VMNetworkAdapter](#)

For more information, see [Virtual Receive Side Scaling \(vRSS\)](#).

Resolve vRSS issues

9/21/2018 • 2 minutes to read • [Edit Online](#)

If you have completed all of the preparation steps and you still do not see vRSS load balancing traffic to the VM LPs, there are two possible issues.

1. Before you performed preparation steps, vRSS was disabled - and now must be enabled. You can run **Set-VMNetworkAdapter** to enable vRSS for the VM.

```
Set-VMNetworkAdapter <VMname> -VrssEnabled $TRUE  
Set-VMNetworkAdapter -ManagementOS -VrssEnabled $TRUE
```

2. RSS was disabled in the VM or on the host vNIC. Windows Server 2016 enables RSS by default; someone might have disabled it.

- Enabled = **True**

View the current settings:

Run the following PowerShell cmdlet in the VM(for vRSS in a VM) or on the host (for host vNIC vRSS).

```
Get-NetAdapterRss
```

Enable the feature:

To change the value from False to True, run the following PowerShell cmdlet.

```
Enable-NetAdapterRss *
```

3. If you find VMMQ is not enabled after you configure vRSS, verify the following settings on each adapter attached to the virtual switch:

- VmmqEnabled = **False**
- VmmqEnabledRequested = **True**

```
PS C:\Windows\system32> Get-VMNetworkAdapter -ManagementOS | Select *VMMQ*, *VRSS*
```

VmmqQueuePairs	:	8
VmmqQueuePairsRequested	:	16
VmmqEnabled	:	False
VmmqEnabledRequested	:	True
VrssEnabled	:	True
VrssEnabledRequested	:	True

View the current settings:

```
Get-NetAdapterAdvancedProperty -Name NICName -DisplayName 'Virtual Switch RSS'
```

Enable the feature:

```
Set-NetAdapterAdvancedProperty -Name NICName -DisplayName 'Virtual Switch RSS' -DisplayValue Enabled"
```

4. (*Windows Server 2019*) You cannot enable VMMQ (`VmmqEnabled = False`) while setting **VrssQueueSchedulingMode** to **Dynamic**. The `VrssQueueSchedulingMode` does not change to Dynamic once VMMQ is enabled.

The **VrssQueueSchedulingMode** of **Dynamic** requires driver support when VMMQ is enabled. VMMQ is an offload of the packet placement on logical processors and as such, requires driver support to leverage the dynamic algorithm. Please install the NIC vendor's driver and firmware that supports Dynamic VMMQ.

Hyper-V Virtual Switch

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

Hyper-V Virtual Switch documentation is now located in the **Virtualization** section of this library, under **Hyper-V Virtual Switch**. Go to [Hyper-V Virtual Switch](#).

IP Address Management (IPAM)

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

IP Address Management (IPAM) is an integrated suite of tools to enable end-to-end planning, deploying, managing and monitoring of your IP address infrastructure, with a rich user experience. IPAM automatically discovers IP address infrastructure servers and Domain Name System (DNS) servers on your network and enables you to manage them from a central interface.

NOTE

In addition to this topic, the following IPAM content is available.

- [What's New in IPAM](#)
- [Manage IPAM](#)
- [IP Address Management \(IPAM\) Server Cmdlets in Windows PowerShell](#)
- Video: [Windows Server 2016: DNS management in IPAM](#)

What's New in IPAM

9/1/2018 • 5 minutes to read • [Edit Online](#)

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

This topic describes the IP Address Management (IPAM) functionality that is new or changed in Windows Server 2016.

IPAM provides highly customizable administrative and monitoring capabilities for the IP address and DNS infrastructure on an Enterprise or Cloud Service Provider (CSP) network. You can monitor, audit, and manage servers running Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) by using IPAM.

Updates in IPAM Server

Following are the new and improved features for IPAM in Windows Server 2016.

FEATURE/FUNCTIONALITY	NEW OR IMPROVED	DESCRIPTION
Enhanced IP address management	Improved	IPAM capabilities are improved for scenarios such as handling IPv4 /32 and IPv6 /128 subnets and finding free IP address subnets and ranges in an IP address block.
Enhanced DNS service management	New	IPAM supports DNS resource record, conditional forwarder, and DNS zone management for both domain-joined Active Directory-integrated and file-backed DNS servers.
Integrated DNS, DHCP, and IP address (DDI) management	Improved	Several new experiences and integrated lifecycle management operations are enabled, such as visualizing all DNS resource records that pertain to an IP address, automated inventory of IP addresses based on DNS resource records, and IP address lifecycle management for both DNS and DHCP operations.
Multiple Active Directory Forest support	New	You can use IPAM to manage the DNS and DHCP servers of multiple Active Directory forests when there is a two-way trust relationship between the forest where IPAM is installed and each of the remote forests.
Purge Utilization Data	New	You can now reduce the IPAM database size by purging the IP address utilization data that is older than a date that you specify.
Windows PowerShell support for Role Based Access Control	New	You can use Windows PowerShell to set access scopes on IPAM objects.

Enhanced IP address management

The following features improve the IPAM address management capabilities.

NOTE

For the IPAM Windows PowerShell command reference, see [IP Address Management \(IPAM\) Server Cmdlets in Windows PowerShell](#).

Support for /31, /32, and /128 subnets

IPAM in Windows Server 2016 now supports /31, /32, and /128 subnets. For example, a two address subnet (/31 IPv4) may be required for a point-to-point link between switches. Also, some switches may require single loopback addresses (/32 for IPv4, /128 for IPv6).

Find free subnets with `Find-IpamFreeSubnet`

This command returns subnets that are available for allocation, given an IP block, prefix length, and number of requested subnets.

If the number of available subnets is less than the number of requested subnets, the available subnets are returned with a warning indicating that the number available is less than the number requested.

NOTE

This function does not actually allocate the subnets, it only reports their availability. However, the cmdlet output can be piped to the **Add-IpamSubnet** command to create the subnet.

For more information, see [Find-IpamFreeSubnet](#).

Find free address ranges with `Find-IpamFreeRange`

This new command returns available IP address ranges given an IP subnet, the number of addresses that are needed in the range, and the number of ranges requested.

The command searches for a continuous series of unallocated IP addresses that match the number of requested addresses. The process is repeated until the requested number of ranges is found, or until there are no more available address ranges available.

NOTE

This function does not actually allocate the ranges, it only reports their availability. However, the cmdlet output can be piped to the **Add-IpamRange** command to create the range.

For more information, see [Find-IpamFreeRange](#).

Enhanced DNS service management

IPAM in Windows Server 2016 now supports discovery of file-based, domain-joined DNS servers in an Active Directory forest in which IPAM is running.

Additionally, the following DNS functions have been added:

- DNS zones and resource records collection (other than those pertaining to DNSSEC) from DNS servers running Windows Server 2008 or later.
- Configure (create, modify, and delete) properties and operations on all types of Resource Records (other than those pertaining to DNSSEC).
- Configure (create, modify, delete) properties and operations on all types of DNS zones including Primary Secondary, and Stub zones).

- Triggered tasks on secondary and stub zones, regardless if they are forward or reverse lookup zones. For example, tasks such as **Transfer from Master** or **Transfer new copy of zone from Master**.
- Role based access control for the supported DNS configuration (DNS records and DNS zones).
- Conditional forwarders collection and configuration (create, delete, edit).

Integrated DNS, DHCP, and IP address (DDI) management

When you view an IP address in the IP Address Inventory, you have the option in the Details View to see all the DNS resource records associated with the IP address.

As part DNS resource record collection, IPAM collects the PTR records for the DNS reverse look-up zones. For all the reverse lookup zones which are mapped to any IP address range, IPAM creates the IP address records for all the PTR records belonging to that zone in the corresponding mapped IP address range. If the IP address already exists, the PTR record is simply associated with that IP address. The IP addresses are not automatically created if the reverse lookup zone is not mapped to any IP address range.

When a PTR record is created in a reverse lookup zone through IPAM, the IP address inventory is updated in the same way as described above. During subsequent collection, since the IP address will already exist in the system, the PTR record will be simply mapped with that IP address.

Multiple Active Directory Forest support

In Windows Server 2012 R2 , IPAM was able to discover and manage DNS and DHCP servers belonging to the same Active Directory forest as the IPAM server. Now you can manage DNS and DHCP servers belonging to a different AD forest when it has a two-way trust relationship with the forest where the IPAM server is installed. You can go to the **Configure Server Discovery** dialog box and add domains from the other trusted forests that you want to manage. After the servers are discovered, the management experience is the same as for the servers that belong to the same forest where IPAM is installed.

For more information, see [Manage Resources in Multiple Active Directory Forests](#)

Purge Utilization Data

Purge Utilization Data allows you to reduce the IPAM database size by deleting old IP address utilization data. To perform data deletion, you specify a date, and IPAM deletes all database entries that are older than or equal to the date you provide.

For more information, see [Purge Utilization Data](#).

Windows PowerShell support for Role Based Access Control

You can now use Windows PowerShell to configure Role Based Access Control. You can use Windows PowerShell commands to retrieve DNS and DHCP objects in IPAM and change their access scopes. Because of this, you can write Windows PowerShell scripts to assign access scopes to the following objects.

- IP address space
- IP address block
- IP address subnets
- IP address ranges
- DNS servers
- DNS zones
- DNS conditional forwarders
- DNS resource records

- DHCP servers
- DHCP superscopes
- DHCP scopes

For more information, see [Manage Role Based Access Control with Windows PowerShell](#) and [IP Address Management \(IPAM\) Server Cmdlets in Windows PowerShell](#).

Manage IPAM

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

This guide provides administration and troubleshooting information for the IP Address Management (IPAM) feature in Windows Server 2016.

In Windows Server 2016, IPAM supports DNS resource record, conditional forwarder, and DNS zone management for both domain-joined Active Directory-integrated and file-backed DNS servers. In addition, IPAM supports role-based access control and all functionality in previous versions of the technology.

This guide includes the following sections:

- [DNS Resource Record Management](#)
- [DNS Zone Management](#)
- [Manage Resources in Multiple Active Directory Forests](#)
- [Purge Utilization Data](#)
- [Role-based Access Control](#)

See Also

[IP Address Management \(IPAM\)](#)

DNS Resource Record Management

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

This topic provides information about managing DNS resource records by using IPAM.

NOTE

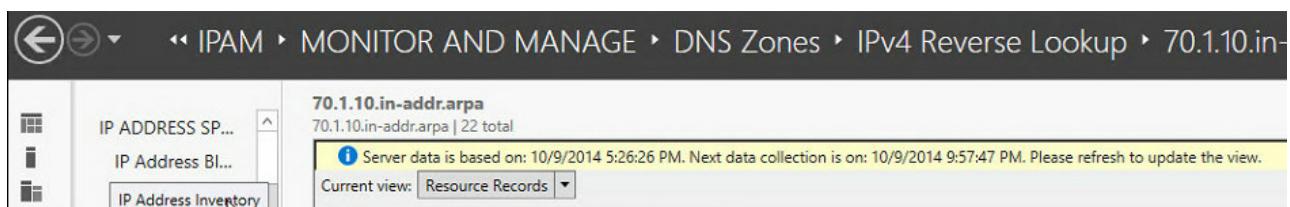
In addition to this topic, the following DNS resource record management topics are available in this section.

- [Add a DNS Resource Record](#)
- [Delete DNS Resource Records](#)
- [Filter the View of DNS Resource Records](#)
- [View DNS Resource Records for a Specific IP Address](#)

Resource record management overview

When you deploy IPAM in Windows Server 2016, you can perform server discovery to add DHCP and DNS servers to the IPAM server management console. The IPAM server then dynamically collects DNS data every six hours from the DNS servers that it is configured to manage. IPAM maintains a local database where it stores this DNS data. IPAM provides you with notification of the day and time that the server data was collected, as well as telling you the next day and time when data collection from DNS servers will occur.

The yellow status bar in the following illustration shows the user interface location of IPAM notifications.



The DNS data that is collected includes DNS zone and resource record information. You can configure IPAM to collect zone information from your preferred DNS server. IPAM collects both file-based and Active Directory zones.

NOTE

IPAM collects data solely from domain-joined Microsoft DNS servers. Third party DNS servers and non-domain joined servers are not supported by IPAM.

Following is a list of DNS resource record types that are collected by IPAM.

- AFS database
- ATM Address
- CNAME
- DHCID
- DNAME

- Host A or AAAA
- Host Information
- ISDN
- MX
- Name Servers
- Pointer (PTR)
- Responsible person
- Route Through
- Service Location
- SOA
- SRV
- Text
- Well Known Services
- WINS
- WINS-R
- X.25

In Windows Server 2016, IPAM provides integration between IP address inventory, DNS Zones, and DNS resource records:

- You can use IPAM to automatically build an IP address inventory from DNS resource records.
- You can manually create an IP address inventory from DNS A and AAAA resource records.
- You can view DNS resource records for a specific DNS zone, and filter the records based on type, IP address, resource record data, and other filtering options.
- IPAM automatically creates a mapping between IP address ranges and DNS Reverse Look-up Zones.
- IPAM creates IP addresses for the PTR records that are present in the reverse look-up zone and which are included in that IP address range. You can also manually modify this mapping if needed.

IPAM allows you to perform the following operations on resource records from the IPAM console.

- Create DNS resource records
- Edit DNS resource records
- Delete DNS resource records
- Create associated resource records

IPAM automatically logs all DNS configuration changes that you make using the IPAM console.

See Also

[Manage IPAM](#)

Add a DNS Resource Record

9/1/2018 • 2 minutes to read • [Edit Online](#)

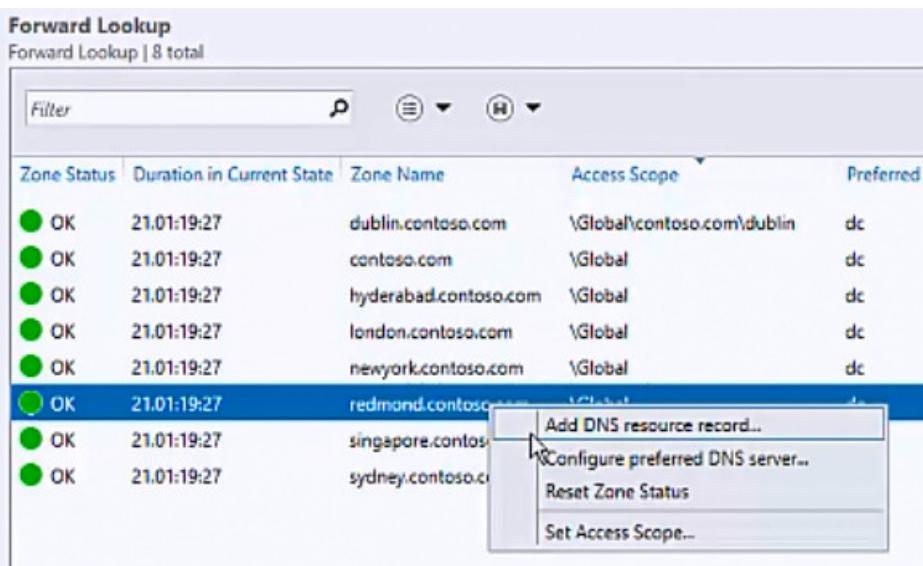
Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016

You can use this topic to add one or more new DNS resource records by using the IPAM client console.

Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To add a DNS resource record

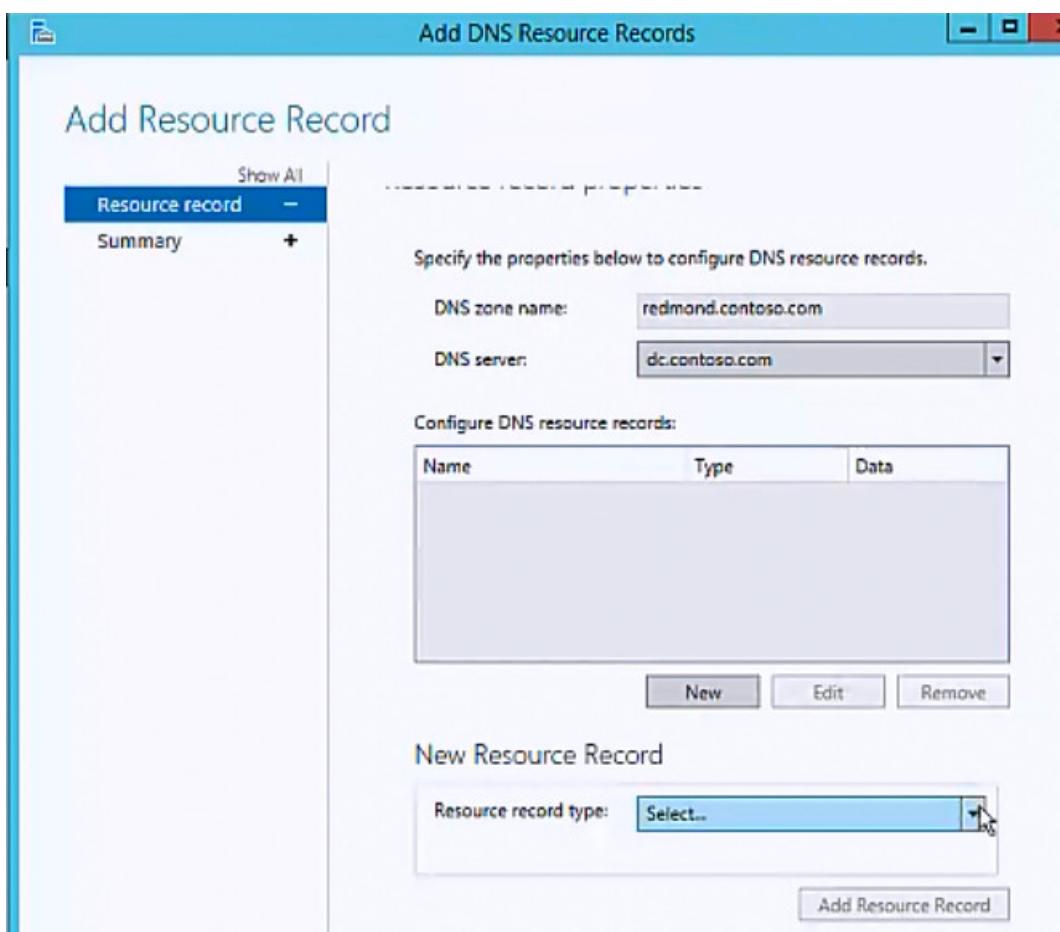
1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, in **MONITOR AND MANAGE**, click **DNS Zones**. The navigation pane divides into an upper navigation pane and a lower navigation pane.
3. In the lower navigation pane, click **Forward Lookup**. All IPAM-managed DNS Forward Lookup zones are displayed in the display pane search results. Right-click the zone where you want to add a resource record, and then click **Add DNS resource record**.



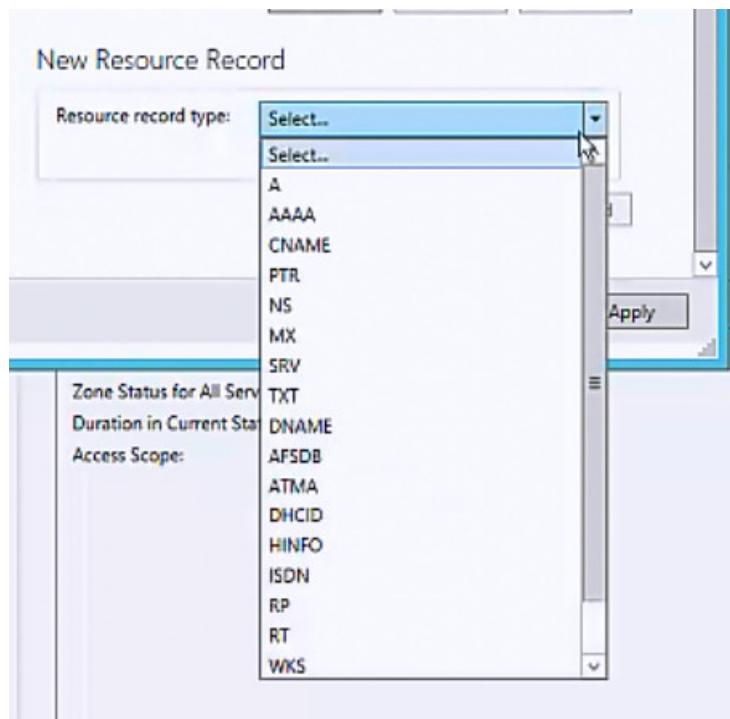
4. The **Add DNS Resource Records** dialog box opens. In **Resource record properties**, click **DNS server** and select the DNS server where you want to add one or more new resource records. In **Configure DNS resource records**, click **New**.



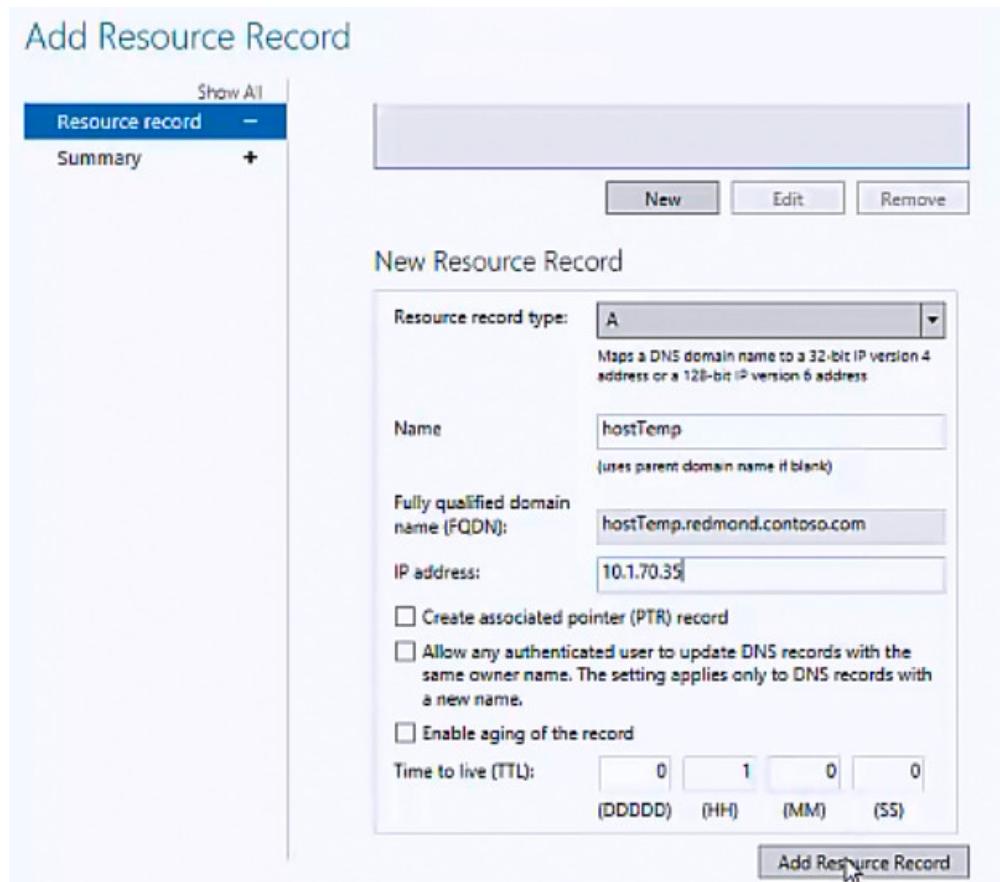
5. The dialog box expands to reveal **New Resource Record**. Click **Resource record type**.



6. The list of resource record types is displayed. Click the resource record type that you want to add.



7. In **New Resource Record**, in **Name**, type a resource record name. In **IP Address**, type an IP address, and then select the resource record properties that are appropriate for your deployment. Click **Add Resource Record**.



8. If you do not want to create additional new resource records, click **OK**. If you want to create additional new resource records, click **New**.

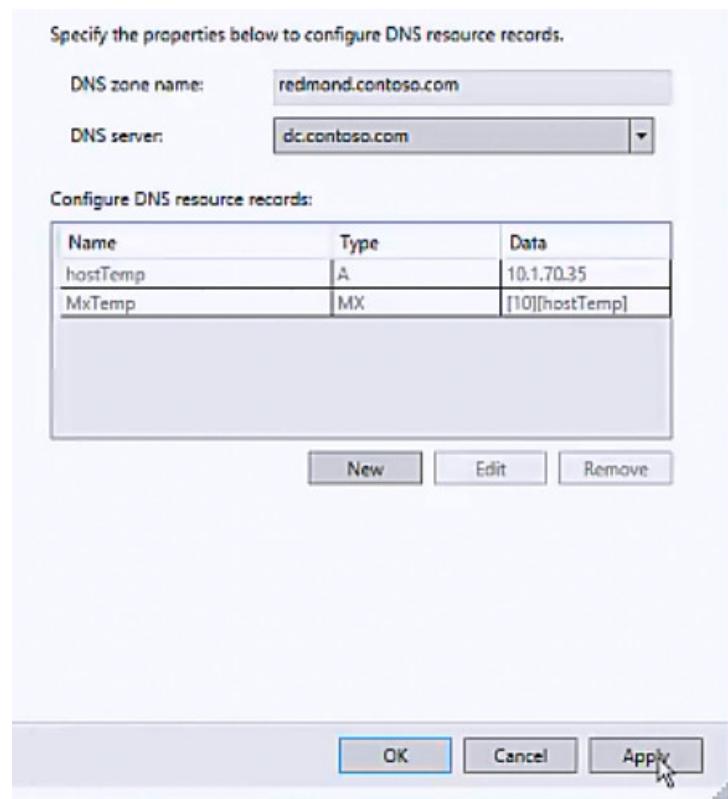
Add Resource Record

The screenshot shows the 'Add Resource Record' dialog box. On the left, there's a navigation pane with 'Show All' at the top, followed by a dropdown menu set to 'Resource record' with a minus sign, and a '+' sign next to 'Summary'. In the main area, the title 'Resource record properties' is displayed above a note: 'Specify the properties below to configure DNS resource records.' Below this are two input fields: 'DNS zone name:' containing 'redmond.contoso.com' and 'DNS server:' containing 'dc.contoso.com' with a dropdown arrow. A table titled 'Configure DNS resource records:' lists one entry: 'hostTemp' under 'Name', 'A' under 'Type', and '10.1.70.35' under 'Data'. At the bottom of this section are three buttons: 'New' (highlighted with a cursor), 'Edit', and 'Remove'.

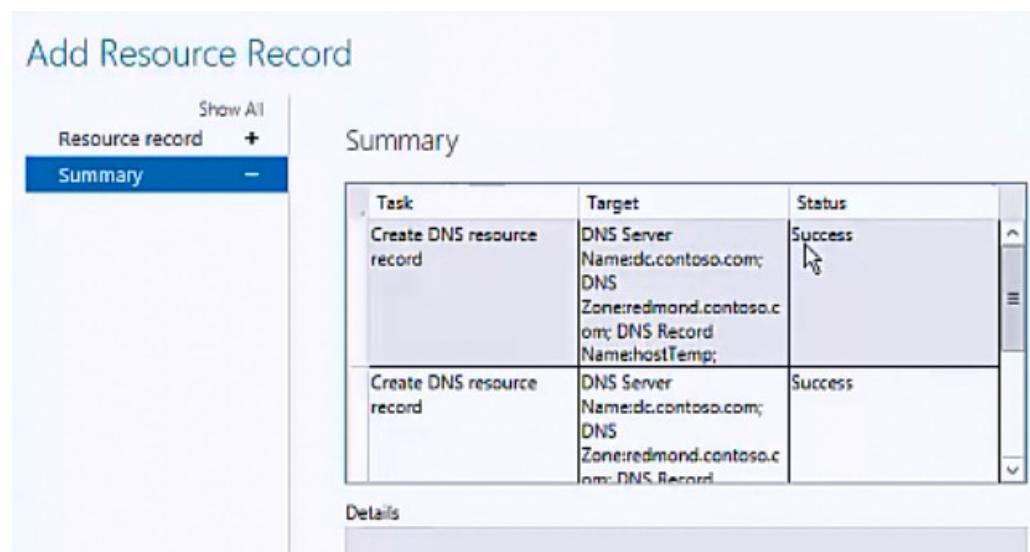
9. The dialog box expands to reveal **New Resource Record**. Click **Resource record type**. The list of resource record types is displayed. Click the resource record type that you want to add.
10. In **New Resource Record**, in **Name**, type a resource record name. In **IP Address**, type an IP address, and then select the resource record properties that are appropriate for your deployment. Click **Add Resource Record**.

The screenshot shows the 'New Resource Record' dialog box for an MX record. The 'Resource record type:' dropdown is set to 'MX', which is highlighted with a cursor. A tooltip explains: 'Provides message routing to a specified mail exchange host that is acting as a mail exchanger for a specified DNS domain name'. The 'Host or child domain:' field contains 'MxTemp', 'Fully qualified domain name (FQDN):' contains 'MxTemp.redmond.contoso.com', 'FQDN of mail server:' contains 'hostTemp', and 'Mail server priority:' contains '10'. There are two unchecked checkboxes: 'Allow any authenticated user to update DNS records with the same owner name. The setting applies only to DNS records with a new name.' and 'Enable aging of the record'. Below these is a 'Time to live (TTL)' field with four input boxes showing '0', '(HH)', '0', and '(SS)'. At the bottom are 'OK', 'Cancel', and 'Apply' buttons, with 'Add Resource Record' also being highlighted with a cursor.

11. If you want to add more resource records, repeat the process for creating records. When you are done creating new resource records, click **Apply**.



12. The **Add Resource Record** dialog box displays a resource records summary while IPAM creates the resource records on the DNS server that you specified. When the records are successfully created, the **Status** of the record is **Success**.



13. Click **OK**.

See Also

[DNS Resource Record Management](#)
[Manage IPAM](#)

Delete DNS Resource Records

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to delete one or more DNS resource records by using the IPAM client console.

Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To delete DNS resource records

1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, in **MONITOR AND MANAGE**, click **DNS Zones**. The navigation pane divides into an upper navigation pane and a lower navigation pane.
3. Click to expand **Forward Lookup** and the domain where the zone and resource records that you want to delete are located. Click on the zone, and in the display pane, click **Current view**. Click **Resource Records**.
4. In the display pane, locate and select the resource records that you want to delete.

Record Name	Record Type	IP Address
abc17	A	10.1.70.17
abc18	A	10.1.70.18
abc19	A	10.1.70.19
abc2	A	10.1.70.2
abc20	A	10.1.70.20
abc3	A	10.1.70.3
abc4	A	10.1.70.4
abc5	A	10.1.70.5
abc6	A	10.1.70.6
abc7	A	10.1.70.7
abc8	A	10.1.70.8
abc9	A	10.1.70.9
hostTemp	A	10.1.70.36
MxTemp	MX	

5. Right-click the selected records, and then click **Delete DNS resource record**.

The screenshot shows the IPAM console interface. On the left, there's a navigation tree with sections like 'Forward Lookup' and 'IPv4 Reverse Lookup'. Under 'Forward Lookup', several DNS zones are listed: contoso.com, dublin, hyderabad, london, newyork, redmond, singapore, and sydney. The 'redmond' zone is currently selected. On the right, a table lists static IP addresses: abc6 (A, 10.1.70.6, Static), abc7 (A, 10.1.70.7, Static), abc8 (A, 10.1.70.8, Static), and abc9 (A, 10.1.70.9, Static). A context menu is open over the first four rows of the table, with 'Delete DNS resource record...' highlighted.

6. The **Delete DNS Resource Record** dialog box opens. Verify that the correct DNS server is selected. If it is not, click **DNS server** and select the server from which you want to delete the resource records. Click **OK**. IPAM deletes the resource records from the DNS server.



See Also

[DNS Resource Record Management](#)

[Manage IPAM](#)

Filter the View of DNS Resource Records

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to filter the view of DNS resource records in the IPAM client console.

Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To filter the view of DNS resource records

1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, in **MONITOR AND MANAGE**, click **DNS Zones**. The navigation pane divides into an upper navigation pane and a lower navigation pane.
3. In the lower navigation pane, click **Forward Lookup**. All IPAM-managed DNS Forward Lookup zones are displayed in the display pane search results.
4. Click on the zone whose records you want to view and filter.
5. In the display pane, click **Current view**, and then click **Resource Records**. The resource records for the zone are shown in the display pane.
6. In the display pane, click **Add criteria**.

The screenshot shows the IPAM client console interface. The left navigation pane is collapsed. The main area displays the 'dublin' zone with 45 total records. A 'Filter' dialog is open, with 'Resource Records' selected. The 'Add criteria' button is highlighted. Below it is a list of filter options: Record Name, Record Type, IP Address, Timestamp, Time to live, Access Scope, Mapped to IP Address, and Data. The 'Record Type' column in the display pane shows entries for SOA, NS, SRV, SRV, DHCID, and HINFO. The 'IP' column shows all entries as 10.0.0.10. The display pane also lists records abc1 through abc15.

Record Type	IP
SOA	10.0.0.10
NS	10.0.0.10
SRV	10.0.0.10
SRV	10.0.0.10
DHCID	10.0.0.10
HINFO	10.0.0.10
A	10.0.0.10

7. Select a criteria from the drop-down list. For example, if you want to view a specific record type, click **Record Type**.

The screenshot shows the Microsoft DNS Manager interface. On the left, a navigation pane includes options like OVERVIEW, SERVER INVENTORY, IP ADDRESS SPACE, VIRTUALIZED IP ADDRESS SPA..., MONITOR AND MANAGE, and DNS Zones. Under MONITOR AND MANAGE, DNS and DHCP Servers, DHCP Scopes, and DNS Zones are listed, with DNS Zones selected. Below this, Server Groups are shown. On the right, the 'dublin' zone is selected, displaying 45 total records. A context menu is open over a table of resource records, with the 'Add criteria' option selected. The 'Record Type' checkbox is checked, and the 'A' button is highlighted.

Record Name	Type	IP
abc1	A	10
abc11	A	10
abc12	A	10
abc13	A	10
abc14	A	10
abc15	A	10

8. Click **Add**.

The screenshot shows the Microsoft DNS Manager interface. The navigation pane and zone selection are identical to the previous screenshot. A context menu is open over a table of resource records, with the 'Add criteria' option selected. The 'Record Type' checkbox is checked, and the 'SRV' button is highlighted.

Record Name	Type	IP
abc1	A	10
abc11	A	10
abc12	A	10
abc13	A	10
abc14	A	10
abc15	A	10

9. **Record Type** is added as a search parameter. Enter text for the type of record that you want to find. For example, if you want to view only SRV records, type **SRV**.

Record Name	Record Type	IP
@	SOA	
@	NS	
_finger._tcp	SRV	
_ftp._tcp	SRV	
abc1	DHCID	
abc1	HINFO	
abc1	A	10
abc11	A	10
abc12	A	10
abc13	A	10
abc14	A	10

10. Press ENTER. The DNS resource records are filtered according to the criteria and search phrase that you specified.

Record Name	Record Type	IP
_finger._tcp	SRV	
_ftp._tcp	SRV	

See Also

[DNS Resource Record Management](#)

[Manage IPAM](#)

View DNS Resource Records for a Specific IP Address

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to view the DNS resource records that are associated with the IP Address that you choose.

Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To view resource records for an IP Address

1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, in **IP ADDRESS SPACE**, click **IP Address Inventory**. In the lower navigation pane, click either **IPv4** or **IPv6**. The IP address inventory appears in the display pane search view. Locate and select the IP address whose DNS resource records you want to view.

The screenshot shows the IPAM client console interface. On the left, the navigation pane is open with the following sections:

- OVERVIEW
- SERVER INVENTORY
- IP ADDRESS SPACE
 - IP Address Blocks
 - IP Address Inventory** (selected)
 - IP Address Range Groups
- VIRTUALIZED IP ADDRESS SPA...
- MONITOR AND MANAGE
 - DNS and DHCP Servers
 - DHCP Scopes
 - DNS Zones
 - Server Groups

Below the navigation pane, there are two buttons: **IPv4** (selected) and **IPv6**.

The main display pane is titled "IPv4" and shows the "IPv4 | 148 total" IP address inventory. It includes a "Filter" bar and a table with columns: Duplicate, Expiry Status, IP Address, MAC Address, and Managed. The table lists 148 entries, all of which are "Not expired". The first few entries are:

No	Expiry Status	IP Address	Managed
No	Not expired	10.1.10.1	IPAM
No	Not expired	10.1.10.2	IPAM
No	Not expired	10.1.10.3	IPAM
No	Not expired	10.1.10.4	IPAM
No	Not expired	10.1.10.5	IPAM
No	Not expired	10.1.10.6	IPAM
No	Not expired	10.1.10.7	IPAM
No	Not expired	10.1.10.8	IPAM
No	Not expired	10.1.10.9	IPAM
No	Not expired	10.1.10.10	IPAM
No	Not expired	10.1.10.11	IPAM
No	Not expired	10.1.10.12	IPAM
No	Not expired	10.1.10.13	IPAM

3. In the display pane **Details View**, click **DNS resource records**. The resource records that are associated with the selected IP address are displayed.

Duplicate	Expiry Status	IP Address	MAC Address	Managed by Service	Service Instance
No	● Not expired	10.1.10.1		IPAM	localhost
No	● Not expired	10.1.10.2		IPAM	localhost
No	● Not expired	10.1.10.3		IPAM	localhost
No	● Not expired	10.1.10.4		IPAM	localhost
No	● Not expired	10.1.10.5		IPAM	localhost
No	● Not expired	10.1.10.6		IPAM	localhost
No	● Not expired	10.1.10.7		IPAM	localhost
No	● Not expired	10.1.10.8		IPAM	localhost
No	● Not expired	10.1.10.9		IPAM	localhost
No	● Not expired	10.1.10.10		IPAM	localhost
No	● Not expired	10.1.10.11		IPAM	localhost
No	● Not expired	10.1.10.12		IPAM	localhost
No	● Not expired	10.1.10.13		IPAM	localhost
No	● Not expired	10.1.10.14		IPAM	localhost
No	● Not expired	10.1.10.15		IPAM	localhost
No	● Not expired	10.1.10.16		IPAM	localhost

Details View

10.1.10.1

Configuration Details		DNS resource records		Event Catalog
Name	Type	Zone name	Data	Mapped
abc1	A	dublin.contoso.com	10.1.10.1	True
1	PTR	10.1.10.in-addr.arpa	abc1.dublin.contoso.com.	True
_ftp._tcp	SRV	dublin.contoso.com	[0][0][21]abc1.dublin.contoso.com.]	False
mxrecord1	MX	dublin.contoso.com	[1][abc1.dublin.contoso.com.]	False
rtrecord1	RT	dublin.contoso.com	[0][abc1.dublin.contoso.com.]	False
_ftp._tcp	SRV	dublin.contoso.com	[0][0][21]abc1.dublin.contoso.com.]	False
mxrecord1	MX	dublin.contoso.com	[1][abc1.dublin.contoso.com.]	False
rtrecord1	RT	dublin.contoso.com	[0][abc1.dublin.contoso.com.]	False

See Also

[DNS Resource Record Management](#)

[Manage IPAM](#)

DNS Zone Management

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

This topic provides information about managing DNS zones by using the IPAM client console.

NOTE

In addition to this topic, the following IPAM DNS zone management topics are available in this section.

- [Create a DNS Zone](#)
- [Edit a DNS Zone](#)
- [View DNS Resource Records for a DNS Zone](#)
- [View DNS Zones](#)

When you deploy IPAM in Windows Server 2016, you can use IPAM to manage DNS zones.

In the IPAM console, you can view DNS resource records for a specific DNS zone, and filter the records based on type, IP address, resource record data, and other filtering options. In addition, you can edit DNS resource records for specific zones

See Also

[Manage IPAM](#)

Create a DNS Zone

9/1/2018 • 2 minutes to read • [Edit Online](#)

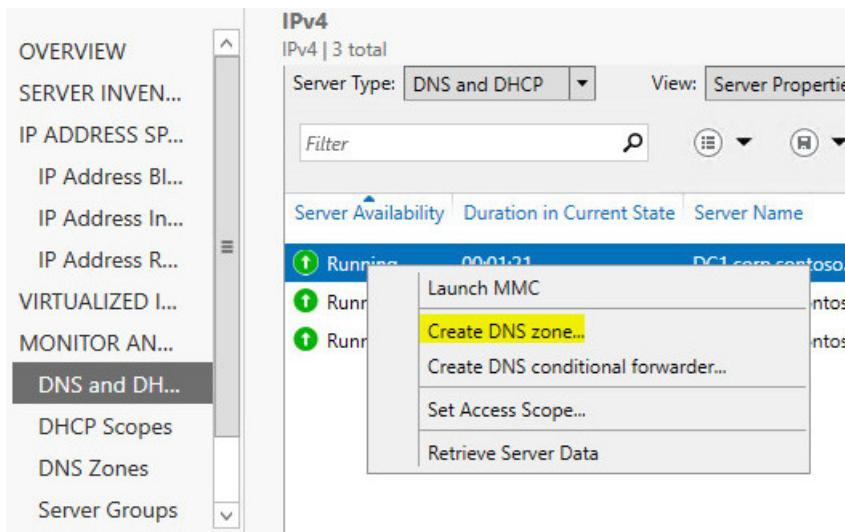
Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016

You can use this topic to create a DNS zone by using the IPAM client console.

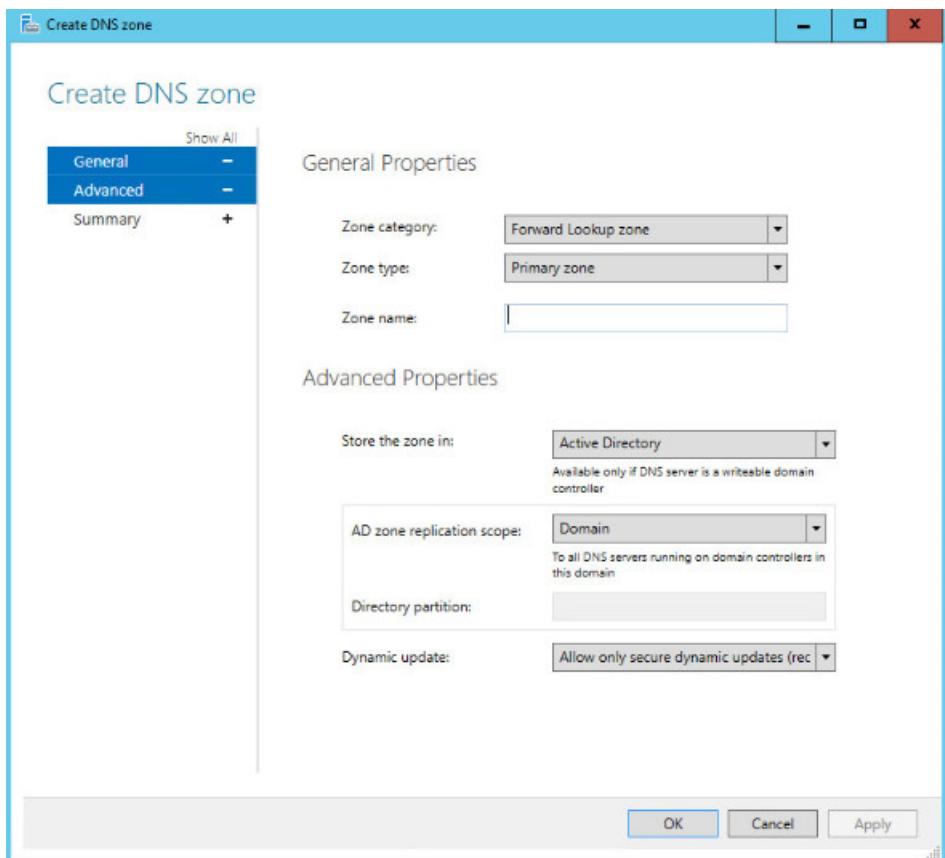
Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To create a DNS zone

1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, in **MONITOR AND MANAGE**, click **DNS and DHCP Servers**. In the display pane, click **Server Type**, and then click **DNS**. All DNS servers that are managed by IPAM are listed in the search results.
3. Locate the server where you want to add a zone, and right-click the server. Click **Create DNS zone**.



4. The **Create DNS Zone** dialog box opens. In **General Properties**, select a zone category, a zone type , and enter a name in **Zone name**. Also select values appropriate for your deployment in **Advanced Properties**, and then click **OK**.



See Also

[DNS Zone Management](#)

[Manage IPAM](#)

Edit a DNS Zone

9/1/2018 • 2 minutes to read • [Edit Online](#)

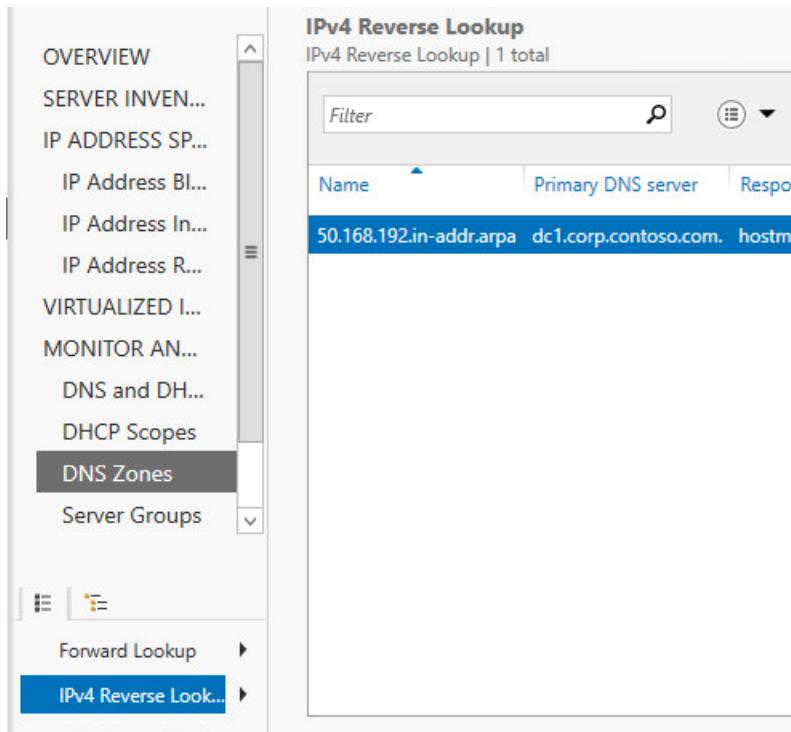
Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016

You can use this topic to edit a DNS zone in the IPAM client console.

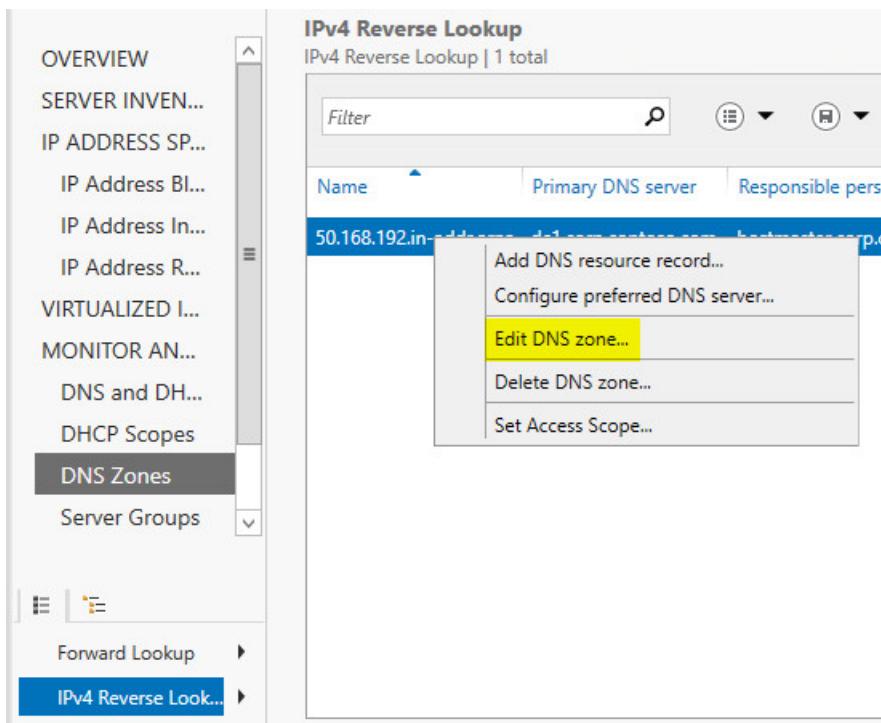
Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To edit a DNS zone

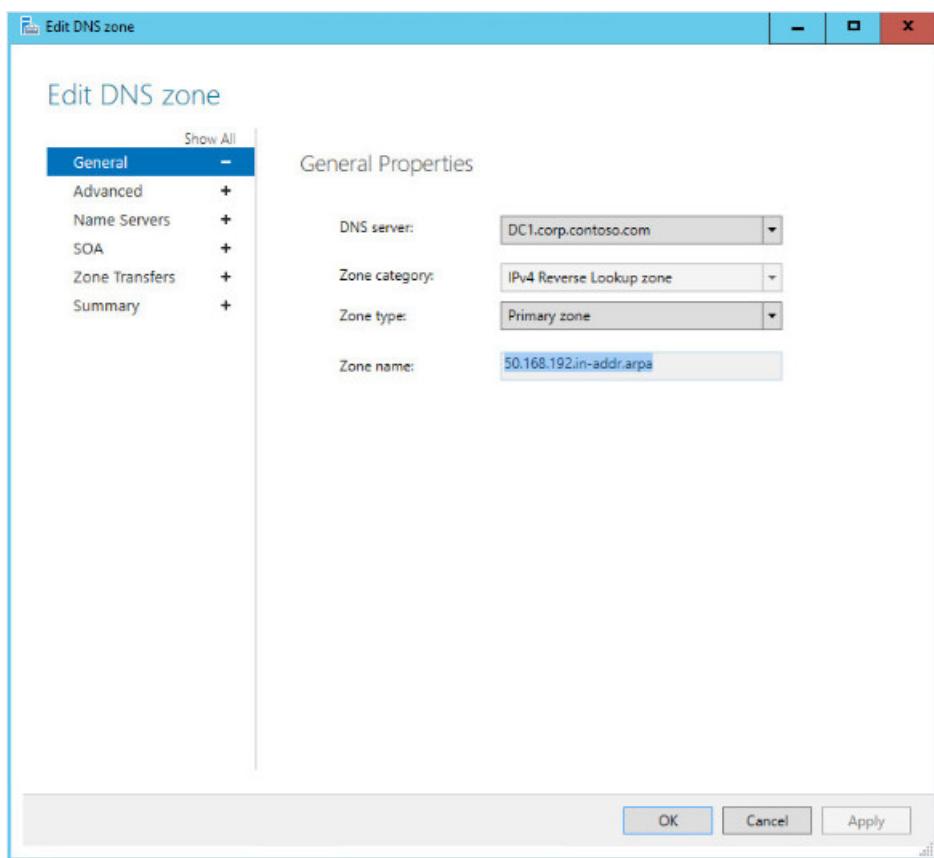
1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, in **MONITOR AND MANAGE**, click **DNS Zones**. The navigation pane divides into an upper navigation pane and a lower navigation pane.
3. In the lower navigation pane, make one of the following selections:
 - Forward Lookup
 - IPv4 Reverse Lookup
 - IPv6 Reverse Lookup
4. For example, select IPv4 Reverse Lookup.



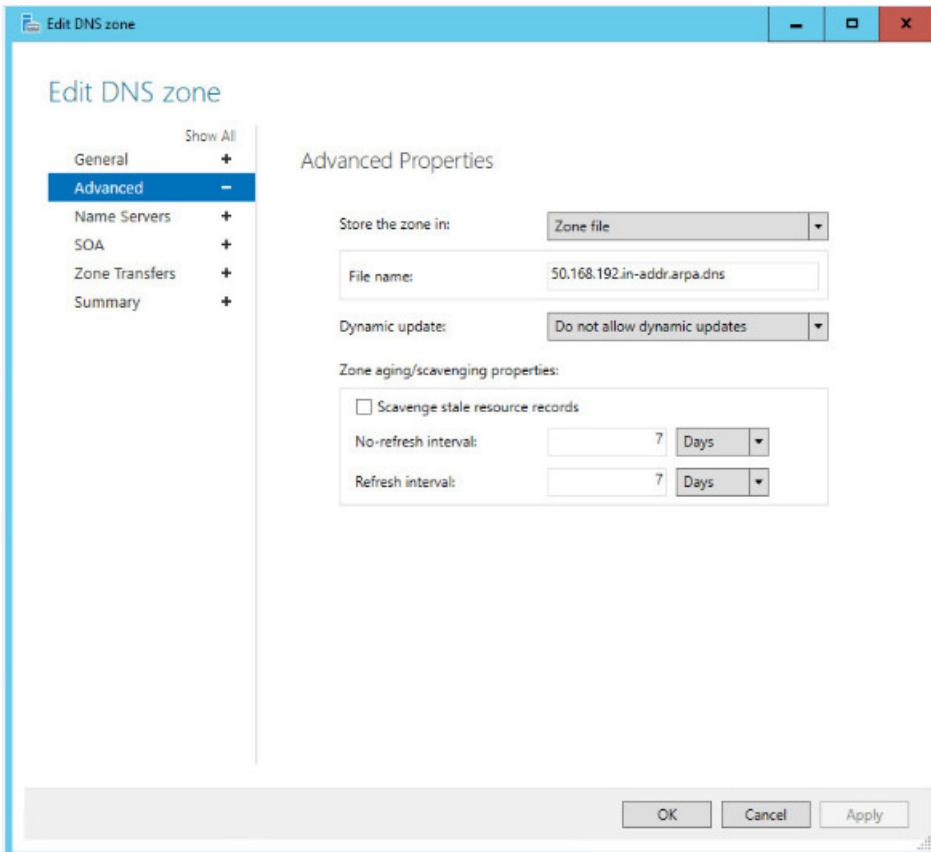
5. In the display pane, right-click the zone that you want to edit, and then click **Edit DNS Zone**.



6. The **Edit DNS Zone** dialog box opens with the **General** page selected. If needed, edit the General zone properties: **DNS server**, **Zone category**, and **Zone type**, and then click **Apply** or, if your edits are complete, **OK**.



7. In the **Edit DNS Zone** dialog box, click **Advanced**. The **Advanced** zone properties page opens. If needed, edit the properties that you want to change, and then click **Apply** or, if your edits are complete, **OK**.



8. If needed, select the additional zone properties page names (Name Servers, SOA, Zone Transfers), make your edits, and click **Apply** or **OK**. To review all of your zone edits, click **Summary**, and then click **OK**.

See Also

[DNS Zone Management](#)

[Manage IPAM](#)

View DNS Resource Records for a DNS Zone

9/1/2018 • 2 minutes to read • [Edit Online](#)

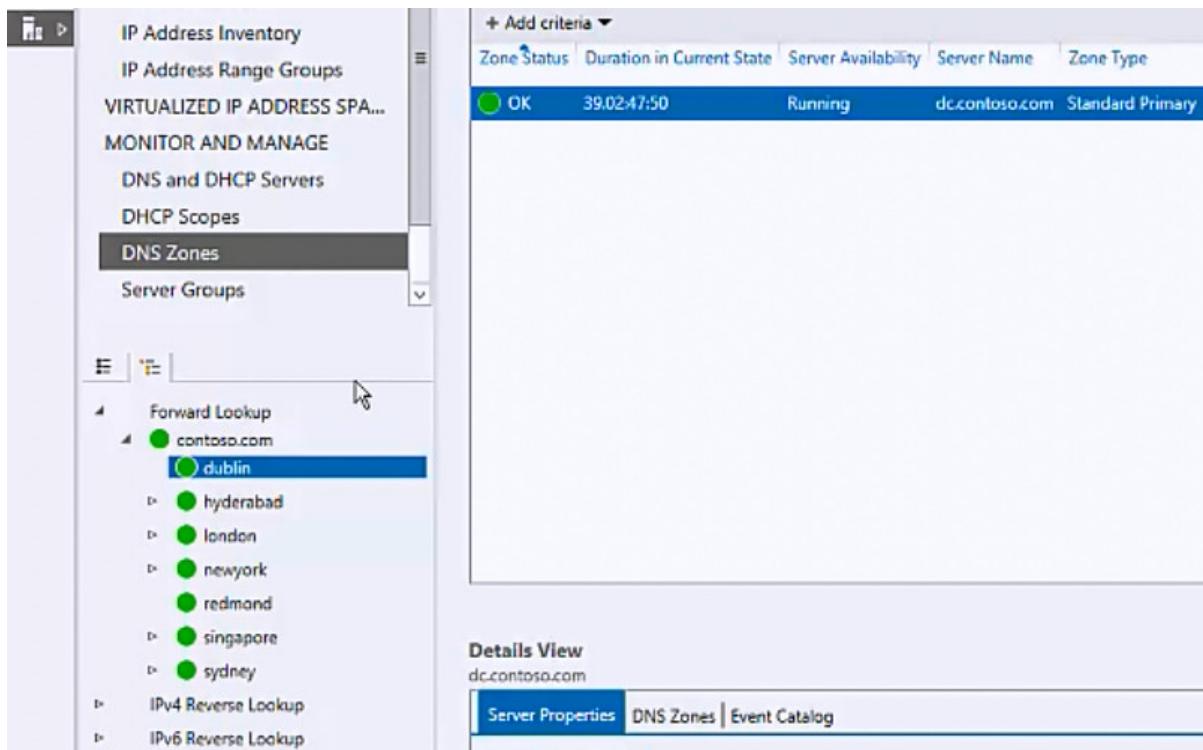
Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016

You can use this topic to view DNS resource records for a DNS zone in the IPAM client console.

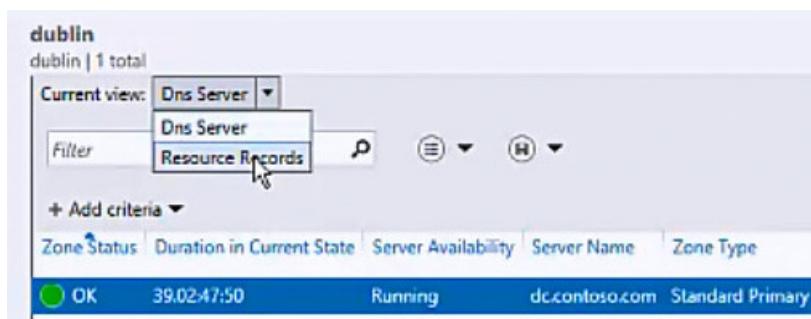
Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To view DNS resource records for a zone

1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, in **MONITOR AND MANAGE**, click **DNS Zones**. The navigation pane divides into an upper navigation pane and a lower navigation pane.
3. In the lower navigation pane, click **Forward Lookup**, and then expand the domain and zone list to locate and select the zone you want to view. For example, if you have a zone named **dublin**, click **dublin**.



4. In the display pane, the default view is of the DNS servers for the zone. To change the view, click **Current view**, and then click **Resource Records**.



5. The DNS resource records for the zone are displayed. To filter the records, type the text you want to find in

Filter.

The screenshot shows a table titled "Resource Records" with the following columns: Record Name, Record Type, IP Address, Timestamp, Time to live, and Access Scope. The table contains 17 rows of data, including an SOA record, several NS, SRV, and DHCID records, and multiple A records pointing to static IP addresses (10.1.10.1 through 10.1.10.17). The "Access Scope" column consistently shows "\Global\cont".

Record Name	Record Type	IP Address	Timestamp	Time to live	Access Scope
@	SOA		Static	01:00:00	\Global\cont
@	NS		Static	01:00:00	\Global\cont
_finger_tcp	SRV		Static	01:00:00	\Global\cont
_ftp_tcp	SRV		Static	01:00:00	\Global\cont
abc1	DHCID		Static	01:00:00	\Global\cont
abc1	HINFO		Static	01:00:00	\Global\cont
abc1	A	10.1.10.1	Static	01:00:00	\Global\cont
abc11	A	10.1.10.11	Static	23.03:33:20	\Global\cont
abc12	A	10.1.10.12	Static	23.03:33:20	\Global\cont
abc13	A	10.1.10.13	Static	23.03:33:20	\Global\cont
abc14	A	10.1.10.14	Static	23.03:33:20	\Global\cont
abc15	A	10.1.10.15	Static	23.03:33:20	\Global\cont
abc16	A	10.1.10.16	Static	23.03:33:20	\Global\cont
abc17	A	10.1.10.17	Static	360.00:00:00	\Global\cont

6. To filter the resource records by record type, access scope, or other criteria, click **Add criteria**, and then make selections from the criteria list and click **Add**.

The screenshot shows the same table as above, but with a modal dialog box titled "+ Add criteria" overlaid. The dialog lists various filtering options with checkboxes: Record Name (which is checked), Record Type, IP Address, Timestamp, Time to live, Access Scope, Mapped to IP Address, and Data. At the bottom of the dialog are "Add" and "Cancel" buttons. The "Record Name" checkbox is currently selected.

Record Type	IP Address	Timestamp	Time to live
SOA		Static	01:00:00
NS		Static	01:00:00
SRV		Static	01:00:00
SRV		Static	01:00:00
DHCID		Static	01:00:00
HINFO		Static	01:00:00

abc1 A 10.1.10.1 Static 01:00:00

See Also

[DNS Zone Management](#)

[Manage IPAM](#)

View DNS Zones

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to view DNS zones in the IPAM client console.

Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To view DNS zones in the IPAM client console

1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, in **MONITOR AND MANAGE**, click **DNS Zones**. The navigation pane divides into an upper navigation pane and a lower navigation pane.
3. In the lower navigation pane, make one of the following selections:
 - Forward Lookup
 - IPv4 Reverse Lookup
 - IPv6 Reverse Lookup
 - Conditional Forwarder

See Also

[DNS Zone Management](#)

[Manage IPAM](#)

Manage Resources in Multiple Active Directory Forests

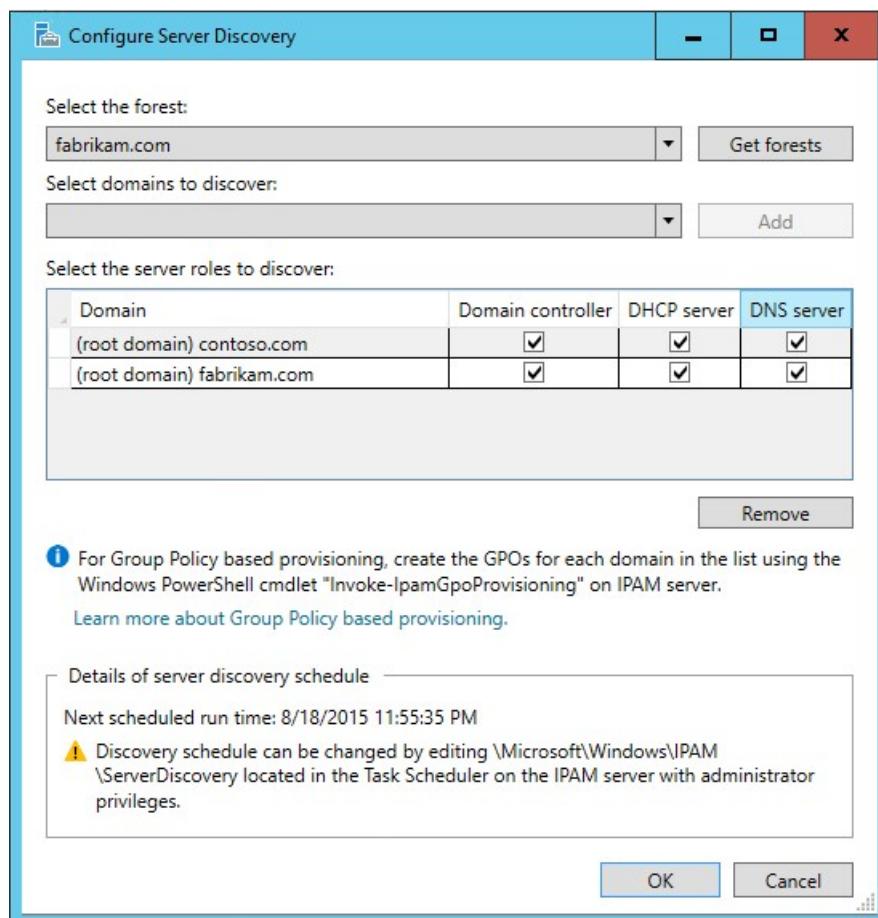
9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to use IPAM to manage domain controllers, DHCP servers, and DNS servers in multiple Active Directory forests.

To use IPAM to manage resources in remote Active Directory forests, each forest that you want to manage must have a two way trust with the forest where IPAM is installed.

To start the discovery process for different Active Directory forests, open Server Manager and click IPAM. In the IPAM client console, click **Configure Server Discovery**, and then click **Get forests**. This initiates a background task that discovers trusted forests and their domains. After the discovery process completes, click **Configure Server Discovery**, which opens the following dialog box.



NOTE

For Group Policy-based provisioning for an Active Directory Cross Forest scenario, ensure that you run the following Windows PowerShell cmdlet on the IPAM server and not on the trusting domain DCs. As an example, if your IPAM server is joined to the forest corp.contoso.com and the trusting forest is fabrikam.com, you can run the following Windows PowerShell cmdlet on the IPAM server in corp.contoso.com for Group Policy-based provisioning on the fabrikam.com forest. To run this cmdlet, you must be a member of the Domain Admins group in the fabrikam.com forest.

```
Invoke-IpamGpoProvisioning -Domain fabrikam.COM -GpoPrefixName IPAMSERVER -IpamServerFqdn IPAM.CORP.CONTOSO.COM
```

In the **Configure Server Discovery** dialog box, click **Select the forest**, and then choose the forest that you want to manage with IPAM. Also select the domains that you want to manage, and then click **Add**.

In **Select the server roles to discover**, for each domain that you want to manage, specify the type of servers to discover. The options are **Domain controller**, **DHCP server**, and **DNS server**.

By default, domain controllers, DHCP servers, and DNS servers are discovered - so if you do not want to discover one of these types of servers, ensure that you deselect the checkbox for that option.

In the example illustration above, the IPAM server is installed in the contoso.com forest, and the root domain of the fabrikam.com forest is added for IPAM management. The selected server roles allow IPAM to discover and manage domain controllers, DHCP servers, and DNS servers in the fabrikam.com root domain and the contoso.com root domain.

After you have specified forests, domains, and server roles, click **OK**. IPAM performs discovery, and when discovery completes, you can manage resources in both the local and remote forest.

Purge Utilization Data

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to delete utilization data from the IPAM database.

You must be a member of **IPAM Administrators**, the local computer **Administrators** group, or equivalent, to perform this procedure.

To purge the IPAM database

1. Open Server Manager, and then browse to the IPAM client interface.
2. Browse to one of the following locations: **IP Address Blocks**, **IP Address Inventory**, or **IP Address Range Groups**.
3. Click **TASKS**, and then click **Purge Utilization Data**. The **Purge Utilization Data** dialog box opens.
4. In **Purge all utilization data on or before**, click **Select a date**.
5. Choose the date for which you want to delete all database records both on and before that date.
6. Click **OK**. IPAM deletes all the records that you have specified.

Role-based Access Control

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

This topic provides information about using role-based access control in IPAM.

NOTE

In addition to this topic, the following IPAM access control documentation is available in this section.

- [Manage Role Based Access Control with Server Manager](#)
- [Manage Role Based Access Control with Windows PowerShell](#)

Role-based access control allows you to specify access privileges at various levels, including the DNS server, DNS zone, and DNS resource record levels.

By using role based access control, you can specify who has granular control over operations to create, edit, and delete different types of DNS resource records.

You can configure access control so that users are restricted to the following permissions.

- Users can edit only specific DNS resource records
- Users can edit DNS resource records of a specific type, such as PTR or MX
- Users can edit DNS resource records for specific zones

See Also

[Manage Role Based Access Control with Server Manager](#)

[Manage Role Based Access Control with Windows PowerShell](#)

[Manage IPAM](#)

Manage Role Based Access Control with Server Manager

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the following topics to manage role based access control by using Server Manager, which has a graphical user interface.

- [Create a User Role for Access Control](#)
- [Create an Access Policy](#)
- [Set Access Scope for a DNS Zone](#)
- [Set Access Scope for DNS Resource Records](#)
- [View Roles and Role Permissions](#)

Alternately, you can use Windows PowerShell to manage IPAM role based access control. For more information, see [Manage Role Based Access Control with Windows PowerShell](#).

See Also

[Manage IPAM](#)

Create a User Role for Access Control

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to create a new Access Control user role in the IPAM client console.

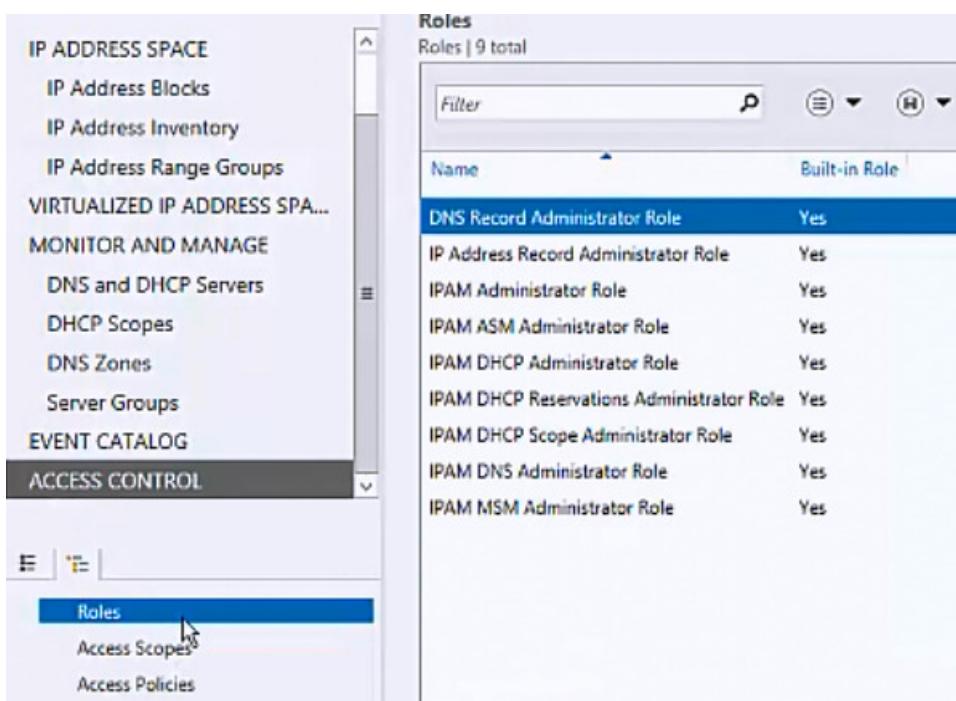
Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

NOTE

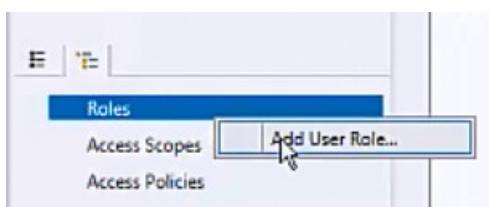
After you create a role, you can create an access policy to assign the role to a specific user or Active Directory group. For more information, see [Create an Access Policy](#).

To create a role

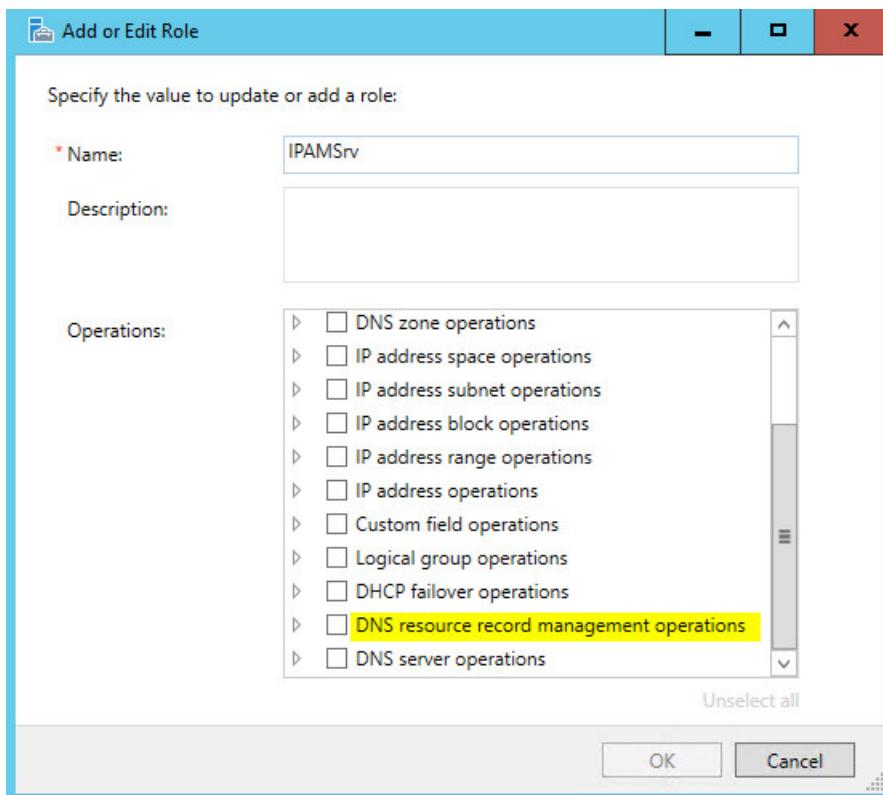
1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, click **ACCESS CONTROL**, and in the lower navigation pane, click **Roles**.



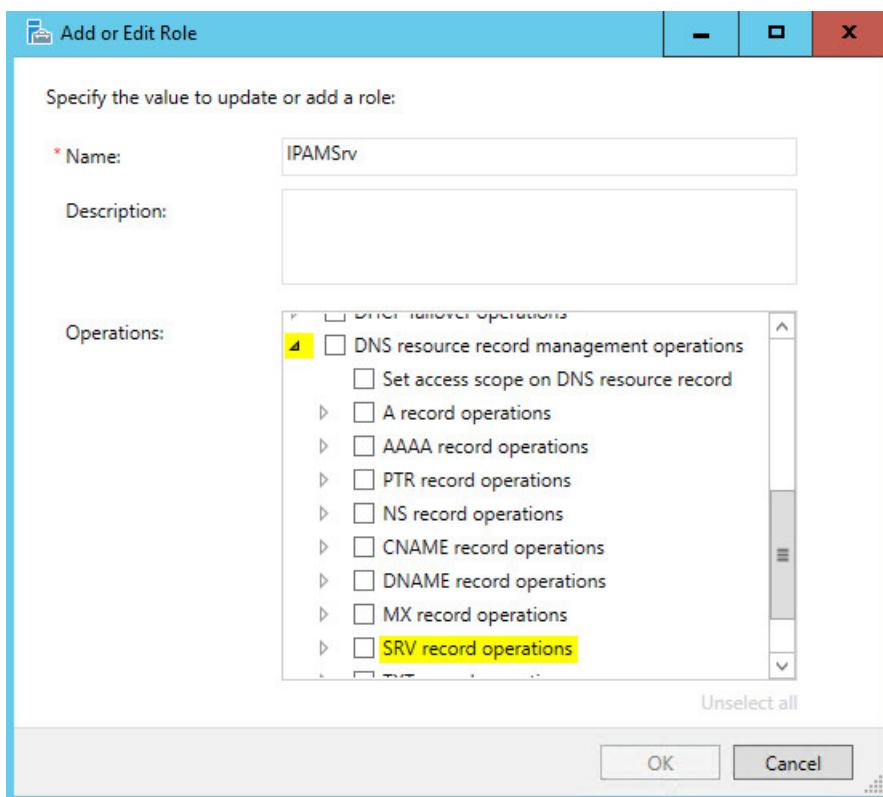
3. Right-click **Roles**, and then click **Add User Role**.



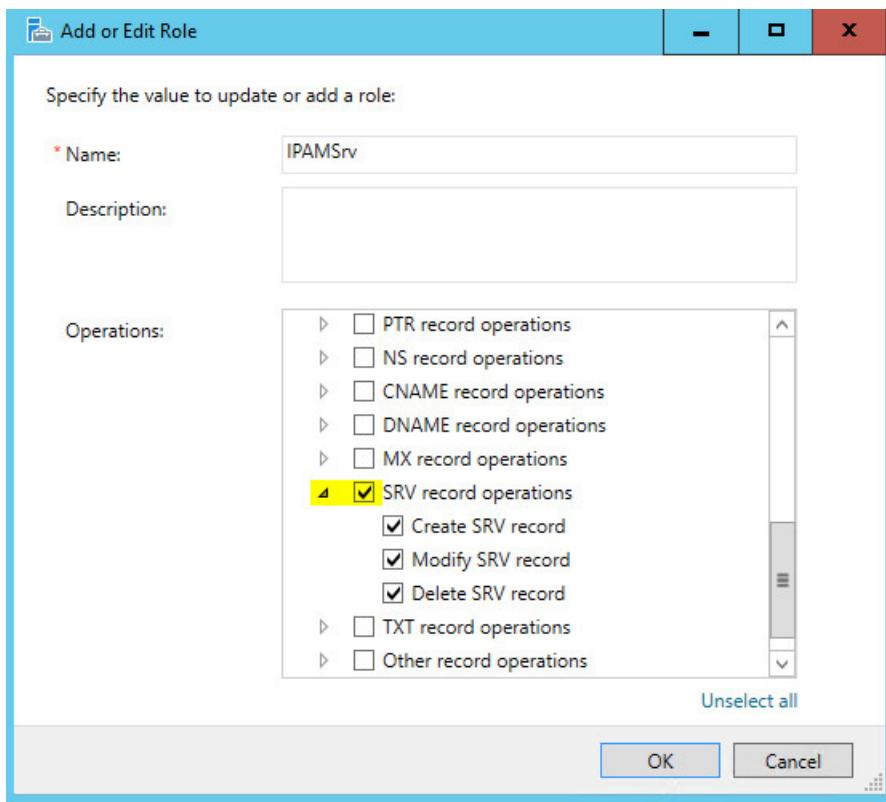
4. The **Add or Edit Role** dialog box opens. In **Name**, type a name for the role that makes the role function clear. For example, if you want to create a role that allows Administrators to manage DNS SRV resource records, you might name the role **IPAMSrv**. If needed, scroll down in **Operations** to locate the type of operations you want to define for the role. For this example, scroll down to **DNS resource record management operations**.



5. Expand **DNS resource record management operations**, and then locate **SRV record operations**.



6. Expand and select **SRV record operations**, and then click **OK**.



7. In the IPAM client console, click the role that you just created. In **Details View**, the permitted operations for the role are displayed.

ACCESS CONT...

IPAM DNS Administrator Role Yes

IPAM MSM Administrator Role Yes

IPAMSRV No

Details View

IPAMSRV

Details Event Catalog

Name: IPAMSRV

Built-in Role: No

Description:

Operations:

- Create SRV record
- Modify SRV record
- Delete SRV record

See Also

[Role-based Access Control](#)
[Manage IPAM](#)

Create an Access Policy

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to create an access policy in the IPAM client console.

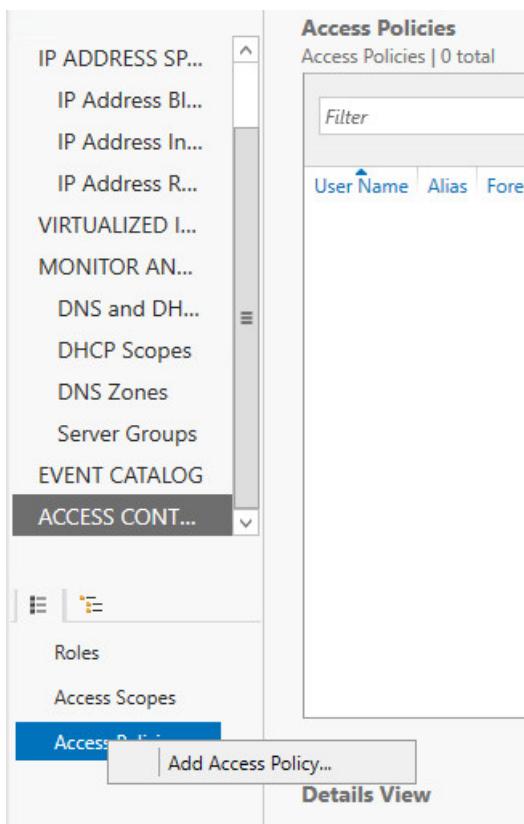
Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

NOTE

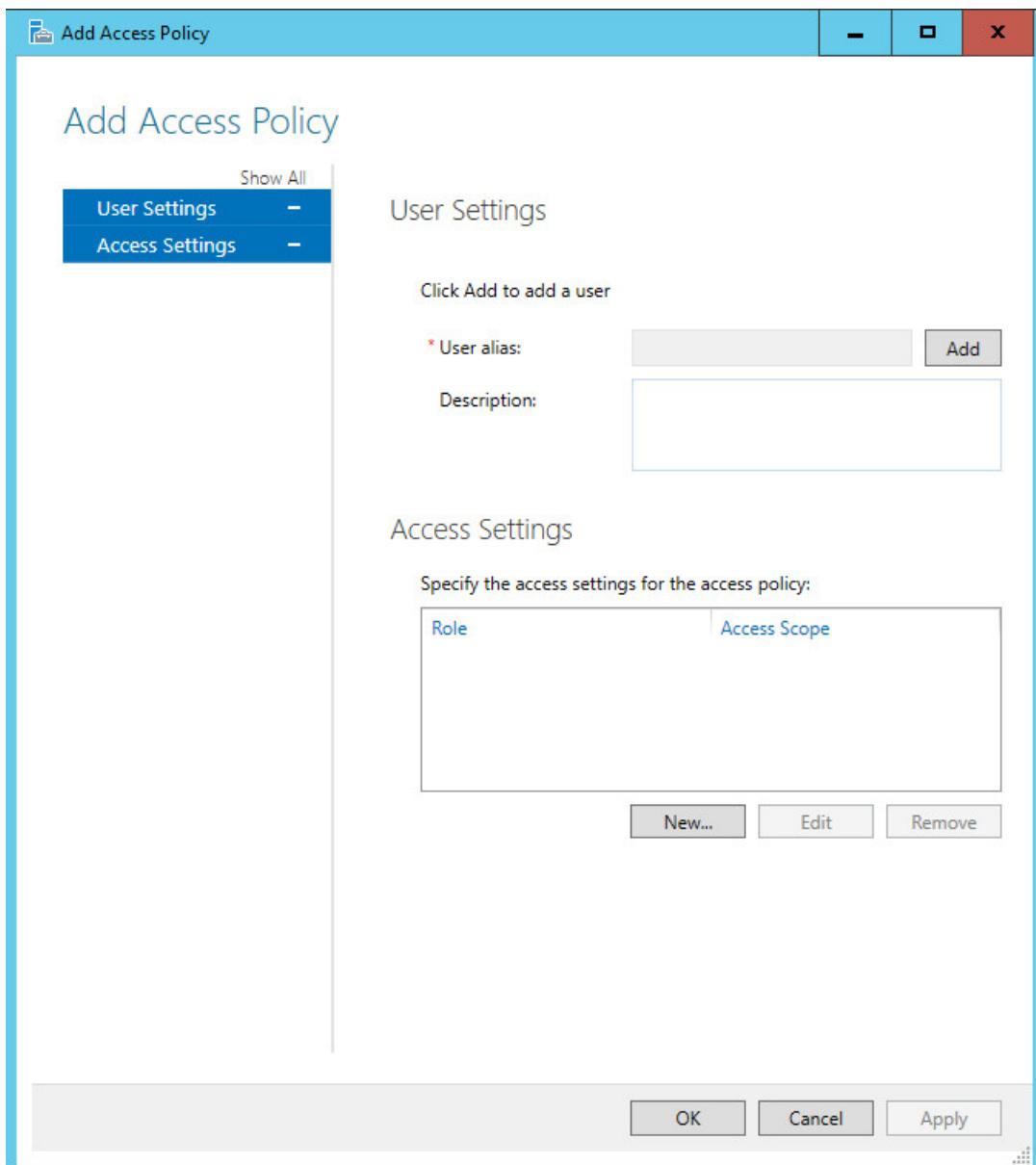
You can create an access policy for a specific user or for a user group in Active Directory. When you create an access policy, you must select either a built-in IPAM role or a custom role that you have created. For more information on custom roles, see [Create a User Role for Access Control](#).

To create an access policy

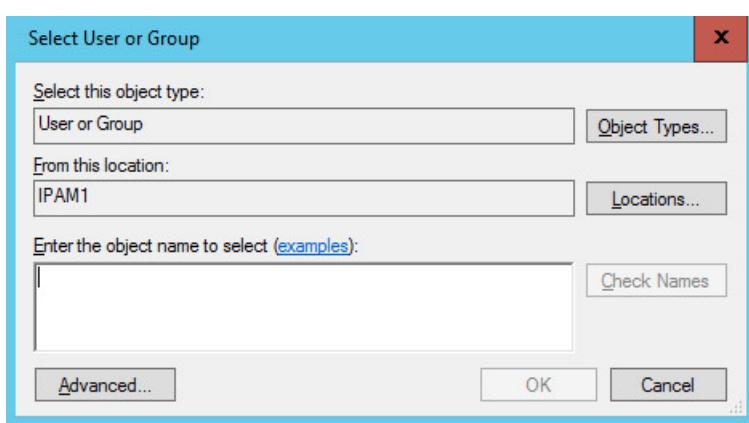
1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, click **ACCESS CONTROL**. In the lower navigation pane, right-click **Access Policies**, and then click **Add Access Policy**.



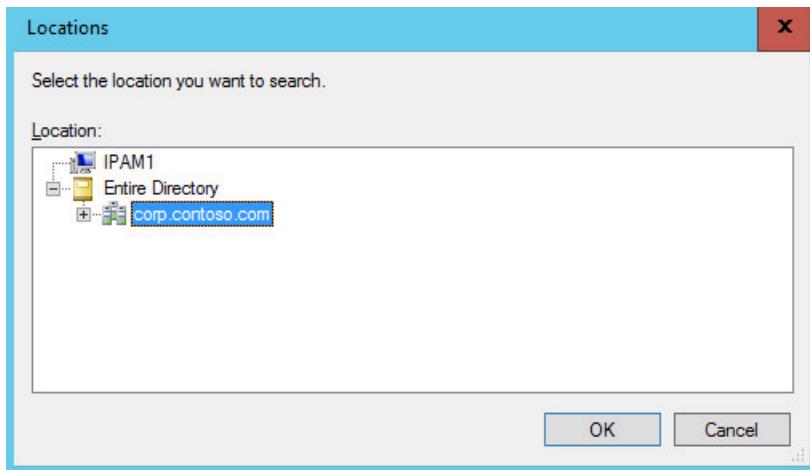
3. The **Add Access Policy** dialog box opens. In **User Settings**, click **Add**.



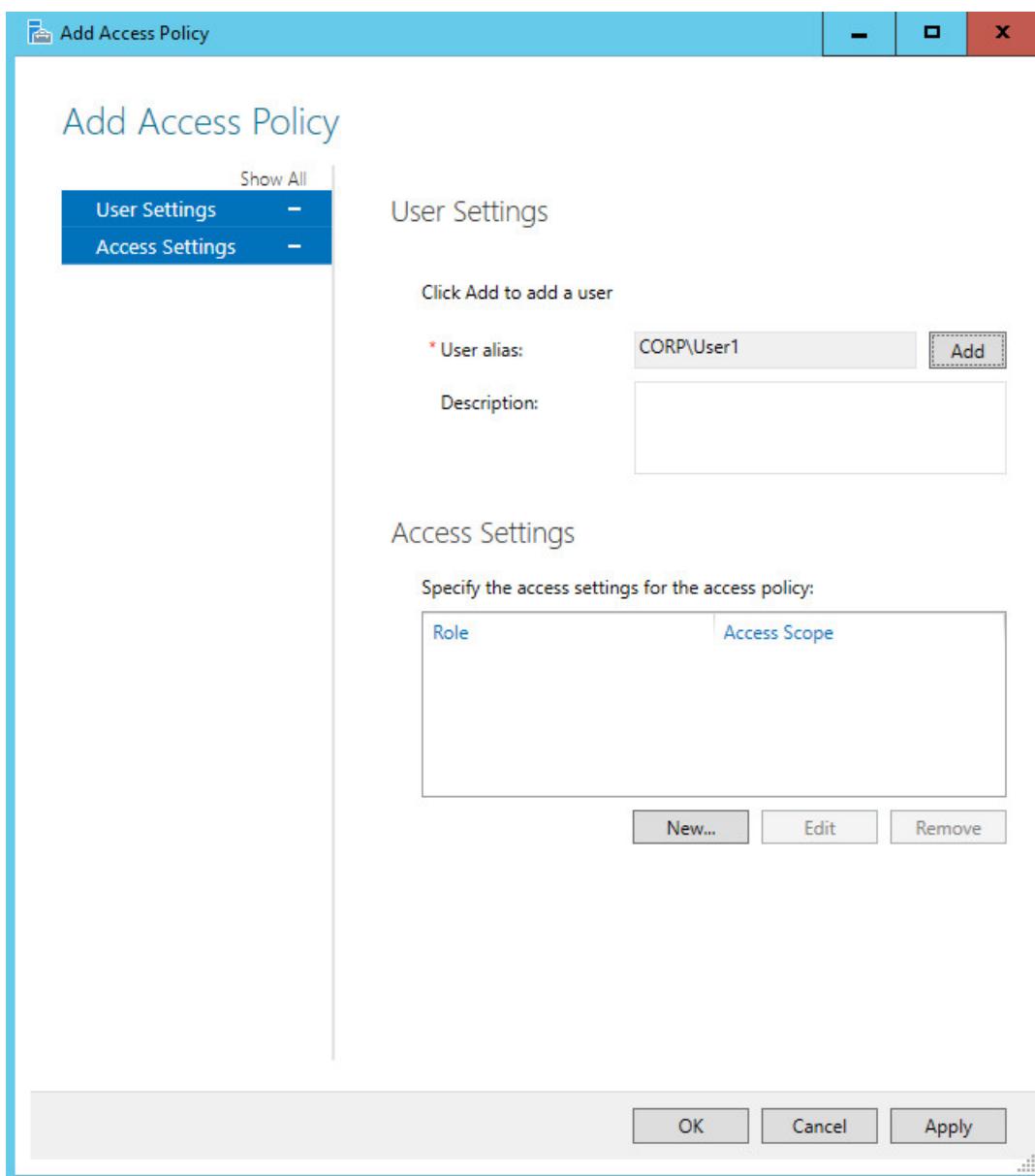
4. The **Select User or Group** dialog box opens. Click **Locations**.



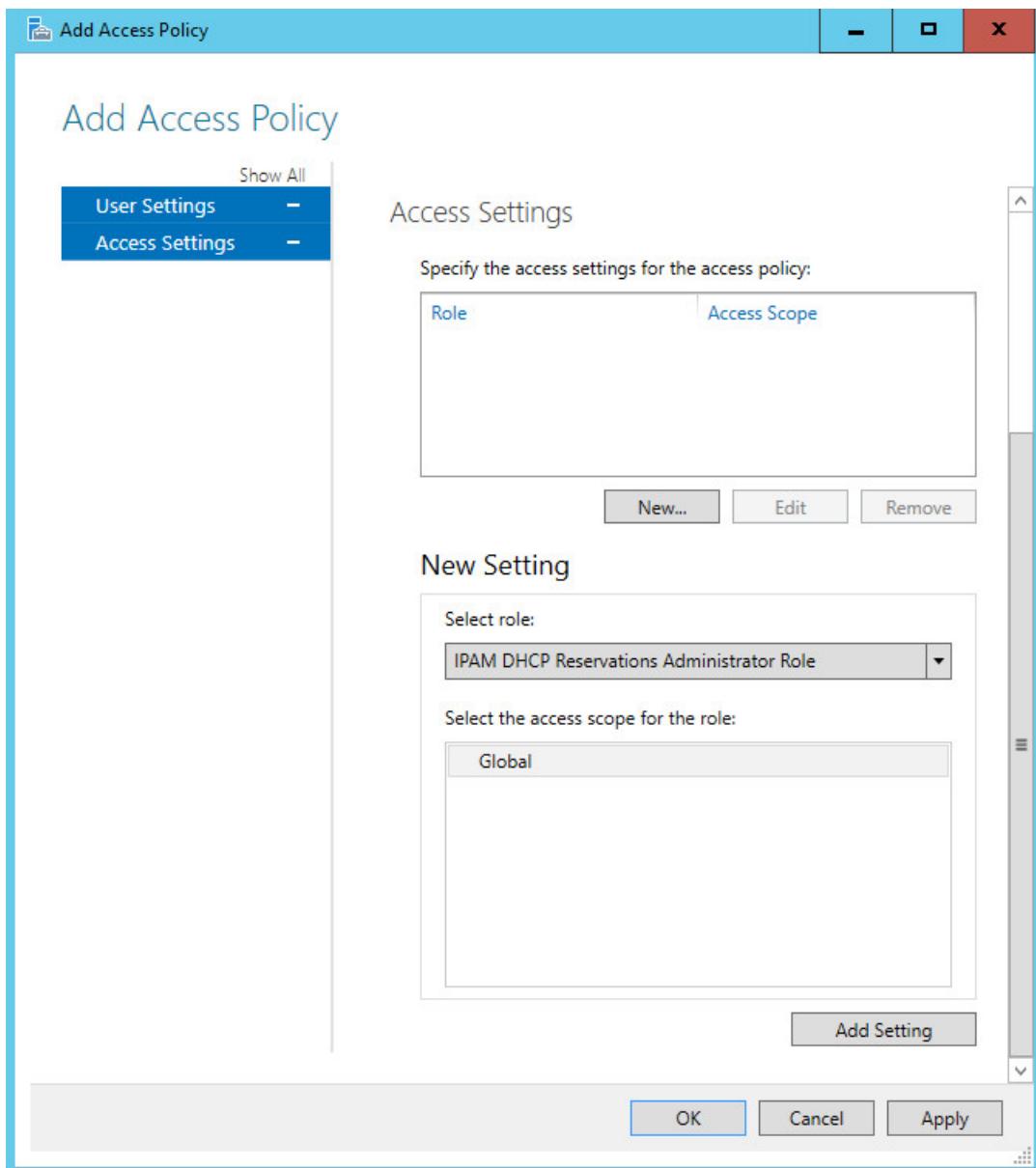
5. The **Locations** dialog box opens. Browse to the location that contains the user account, select the location, and then click **OK**. The **Locations** dialog box closes.



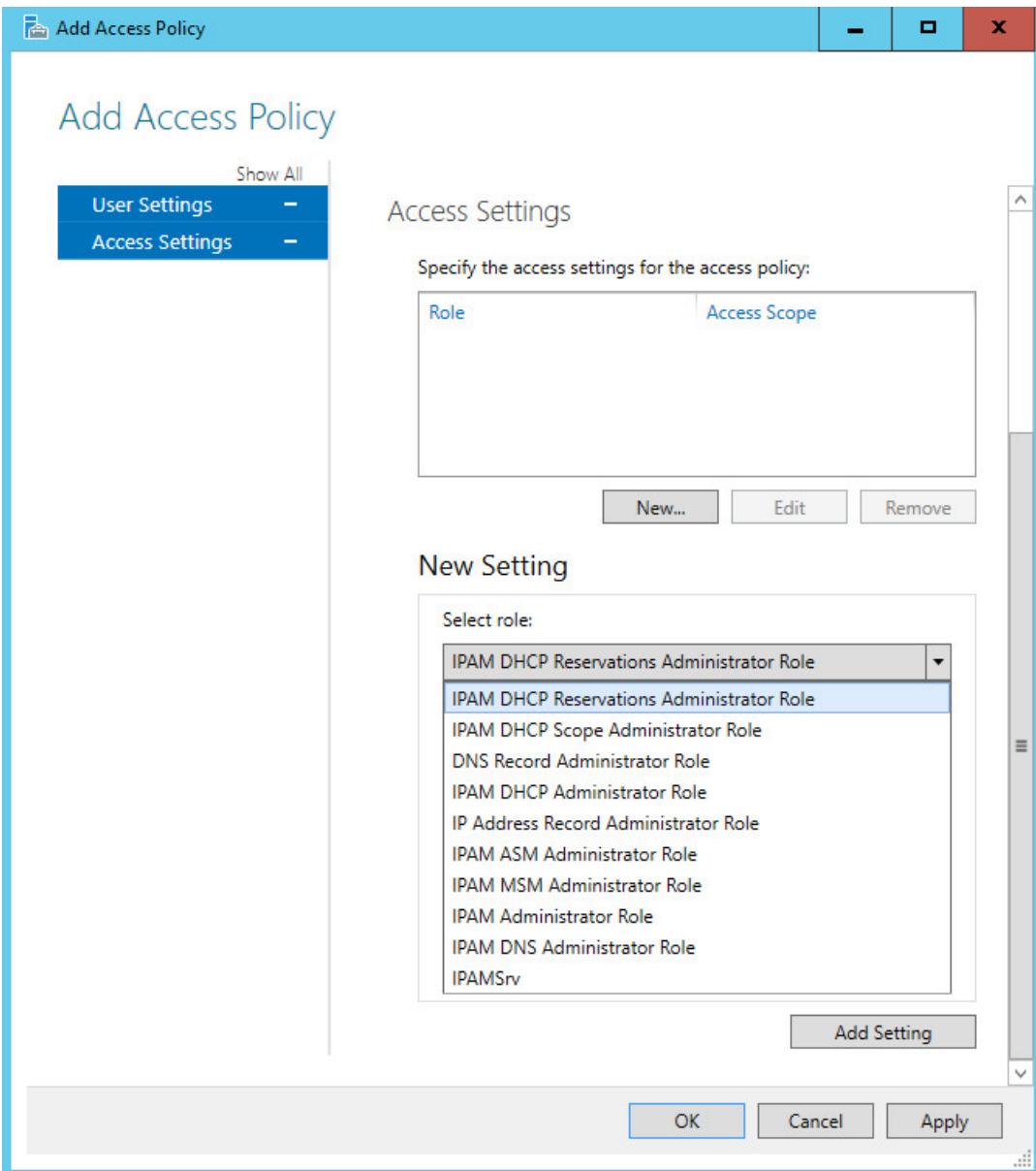
6. In the **Select User or Group** dialog box, in **Enter the object name to select**, type the user account name for which you want to create an access policy. Click **OK**.
7. In **Add Access Policy**, in **User Settings**, **User alias** now contains the user account to which the policy applies. In **Access Settings**, click **New...**.



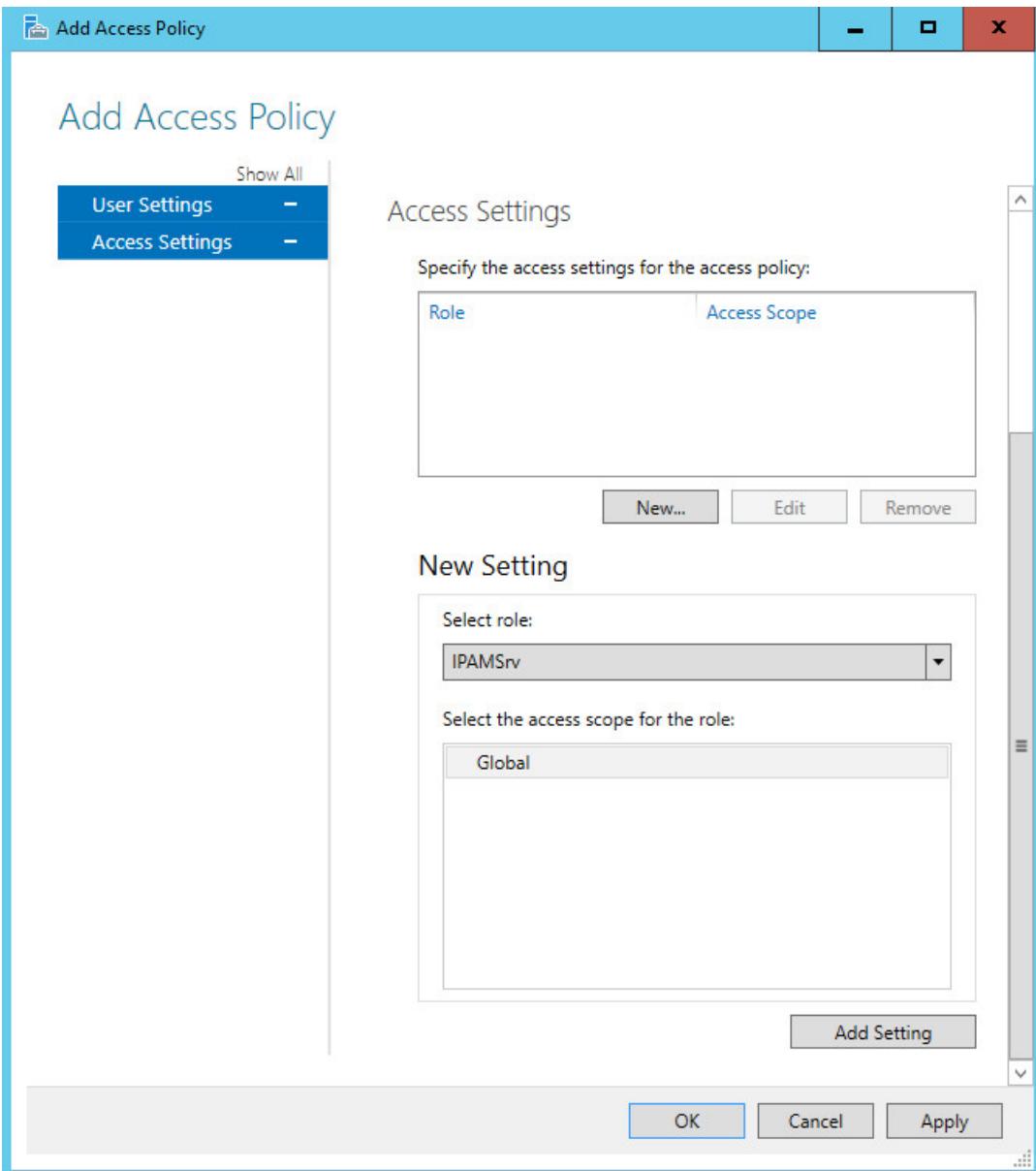
8. In **Add Access Policy**, **Access Settings** changes to **New Setting**.



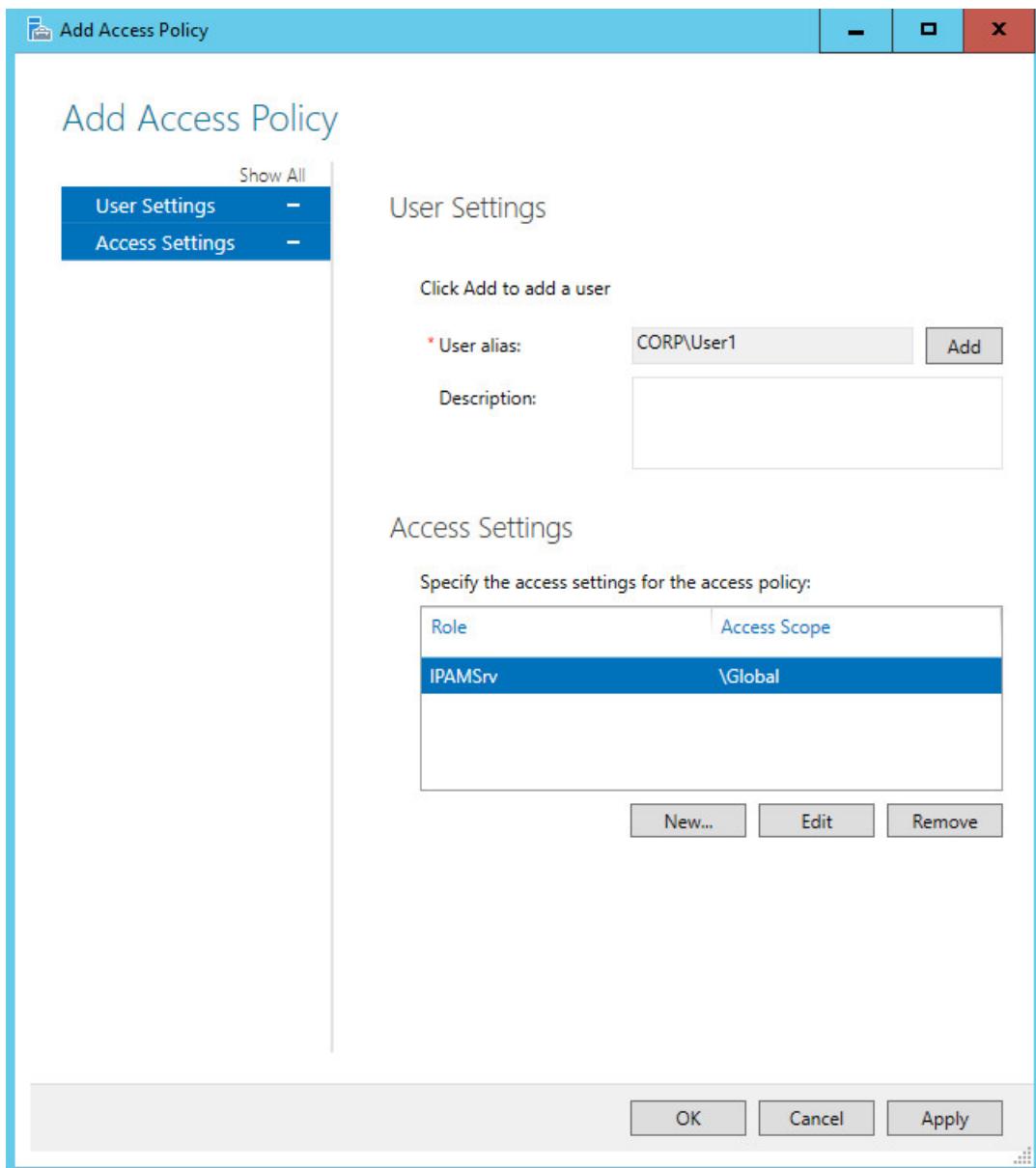
9. Click **Select role** to expand the list of roles. Select one of the built-in roles or, if you have created new roles, select one of the roles that you created. For example, if you created the IPAMSrv role to apply to the user, click **IPAMSrv**.



10. Click **Add Setting**.



11. The role is added to the access policy. To create additional access policies, click **Apply**, and then repeat these steps for each policy that you want to create. If you do not want to create additional policies, click **OK**.



12. In the IPAM client console display pane, verify that the new access policy is created.

Access Policies
Access Policies | 1 total

User Name	Alias	Forest name
User1	CORP\User1	corp.contoso.com

See Also

[Role-based Access Control](#)

[Manage IPAM](#)

Set Access Scope for a DNS Zone

9/1/2018 • 2 minutes to read • [Edit Online](#)

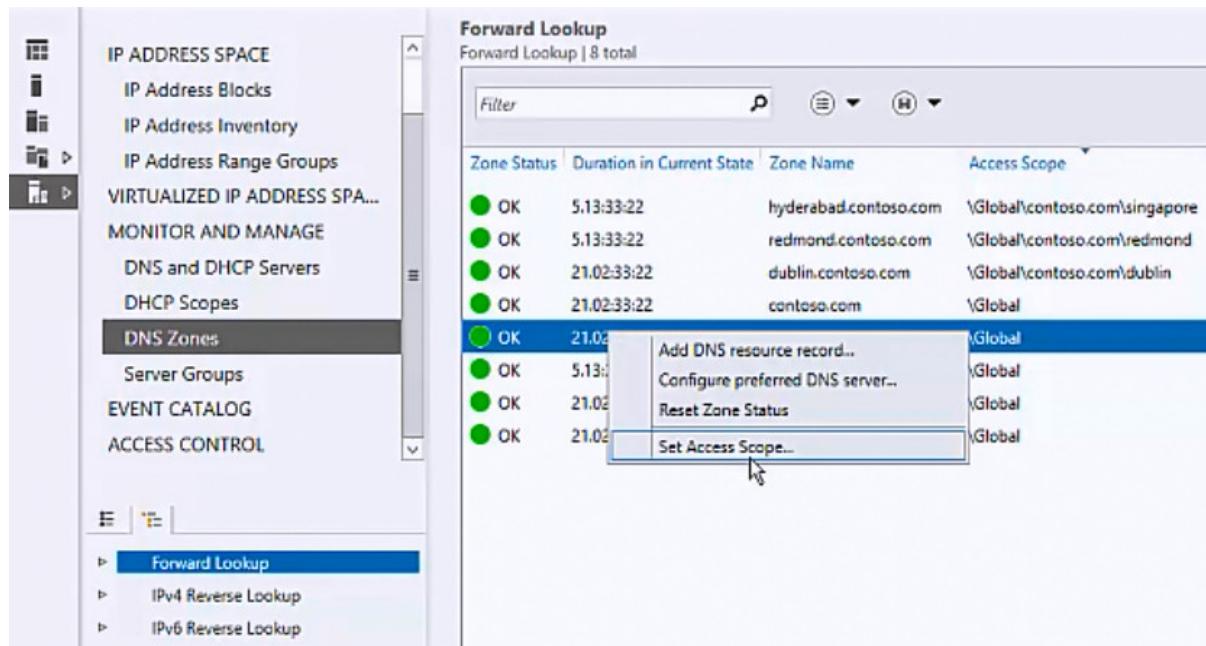
Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016

You can use this topic to set the access scope for a DNS zone by using the IPAM client console.

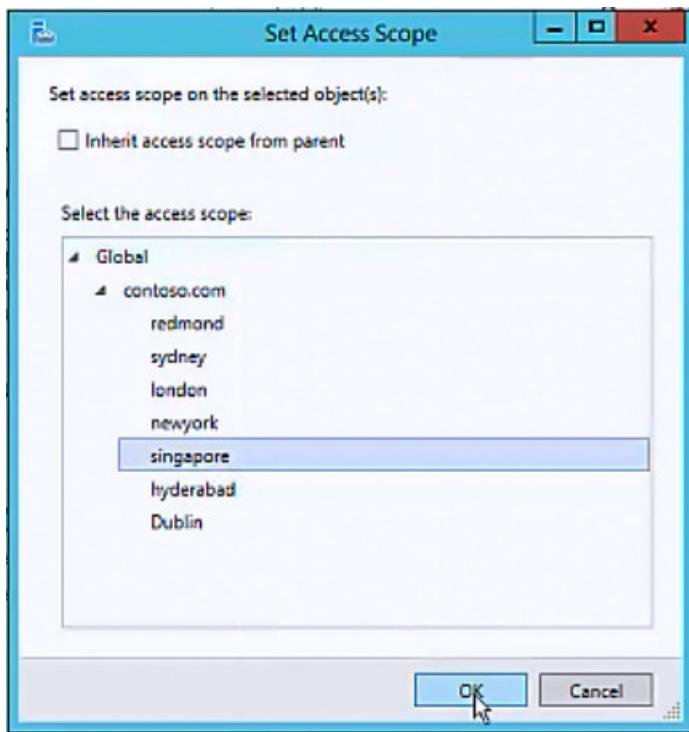
Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To set the access scope for a DNS zone

1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, click **DNS Zones**. In the display pane, right-click the DNS zone for which you want to change the access scope., and then click **Set Access Scope**.



3. The **Set Access Scope** dialog box opens. If required for your deployment, click to deselect **Inherit access scope from parent**. In **Select the access scope**, select an item, and then click **OK**.



4. In the IPAM client console display pane, verify that the access scope for the zone is changed.

Zone Status	Duration in Current State	Zone Name	Access Scope	Preferred
OK	5.13:33:22	hyderabad.contoso.com	\Global\contoso.com\singapore	
OK	5.13:33:22	redmond.contoso.com	\Global\contoso.com\redmond	
OK	21.02:33:22	dublin.contoso.com	\Global\contoso.com\dublin	dc
OK	21.02:33:22	contoso.com	\Global	dc
OK	5.13:33:38	london.contoso.com	\Global\contoso.com\singapore	dc
OK	5.13:33:22	newyork.contoso.com	\Global	
OK	21.02:33:22	singapore.contoso.com	\Global	dc
OK	21.02:33:22	sydney.contoso.com	\Global	dc

See Also

[Role-based Access Control](#)

[Manage IPAM](#)

Set Access Scope for DNS Resource Records

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to set the access scope for a DNS resource records by using the IPAM client console.

Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To set access scope for DNS resource records

1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, click **DNS Zones**. In the lower navigation pane, expand **Forward Lookup** and browse to and select the zone that contains the resource records whose access scope you want to change.
3. In the display pane, locate and select the resource records whose access scope you want to change.

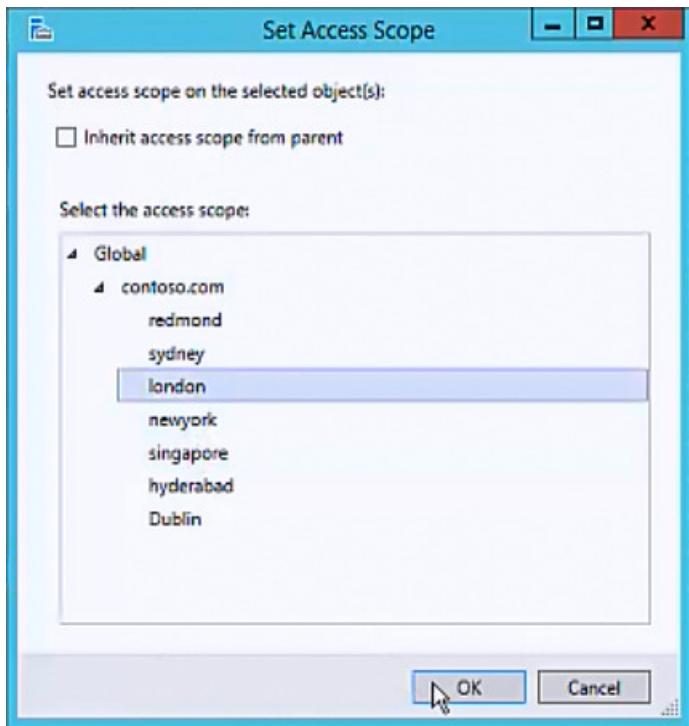
The screenshot shows the IPAM client console. The left navigation pane is expanded to show 'Forward Lookup' under 'DNS Zones'. Under 'Forward Lookup', several zones are listed: contoso.com, dublin, hyderabad, and london. The 'london' zone is selected and highlighted in blue. The right pane displays a list of DNS resource records (resource records) for the 'london' zone. The records are listed in a table with columns: Name, Type, IP Address, and Scope Type. The records are: abc1 (A, 10.1.60.1, Static), abc10 (A, 10.1.60.10, Static), abc11 (A, 10.1.60.11, Static), abc12 (A, 10.1.60.12, Static), abc13 (A, 10.1.60.13, Static), abc14 (A, 10.1.60.14, Static), abc15 (A, 10.1.60.15, Static), abc16 (A, 10.1.60.16, Static), abc17 (A, 10.1.60.17, Static), abc18 (A, 10.1.60.18, Static), abc19 (A, 10.1.60.19, Static), abc2 (A, 10.1.60.2, Static), and abc20 (A, 10.1.60.20, Static). The record 'abc13' is also highlighted in blue.

Name	Type	IP Address	Scope Type
abc1	A	10.1.60.1	Static
abc10	A	10.1.60.10	Static
abc11	A	10.1.60.11	Static
abc12	A	10.1.60.12	Static
abc13	A	10.1.60.13	Static
abc14	A	10.1.60.14	Static
abc15	A	10.1.60.15	Static
abc16	A	10.1.60.16	Static
abc17	A	10.1.60.17	Static
abc18	A	10.1.60.18	Static
abc19	A	10.1.60.19	Static
abc2	A	10.1.60.2	Static
abc20	A	10.1.60.20	Static

4. Right-click the selected DNS resource records, and then click **Set Access Scope**.

The screenshot shows the IPAM client console with the same navigation and resource record list as the previous screenshot. The 'abc13' record is selected. A context menu is open over the selected records, showing options: 'Edit DNS resource record...', 'Create IP Address...', 'Delete DNS resource record...', 'Create Associated DNS resource record...', and 'Set Access Scope...'. The 'Set Access Scope...' option is highlighted with a blue selection bar and has a small arrow indicating it is the active choice.

5. The **Set Access Scope** dialog box opens. If required for your deployment, click to deselect **Inherit access scope from parent**. In **Select the access scope**, select an item, and then click **OK**.



See Also

[Role-based Access Control](#)

[Manage IPAM](#)

View Roles and Role Permissions

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to view Access Control user roles in the IPAM client console.

Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

To view Access Control roles

1. In Server Manager, click **IPAM**. The IPAM client console appears.
2. In the navigation pane, click **ACCESS CONTROL**.
3. In the lower navigation pane, click **Roles**. In the display pane, the roles are listed.

The screenshot shows the IPAM client console interface. On the left, there is a navigation pane with several categories: IP ADDRESS SPACE (IP Address Blocks, IP Address Inventory, IP Address Range Groups), VIRTUALIZED IP ADDRESS SPA..., MONITOR AND MANAGE (DNS and DHCP Servers, DHCP Scopes, DNS Zones, Server Groups), EVENT CATALOG, and ACCESS CONTROL (which is selected). Below these are buttons for Refresh, Stop, and Start. In the bottom left of the navigation pane, the 'Roles' tab is selected, with 'Access Scopes' and 'Access Policies' also listed. The main right pane is titled 'Roles' and shows a table with 9 total entries. The table has columns for 'Name' and 'Built-in Role'. The rows are: DNS Record Administrator Role (Yes), IP Address Record Administrator Role (Yes), IPAM Administrator Role (Yes), IPAM ASM Administrator Role (Yes), IPAM DHCP Administrator Role (Yes), IPAM DHCP Reservations Administrator Role (Yes), IPAM DHCP Scope Administrator Role (Yes), IPAM DNS Administrator Role (Yes), and IPAM MSM Administrator Role (Yes). The 'DNS Record Administrator Role' is currently selected, as indicated by a blue highlight.

Name	Built-in Role
DNS Record Administrator Role	Yes
IP Address Record Administrator Role	Yes
IPAM Administrator Role	Yes
IPAM ASM Administrator Role	Yes
IPAM DHCP Administrator Role	Yes
IPAM DHCP Reservations Administrator Role	Yes
IPAM DHCP Scope Administrator Role	Yes
IPAM DNS Administrator Role	Yes
IPAM MSM Administrator Role	Yes

4. Select the role whose permissions you want to view. In the lower details pane, the operations that are permitted for the role are displayed.

Roles	
Roles 9 total	
Name	Built-in Role
DNS Record Administrator Role	Yes
IP Address Record Administrator Role	Yes
IPAM Administrator Role	Yes
IPAM ASM Administrator Role	Yes
IPAM DHCP Administrator Role	Yes
IPAM DHCP Reservations Administrator Role	Yes
IPAM DHCP Scope Administrator Role	Yes
IPAM DNS Administrator Role	Yes
IPAM MSM Administrator Role	Yes

Details View

DNS Record Administrator Role

Details	Event Catalog
<p>Name: DNS Record Administrator Role</p> <p>Built-in Role: Yes</p> <p>Description: This built-in role provides permissions to manage the DNS resource records.</p> <p>Operations:</p> <ul style="list-style-type: none"> Set access scope on DNS Record Create CNAME record Create DNAME record Create A record Create AAAA record Create MX record Create NS record Create PTR record Create SRV record 	

See Also

[Role-based Access Control](#)

[Manage IPAM](#)

Manage Role Based Access Control with Windows PowerShell

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to use IPAM to manage role based access control with Windows PowerShell.

NOTE

For the IPAM Windows PowerShell command reference, see [IP Address Management \(IPAM\) Server Cmdlets in Windows PowerShell](#).

The new Windows PowerShell IPAM commands provide you with the ability to retrieve and change the access scopes of DNS and DHCP objects. The following table illustrates the correct command to use for each IPAM object.

IPAM OBJECT	COMMAND	DESCRIPTION
DNS Server	Get-IpamDnsServer	This cmdlet returns the DNS server object in IPAM
DNS Zone	Get-IpamDnsZone	This cmdlet returns the DNS zone object in IPAM
DNS Resource Record	Get-IpamResourceRecord	This cmdlet returns the DNS resource record object in IPAM
DNS Conditional Forwarder	Get-IpamDnsConditionalForwarder	This cmdlet returns the DNS conditional forwarder object in IPAM
DHCP Server	Get-IpamDhcpServer	This cmdlet returns the DHCP server object in IPAM
DHCP Superscope	Get-IpamDhcpSuperscope	This cmdlet returns the DHCP superscope object in IPAM
DHCP Scope	Get-IpamDhcpScope	This cmdlet returns the DHCP scope object in IPAM

In the following example of command output, the `Get-IpamDnsZone` cmdlet retrieves the **dublin.contoso.com** DNS zone.

```
PS C:\Users\Administrator.CONTOSO> Get-IpamDnsZone -ZoneType Forward -ZoneName dublin.contoso.com

ZoneName      : dublin.contoso.com
ZoneType     : Forward
AccessScopePath : \Global\Dublin
IsSigned      : False
DynamicUpdateStatus : None
ScavengeStaleRecords : False
```

Setting Access Scopes on IPAM Objects

You can set access scopes on IPAM objects by using the `Set-IpamAccessScope` command. You can use this command to set the access scope to a specific value for an object or to cause the objects to inherit access scope from parent objects. Following are the objects that you can configure with this command.

- DHCP Scope
- DHCP Server
- DHCP Superscope
- DNS Conditional Forwarder
- DNS Resource Records
- DNS Server
- DNS Zone
- IP Address Block
- IP Address Range
- IP Address Space
- IP Address Subnet

Following is the syntax for the `Set-IpamAccessScope` command.

NAME

Set-IpamAccessScope

SYNTAX

```
Set-IpamAccessScope [-IpamRange] -InputObject <ciminstance[]> [-AccessScopePath <string>] [-  
IsInheritedAccessScope] [-PassThru] [-CimSession <CimSession[]>] [-ThrottleLimit <int>] [-AsJob] [-WhatIf] [-  
Confirm] [<CommonParameters>]  
  
Set-IpamAccessScope [-IpamDnsServer] -InputObject <ciminstance[]> [-AccessScopePath <string>] [-  
IsInheritedAccessScope] [-PassThru] [-CimSession <CimSession[]>] [-ThrottleLimit <int>] [-AsJob] [-WhatIf] [-  
Confirm]  
[<CommonParameters>]  
  
Set-IpamAccessScope [-IpamDhcpServer] -InputObject <ciminstance[]> [-AccessScopePath <string>] [-  
IsInheritedAccessScope] [-PassThru] [-CimSession <CimSession[]>] [-ThrottleLimit <int>] [-AsJob] [-WhatIf] [-  
Confirm]  
[<CommonParameters>]  
  
Set-IpamAccessScope [-IpamDhcpSuperscope] -InputObject <ciminstance[]> [-AccessScopePath <string>] [-  
IsInheritedAccessScope] [-PassThru] [-CimSession <CimSession[]>] [-ThrottleLimit <int>] [-AsJob] [-WhatIf] [-  
Confirm]  
[<CommonParameters>]  
  
Set-IpamAccessScope [-IpamDhcpScope] -InputObject <ciminstance[]> [-AccessScopePath <string>] [-  
IsInheritedAccessScope] [-PassThru] [-CimSession <CimSession[]>] [-ThrottleLimit <int>] [-AsJob] [-WhatIf] [-  
Confirm]  
[<CommonParameters>]  
  
Set-IpamAccessScope [-IpamDnsConditionalForwarder] -InputObject <ciminstance[]> [-AccessScopePath <string>]  
[-IsInheritedAccessScope] [-PassThru] [-CimSession <CimSession[]>] [-ThrottleLimit <int>] [-AsJob] [-WhatIf] [-  
Confirm]  
[<CommonParameters>]  
  
Set-IpamAccessScope [-IpamDnsResourceRecord] -InputObject <ciminstance[]> [-AccessScopePath <string>] [-  
IsInheritedAccessScope] [-PassThru] [-CimSession <CimSession[]>] [-ThrottleLimit <int>] [-AsJob] [-WhatIf] [-  
Confirm]  
[<CommonParameters>]  
  
Set-IpamAccessScope [-IpamDnsZone] -InputObject <ciminstance[]> [-AccessScopePath <string>] [-  
IsInheritedAccessScope] [-PassThru] [-CimSession <CimSession[]>] [-ThrottleLimit <int>] [-AsJob] [-WhatIf] [-  
Confirm]  
[<CommonParameters>]  
  
Set-IpamAccessScope [-IpamAddressSpace] -InputObject <ciminstance[]> [-AccessScopePath <string>] [-  
IsInheritedAccessScope] [-PassThru] [-CimSession <CimSession[]>] [-ThrottleLimit <int>] [-AsJob] [-WhatIf] [-  
Confirm]  
[<CommonParameters>]  
  
Set-IpamAccessScope [-IpamSubnet] -InputObject <ciminstance[]> [-AccessScopePath <string>] [-  
IsInheritedAccessScope] [-PassThru] [-CimSession <CimSession[]>] [-ThrottleLimit <int>] [-AsJob] [-WhatIf] [-  
Confirm] [<CommonParameters>]  
  
Set-IpamAccessScope [-IpamBlock] -InputObject <ciminstance[]> [-AccessScopePath <string>] [-  
IsInheritedAccessScope] [-PassThru] [-CimSession <CimSession[]>] [-ThrottleLimit <int>] [-AsJob] [-WhatIf] [-  
Confirm] [<CommonParameters>]
```

In the following example, the access scope of the DNS zone **dublin.contoso.com** is changed from **Dublin** to **Europe**.

```
PS C:\Users\Administrator.CONTOSO> Get-IpamDnsZone -ZoneType Forward -ZoneName dublin.contoso.com

ZoneName      : dublin.contoso.com
ZoneType      : Forward
AccessScopePath : \Global\Dublin
IsSigned      : False
DynamicUpdateStatus : None
ScavengeStaleRecords : False

PS C:\Users\Administrator.CONTOSO> $a = Get-IpamDnsZone -ZoneType Forward -ZoneName dublin.contoso.com
PS C:\Users\Administrator.CONTOSO> Set-IpamAccessScope -IpamDnsZone $a -AccessScopePath
\Global\Europe -PassThru

ZoneName      : dublin.contoso.com
ZoneType      : Forward
AccessScopePath : \Global\Europe
IsSigned      : False
DynamicUpdateStatus : None
ScavengeStaleRecords : False
```

Network Load Balancing

9/21/2018 • 7 minutes to read • [Edit Online](#)

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

In this topic, we provide you with an overview of the Network Load Balancing (NLB) feature in Windows Server 2016. You can use NLB to manage two or more servers as a single virtual cluster. NLB enhances the availability and scalability of Internet server applications such as those used on web, FTP, firewall, proxy, virtual private network (VPN), and other mission-critical servers.

NOTE

Windows Server 2016 includes a new Azure-inspired Software Load Balancer (SLB) as a component of the Software Defined Networking (SDN) infrastructure. Use SLB instead of NLB if you are using SDN, are using non-Windows workloads, need outbound network address translation (NAT), or need Layer 3 (L3) or non-TCP based load balancing. You can continue to use NLB with Windows Server 2016 for non-SDN deployments. For more information about SLB, see [Software Load Balancing \(SLB\) for SDN](#).

The Network Load Balancing (NLB) feature distributes traffic across several servers by using the TCP/IP networking protocol. By combining two or more computers that are running applications into a single virtual cluster, NLB provides reliability and performance for web servers and other mission-critical servers.

The servers in an NLB cluster are called *hosts*, and each host runs a separate copy of the server applications. NLB distributes incoming client requests across the hosts in the cluster. You can configure the load that is to be handled by each host. You can also add hosts dynamically to the cluster to handle increased load. NLB can also direct all traffic to a designated single host, which is called the *default host*.

NLB allows all of the computers in the cluster to be addressed by the same set of IP addresses, and it maintains a set of unique, dedicated IP addresses for each host. For load-balanced applications, when a host fails or goes offline, the load is automatically redistributed among the computers that are still operating. When it is ready, the offline computer can transparently rejoin the cluster and regain its share of the workload, which allows the other computers in the cluster to handle less traffic.

Practical applications

NLB is useful for ensuring that stateless applications, such as web servers running Internet Information Services (IIS), are available with minimal downtime, and that they are scalable (by adding additional servers as the load increases). The following sections describe how NLB supports high availability, scalability, and manageability of the clustered servers that run these applications.

High availability

A high availability system reliably provides an acceptable level of service with minimal downtime. To provide high availability, NLB includes built-in features that can automatically:

- Detect a cluster host that fails or goes offline, and then recover.
- Balance the network load when hosts are added or removed.
- Recover and redistribute the workload within ten seconds.

Scalability

Scalability is the measure of how well a computer, service, or application can grow to meet increasing performance demands. For NLB clusters, scalability is the ability to incrementally add one or more systems to an existing cluster when the overall load of the cluster exceeds its capabilities. To support scalability, you can do the following with NLB:

- Balance load requests across the NLB cluster for individual TCP/IP services.
- Support up to 32 computers in a single cluster.
- Balance multiple server load requests (from the same client or from several clients) across multiple hosts in the cluster.
- Add hosts to the NLB cluster as the load increases, without causing the cluster to fail.
- Remove hosts from the cluster when the load decreases.
- Enable high performance and low overhead through a fully pipelined implementation. Pipelining allows requests to be sent to the NLB cluster without waiting for a response to a previous request.

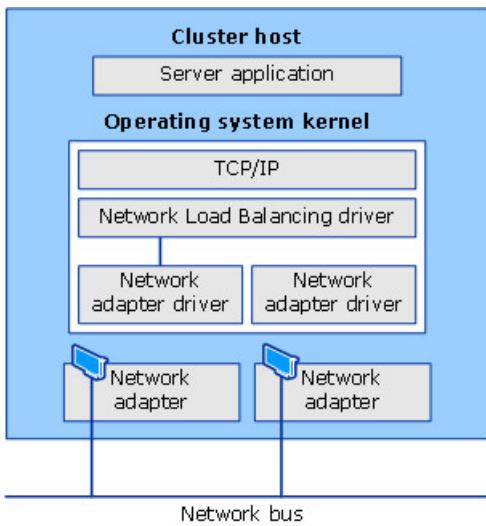
Manageability

To support manageability, you can do the following with NLB:

- Manage and configure multiple NLB clusters and the cluster hosts from a single computer by using NLB Manager or the [Network Load Balancing \(NLB\) Cmdlets in Windows PowerShell](#).
- Specify the load balancing behavior for a single IP port or group of ports by using port management rules.
- Define different port rules for each website. If you use the same set of load-balanced servers for multiple applications or websites, port rules are based on the destination virtual IP address (using virtual clusters).
- Direct all client requests to a single host by using optional, single-host rules. NLB routes client requests to a particular host that is running specific applications.
- Block undesired network access to certain IP ports.
- Enable Internet Group Management Protocol (IGMP) support on the cluster hosts to control switch port flooding (where incoming network packets are sent to all ports on the switch) when operating in multicast mode.
- Start, stop, and control NLB actions remotely by using Windows PowerShell commands or scripts.
- View the Windows Event Log to check NLB events. NLB logs all actions and cluster changes in the event log.

Important functionality

NLB is installed as a standard Windows Server networking driver component. Its operations are transparent to the TCP/IP networking stack. The following figure shows the relationship between NLB and other software components in a typical configuration.



Following are the primary features of NLB.

- Requires no hardware changes to run.
- Provides Network Load Balancing Tools to configure and manage multiple clusters and all of the hosts from a single remote or local computer.
- Enables clients to access the cluster by using a single, logical Internet name and virtual IP address, which is known as the cluster IP address (it retains individual names for each computer). NLB allows multiple virtual IP addresses for multihomed servers.

NOTE

When you deploy VMs as virtual clusters, NLB does not require servers to be multihomed to have multiple virtual IP addresses.

- Enables NLB to be bound to multiple network adapters, which enables you to configure multiple independent clusters on each host. Support for multiple network adapters differs from virtual clusters in that virtual clusters allow you to configure multiple clusters on a single network adapter.
- Requires no modifications to server applications so that they can run in an NLB cluster.
- Can be configured to automatically add a host to the cluster if that cluster host fails and is subsequently brought back online. The added host can start handling new server requests from clients.
- Enables you to take computers offline for preventive maintenance without disturbing the cluster operations on the other hosts.

Hardware requirements

Following are the hardware requirements to run an NLB cluster.

- All hosts in the cluster must reside on the same subnet.
- There is no restriction on the number of network adapters on each host, and different hosts can have a different number of adapters.
- Within each cluster, all network adapters must be either multicast or unicast. NLB does not support a mixed environment of multicast and unicast within a single cluster.
- If you use the unicast mode, the network adapter that is used to handle client-to-cluster traffic must support changing its media access control (MAC) address.

Software requirements

Following are the software requirements to run an NLB cluster.

- Only TCP/IP can be used on the adapter for which NLB is enabled on each host. Do not add any other protocols (for example, IPX) to this adapter.
- The IP addresses of the servers in the cluster must be static.

NOTE

NLB does not support Dynamic Host Configuration Protocol (DHCP). NLB disables DHCP on each interface that it configures.

Installation information

You can install NLB by using either Server Manager or the Windows PowerShell commands for NLB.

Optionally you can install the Network Load Balancing Tools to manage a local or remote NLB cluster. The tools include Network Load Balancing Manager and the NLB Windows PowerShell commands.

Installation with Server Manager

In Server Manager, you can use the Add Roles and Features Wizard to add the **Network Load Balancing** feature. When you complete the wizard, NLB is installed, and you do not need to restart the computer.

Installation with Windows PowerShell

To install NLB by using Windows PowerShell, run the following command at an elevated Windows PowerShell prompt on the computer where you want to install NLB.

```
Install-WindowsFeature NLB -IncludeManagementTools
```

After installation is complete, no restart of the computer is required.

For more information, see [Install-WindowsFeature](#).

Network Load Balancing Manager

To open Network Load Balancing Manager in Server Manager, click **Tools**, and then click **Network Load Balancing Manager**.

Additional resources

The following table provides links to additional information about the NLB feature.

CONTENT TYPE	REFERENCES
Deployment	Network Load Balancing Deployment Guide Configuring Network Load Balancing with Terminal Services
Operations	Managing Network Load Balancing Clusters Setting Network Load Balancing Parameters Controlling Hosts on Network Load Balancing Clusters
Troubleshooting	Troubleshooting Network Load Balancing Clusters NLB Cluster Events and Errors
Tools and settings	Network Load Balancing Windows PowerShell cmdlets

CONTENT TYPE	REFERENCES
Community resources	High Availability (Clustering) Forum

Network Policy Server (NPS)

9/21/2018 • 12 minutes to read • [Edit Online](#)

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

You can use this topic for an overview of Network Policy Server in Windows Server 2016.

NOTE

In addition to this topic, the following NPS documentation is available.

- [Network Policy Server Best Practices](#)
- [Getting Started with Network Policy Server](#)
- [Plan Network Policy Server](#)
- [Deploy Network Policy Server](#)
- [Manage Network Policy Server](#)
- [Network Policy Server \(NPS\) Cmdlets in Windows PowerShell](#) for Windows Server 2016 and Windows 10
- [Network Policy Server \(NPS\) Cmdlets in Windows PowerShell](#) for Windows Server 2012 R2 and Windows 8.1
- [NPS Cmdlets in Windows PowerShell](#) for Windows Server 2012 and Windows 8

Network Policy Server (NPS) allows you to create and enforce organization-wide network access policies for connection request authentication and authorization.

You can also configure NPS as a Remote Authentication Dial-In User Service (RADIUS) proxy to forward connection requests to a remote NPS or other RADIUS server so that you can load balance connection requests and forward them to the correct domain for authentication and authorization.

NPS allows you to centrally configure and manage network access authentication, authorization, and accounting with the following features:

- **RADIUS server.** NPS performs centralized authentication, authorization, and accounting for wireless, authenticating switch, remote access dial-up and virtual private network (VPN) connections. When you use NPS as a RADIUS server, you configure network access servers, such as wireless access points and VPN servers, as RADIUS clients in NPS. You also configure network policies that NPS uses to authorize connection requests, and you can configure RADIUS accounting so that NPS logs accounting information to log files on the local hard disk or in a Microsoft SQL Server database. For more information, see [RADIUS server](#).
- **RADIUS proxy.** When you use NPS as a RADIUS proxy, you configure connection request policies that tell the NPS which connection requests to forward to other RADIUS servers and to which RADIUS servers you want to forward connection requests. You can also configure NPS to forward accounting data to be logged by one or more computers in a remote RADIUS server group. To configure NPS as a RADIUS proxy server, see the following topics. For more information, see [RADIUS proxy](#).
 - [Configure Connection Request Policies](#)
- **RADIUS accounting.** You can configure NPS to log events to a local log file or to a local or remote instance of Microsoft SQL Server. For more information, see [NPS logging](#).

IMPORTANT

Network Access Protection (NAP), Health Registration Authority (HRA), and Host Credential Authorization Protocol (HCAP) were deprecated in Windows Server 2012 R2, and are not available in Windows Server 2016. If you have a NAP deployment using operating systems earlier than Windows Server 2016, you cannot migrate your NAP deployment to Windows Server 2016.

You can configure NPS with any combination of these features. For example, you can configure one NPS as a RADIUS server for VPN connections and also as a RADIUS proxy to forward some connection requests to members of a remote RADIUS server group for authentication and authorization in another domain.

Windows Server Editions and NPS

NPS provides different functionality depending on the edition of Windows Server that you install.

Windows Server 2016 Standard/Datacenter Edition

With NPS in Windows Server 2016 Standard or Datacenter, you can configure an unlimited number of RADIUS clients and remote RADIUS server groups. In addition, you can configure RADIUS clients by specifying an IP address range.

The following sections provide more detailed information about NPS as a RADIUS server and proxy.

RADIUS server and proxy

You can use NPS as a RADIUS server, a RADIUS proxy, or both.

RADIUS server

NPS is the Microsoft implementation of the RADIUS standard specified by the Internet Engineering Task Force (IETF) in RFCs 2865 and 2866. As a RADIUS server, NPS performs centralized connection authentication, authorization, and accounting for many types of network access, including wireless, authenticating switch, dial-up and virtual private network (VPN) remote access, and router-to-router connections.

NOTE

For information on deploying NPS as a RADIUS server, see [Deploy Network Policy Server](#).

NPS enables the use of a heterogeneous set of wireless, switch, remote access, or VPN equipment. You can use NPS with the Remote Access service, which is available in Windows Server 2016.

NPS uses an Active Directory Domain Services (AD DS) domain or the local Security Accounts Manager (SAM) user accounts database to authenticate user credentials for connection attempts. When a server running NPS is a member of an AD DS domain, NPS uses the directory service as its user account database and is part of a single sign-on solution. The same set of credentials is used for network access control (authenticating and authorizing access to a network) and to log on to an AD DS domain.

NOTE

NPS uses the dial-in properties of the user account and network policies to authorize a connection.

Internet service providers (ISPs) and organizations that maintain network access have the increased challenge of managing all types of network access from a single point of administration, regardless of the type of network access equipment used. The RADIUS standard supports this functionality in both homogeneous and heterogeneous environments. RADIUS is a client-server protocol that enables network access equipment (used

as RADIUS clients) to submit authentication and accounting requests to a RADIUS server.

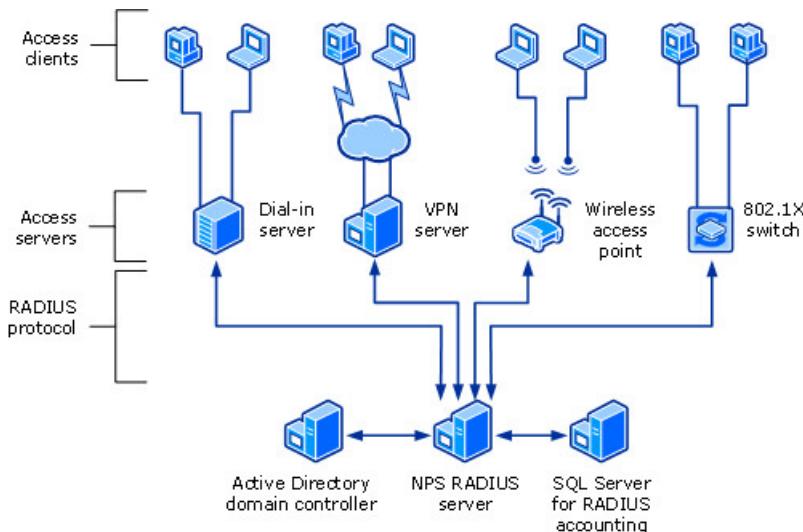
A RADIUS server has access to user account information and can check network access authentication credentials. If user credentials are authenticated and the connection attempt is authorized, the RADIUS server authorizes user access on the basis of specified conditions, and then logs the network access connection in an accounting log. The use of RADIUS allows the network access user authentication, authorization, and accounting data to be collected and maintained in a central location, rather than on each access server.

Using NPS as a RADIUS server

You can use NPS as a RADIUS server when:

- You are using an AD DS domain or the local SAM user accounts database as your user account database for access clients.
- You are using Remote Access on multiple dial-up servers, VPN servers, or demand-dial routers and you want to centralize both the configuration of network policies and connection logging and accounting.
- You are outsourcing your dial-up, VPN, or wireless access to a service provider. The access servers use RADIUS to authenticate and authorize connections that are made by members of your organization.
- You want to centralize authentication, authorization, and accounting for a heterogeneous set of access servers.

The following illustration shows NPS as a RADIUS server for a variety of access clients.



RADIUS proxy

As a RADIUS proxy, NPS forwards authentication and accounting messages to NPS and other RADIUS servers. You can use NPS as a RADIUS proxy to provide the routing of RADIUS messages between RADIUS clients (also called network access servers) and RADIUS servers that perform user authentication, authorization, and accounting for the connection attempt.

When used as a RADIUS proxy, NPS is a central switching or routing point through which RADIUS access and accounting messages flow. NPS records information in an accounting log about the messages that are forwarded.

Using NPS as a RADIUS proxy

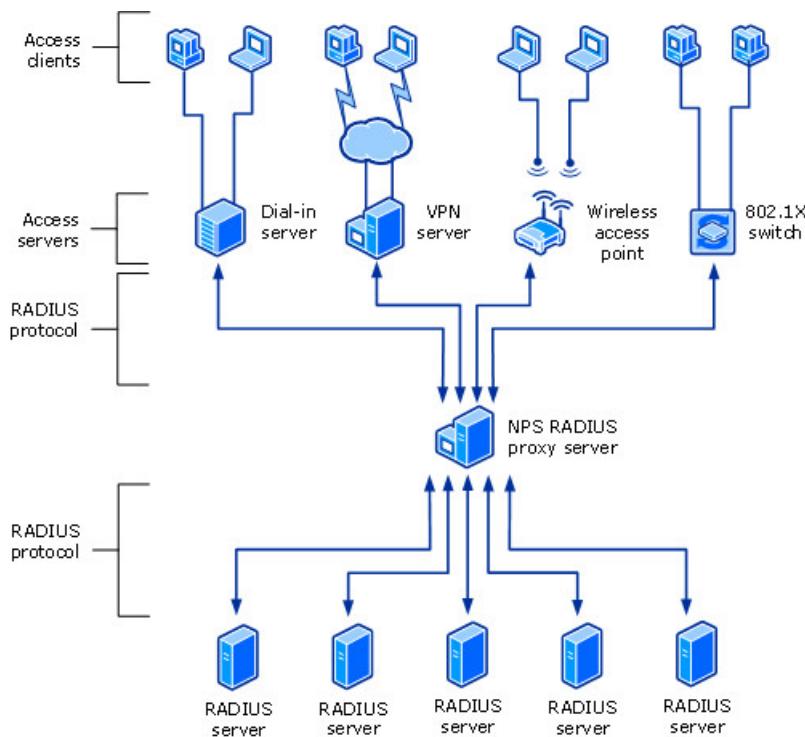
You can use NPS as a RADIUS proxy when:

- You are a service provider who offers outsourced dial-up, VPN, or wireless network access services to multiple customers. Your NASs send connection requests to the NPS RADIUS proxy. Based on the realm portion of the user name in the connection request, the NPS RADIUS proxy forwards the connection request to a RADIUS server that is maintained by the customer and can authenticate and authorize the connection attempt.
- You want to provide authentication and authorization for user accounts that are not members of either the

domain in which the NPS is a member or another domain that has a two-way trust with the domain in which the NPS is a member. This includes accounts in untrusted domains, one-way trusted domains, and other forests. Instead of configuring your access servers to send their connection requests to an NPS RADIUS server, you can configure them to send their connection requests to an NPS RADIUS proxy. The NPS RADIUS proxy uses the realm name portion of the user name and forwards the request to an NPS in the correct domain or forest. Connection attempts for user accounts in one domain or forest can be authenticated for NASs in another domain or forest.

- You want to perform authentication and authorization by using a database that is not a Windows account database. In this case, connection requests that match a specified realm name are forwarded to a RADIUS server, which has access to a different database of user accounts and authorization data. Examples of other user databases include Novell Directory Services (NDS) and Structured Query Language (SQL) databases.
- You want to process a large number of connection requests. In this case, instead of configuring your RADIUS clients to attempt to balance their connection and accounting requests across multiple RADIUS servers, you can configure them to send their connection and accounting requests to an NPS RADIUS proxy. The NPS RADIUS proxy dynamically balances the load of connection and accounting requests across multiple RADIUS servers and increases the processing of large numbers of RADIUS clients and authentications per second.
- You want to provide RADIUS authentication and authorization for outsourced service providers and minimize intranet firewall configuration. An intranet firewall is between your perimeter network (the network between your intranet and the Internet) and intranet. By placing an NPS on your perimeter network, the firewall between your perimeter network and intranet must allow traffic to flow between the NPS and multiple domain controllers. By replacing the NPS with an NPS proxy, the firewall must allow only RADIUS traffic to flow between the NPS proxy and one or multiple NPSs within your intranet.

The following illustration shows NPS as a RADIUS proxy between RADIUS clients and RADIUS servers.



With NPS, organizations can also outsource remote access infrastructure to a service provider while retaining control over user authentication, authorization, and accounting.

NPS configurations can be created for the following scenarios:

- Wireless access
- Organization dial-up or virtual private network (VPN) remote access
- Outsourced dial-up or wireless access

- Internet access
- Authenticated access to extranet resources for business partners

RADIUS server and RADIUS proxy configuration examples

The following configuration examples demonstrate how you can configure NPS as a RADIUS server and a RADIUS proxy.

NPS as a RADIUS server. In this example, NPS is configured as a RADIUS server, the default connection request policy is the only configured policy, and all connection requests are processed by the local NPS. The NPS can authenticate and authorize users whose accounts are in the domain of the NPS and in trusted domains.

NPS as a RADIUS proxy. In this example, the NPS is configured as a RADIUS proxy that forwards connection requests to remote RADIUS server groups in two untrusted domains. The default connection request policy is deleted, and two new connection request policies are created to forward requests to each of the two untrusted domains. In this example, NPS does not process any connection requests on the local server.

NPS as both RADIUS server and RADIUS proxy. In addition to the default connection request policy, which designates that connection requests are processed locally, a new connection request policy is created that forwards connection requests to an NPS or other RADIUS server in an untrusted domain. This second policy is named the Proxy policy. In this example, the Proxy policy appears first in the ordered list of policies. If the connection request matches the Proxy policy, the connection request is forwarded to the RADIUS server in the remote RADIUS server group. If the connection request does not match the Proxy policy but does match the default connection request policy, NPS processes the connection request on the local server. If the connection request does not match either policy, it is discarded.

NPS as a RADIUS server with remote accounting servers. In this example, the local NPS is not configured to perform accounting and the default connection request policy is revised so that RADIUS accounting messages are forwarded to an NPS or other RADIUS server in a remote RADIUS server group. Although accounting messages are forwarded, authentication and authorization messages are not forwarded, and the local NPS performs these functions for the local domain and all trusted domains.

NPS with remote RADIUS to Windows user mapping. In this example, NPS acts as both a RADIUS server and as a RADIUS proxy for each individual connection request by forwarding the authentication request to a remote RADIUS server while using a local Windows user account for authorization. This configuration is implemented by configuring the Remote RADIUS to Windows User Mapping attribute as a condition of the connection request policy. (In addition, a user account must be created locally on the RADIUS server that has the same name as the remote user account against which authentication is performed by the remote RADIUS server.)

Configuration

To configure NPS as a RADIUS server, you can use either standard configuration or advanced configuration in the NPS console or in Server Manager. To configure NPS as a RADIUS proxy, you must use advanced configuration.

Standard configuration

With standard configuration, wizards are provided to help you configure NPS for the following scenarios:

- RADIUS server for dial-up or VPN connections
- RADIUS server for 802.1X wireless or wired connections

To configure NPS using a wizard, open the NPS console, select one of the preceding scenarios, and then click the link that opens the wizard.

Advanced configuration

When you use advanced configuration, you manually configure NPS as a RADIUS server or RADIUS proxy.

To configure NPS by using advanced configuration, open the NPS console, and then click the arrow next to **Advanced Configuration** to expand this section.

The following advanced configuration items are provided.

Configure RADIUS server

To configure NPS as a RADIUS server, you must configure RADIUS clients, network policy, and RADIUS accounting.

For instructions on making these configurations, see the following topics.

- [Configure RADIUS Clients](#)
- [Configure Network Policies](#)
- [Configure Network Policy Server Accounting](#)

Configure RADIUS proxy

To configure NPS as a RADIUS proxy, you must configure RADIUS clients, remote RADIUS server groups, and connection request policies.

For instructions on making these configurations, see the following topics.

- [Configure RADIUS Clients](#)
- [Configure Remote RADIUS Server Groups](#)
- [Configure Connection Request Policies](#)

NPS logging

NPS logging is also called RADIUS accounting. Configure NPS logging to your requirements whether NPS is used as a RADIUS server, proxy, or any combination of these configurations.

To configure NPS logging, you must configure which events you want logged and viewed with Event Viewer, and then determine which other information you want to log. In addition, you must decide whether you want to log user authentication and accounting information to text log files stored on the local computer or to a SQL Server database on either the local computer or a remote computer.

For more information, see [Configure Network Policy Server Accounting](#).

Network Policy Server Best Practices

9/1/2018 • 7 minutes to read • [Edit Online](#)

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

You can use this topic to learn about best practices for deploying and managing Network Policy Server (NPS).

The following sections provide best practices for different aspects of your NPS deployment.

Accounting

Following are the best practices for NPS logging.

There are two types of accounting, or logging, in NPS:

- Event logging for NPS. You can use event logging to record NPS events in the system and security event logs. This is used primarily for auditing and troubleshooting connection attempts.
- Logging user authentication and accounting requests. You can log user authentication and accounting requests to log files in text format or database format, or you can log to a stored procedure in a SQL Server 2000 database. Request logging is used primarily for connection analysis and billing purposes, and is also useful as a security investigation tool, providing you with a method of tracking down the activity of an attacker.

To make the most effective use of NPS logging:

- Turn on logging (initially) for both authentication and accounting records. Modify these selections after you have determined what is appropriate for your environment.
- Ensure that event logging is configured with a capacity that is sufficient to maintain your logs.
- Back up all log files on a regular basis because they cannot be recreated when they are damaged or deleted.
- Use the RADIUS Class attribute to both track usage and simplify the identification of which department or user to charge for usage. Although the automatically generated Class attribute is unique for each request, duplicate records might exist in cases where the reply to the access server is lost and the request is resent. You might need to delete duplicate requests from your logs to accurately track usage.
- If your network access servers and RADIUS proxy servers periodically send fictional connection request messages to NPS to verify that the NPS is online, use the **ping user-name** registry setting. This setting configures NPS to automatically reject these false connection requests without processing them. In addition, NPS does not record transactions involving the fictional user name in any log files, which makes the event log easier to interpret.
- Disable NAS Notification Forwarding. You can disable the forwarding of start and stop messages from network access servers (NASs) to members of a remote RADIUS server group THAT IS configured in NPS. For more information, see [Disable NAS Notification Forwarding](#).

For more information, see [Configure Network Policy Server Accounting](#).

- To provide failover and redundancy with SQL Server logging, place two computers running SQL Server on different subnets. Use the SQL Server **Create Publication Wizard** to set up database replication between the two servers. For more information, see [SQL Server Technical Documentation](#) and [SQL Server Replication](#).

Authentication

Following are the best practices for authentication.

- Use certificate-based authentication methods such as Protected Extensible Authentication Protocol (PEAP) and Extensible Authentication Protocol (EAP) for strong authentication. Do not use password-only authentication methods because they are vulnerable to a variety of attacks and are not secure. For secure wireless authentication, using PEAP-MS-CHAP v2 is recommended, because the NPS proves its identity to wireless clients by using a server certificate, while users prove their identity with their user name and password. For more information about using NPS in your wireless deployment, see [Deploy Password-Based 802.1X Authenticated Wireless Access](#).
- Deploy your own certification authority (CA) with Active Directory® Certificate Services (AD CS) when you use strong certificate-based authentication methods, such as PEAP and EAP, that require the use of a server certificate on NPSs. You can also use your CA to enroll computer certificates and user certificates. For more information on deploying server certificates to NPS and Remote Access servers, see [Deploy Server Certificates for 802.1X Wired and Wireless Deployments](#).

Client computer configuration

Following are the best practices for client computer configuration.

- Automatically configure all of your domain member 802.1X client computers by using Group Policy. For more information, see the section "Configure Wireless Network (IEEE 802.11) Policies" in the topic [Wireless Access Deployment](#).

Installation suggestions

Following are the best practices for installing NPS.

- Before installing NPS, install and test each of your network access servers using local authentication methods before you configure them as RADIUS clients in NPS.
- After you install and configure NPS, save the configuration by using the Windows PowerShell command [Export-NpsConfiguration](#). Save the NPS configuration with this command each time you reconfigure the NPS.

Caution

- The exported NPS configuration file contains unencrypted shared secrets for RADIUS clients and members of remote RADIUS server groups. Because of this, make sure that you save the file to a secure location.
- The export process does not include logging settings for Microsoft SQL Server in the exported file. If you import the exported file to another NPS, you must manually configure SQL Server Logging on the new server.

Performance tuning NPS

Following are the best practices for performance tuning NPS.

- To optimize NPS authentication and authorization response times and minimize network traffic, install NPS on a domain controller.
- When universal principal names (UPNs) or Windows Server 2008 and Windows Server 2003 domains are used, NPS uses the global catalog to authenticate users. To minimize the time it takes to do this, install NPS on either a global catalog server or a server that is on the same subnet as the global catalog server.
- When you have remote RADIUS server groups configured and, in NPS Connection Request Policies, you clear the **Record accounting information on the servers in the following remote RADIUS server group** check box, these groups are still sent network access server (NAS) start and stop notification

messages. This creates unnecessary network traffic. To eliminate this traffic, disable NAS notification forwarding for individual servers in each remote RADIUS server group by clearing the **Forward network start and stop notifications to this server** check box.

Using NPS in large organizations

Following are the best practices for using NPS in large organizations.

- If you are using network policies to restrict access for all but certain groups, create a universal group for all of the users for whom you want to allow access, and then create a network policy that grants access for this universal group. Do not put all of your users directly into the universal group, especially if you have a large number of them on your network. Instead, create separate groups that are members of the universal group, and add users to those groups.
- Use a user principal name to refer to users whenever possible. A user can have the same user principal name regardless of domain membership. This practice provides scalability that might be required in organizations with a large number of domains.
- If you installed Network Policy Server (NPS) on a computer other than a domain controller and the NPS is receiving a large number of authentication requests per second, you can improve NPS performance by increasing the number of concurrent authentications allowed between the NPS and the domain controller. For more information, see

Security issues

Following are the best practices for reducing security issues.

When you are administering a NPS remotely, do not send sensitive or confidential data (for example, shared secrets or passwords) over the network in plaintext. There are two recommended methods for remote administration of NPSs:

- Use Remote Desktop Services to access the NPS. When you use Remote Desktop Services, data is not sent between client and server. Only the user interface of the server (for example, the operating system desktop and NPS console image) is sent to the Remote Desktop Services client, which is named Remote Desktop Connection in Windows® 10. The client sends keyboard and mouse input, which is processed locally by the server that has Remote Desktop Services enabled. When Remote Desktop Services users log on, they can view only their individual client sessions, which are managed by the server and are independent of each other. In addition, Remote Desktop Connection provides 128-bit encryption between client and server.
- Use Internet Protocol security (IPsec) to encrypt confidential data. You can use IPsec to encrypt communication between the NPS and the remote client computer that you are using to administer NPS. To administer the server remotely, you can install the [Remote Server Administration Tools for Windows 10](#) on the client computer. After installation, use the Microsoft Management Console (MMC) to add the NPS snap-in to the console.

IMPORTANT

You can install Remote Server Administration Tools for Windows 10 only on the full release of Windows 10 Professional or Windows 10 Enterprise.

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Getting Started with Network Policy Server

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the topics in this section to learn about Network Policy Server features and capabilities.

NOTE

For additional Network Policy Server documentation, you can use the following library sections.

- [Plan Network Policy Server](#)
- [Deploy Network Policy Server](#)
- [Manage Network Policy Server](#)

This section contains the following topics.

- [Connection Request Processing](#)
- [Network Policies](#)
- [NPS Templates](#)
- [RADIUS Clients](#)

Connection Request Processing

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

You can use this topic to learn about connection request processing in Network Policy Server in Windows Server 2016.

NOTE

In addition to this topic, the following connection request processing documentation is available.

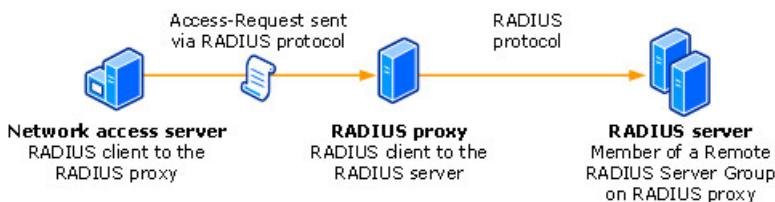
- [Connection Request Policies](#)
- [Realm Names](#)
- [Remote RADIUS Server Groups](#)

You can use connection request processing to specify where the authentication of connection requests is performed - on the local computer or at a remote RADIUS server that is a member of a remote RADIUS server group.

If you want the local server running Network Policy Server (NPS) to perform authentication for connection requests, you can use the default connection request policy without additional configuration. Based on the default policy, NPS authenticates users and computers that have an account in the local domain and in trusted domains.

If you want to forward connection requests to a remote NPS or other RADIUS server, create a remote RADIUS server group and then configure a connection request policy that forwards requests to that remote RADIUS server group. With this configuration, NPS can forward authentication requests to any RADIUS server, and users with accounts in untrusted domains can be authenticated.

The following illustration shows the path of an Access-Request message from a network access server to a RADIUS proxy, and then on to a RADIUS server in a remote RADIUS server group. On the RADIUS proxy, the network access server is configured as a RADIUS client; and on each RADIUS server, the RADIUS proxy is configured as a RADIUS client.



NOTE

The network access servers that you use with NPS can be gateway devices that are compliant with the RADIUS protocol, such as 802.1X wireless access points and authenticating switches, servers running Remote Access that are configured as VPN or dial-up servers, or other RADIUS compatible devices.

If you want NPS to process some authentication requests locally while forwarding other requests to a remote RADIUS server group, configure more than one connection request policy.

To configure a connection request policy that specifies which NPS or RADIUS server group processes authentication requests, see [Connection Request Policies](#).

To specify NPS or other RADIUS servers to which authentication requests are forwarded, see Remote RADIUS Server Groups.

NPS as a RADIUS server connection request processing

When you use NPS as a RADIUS server, RADIUS messages provide authentication, authorization, and accounting for network access connections in the following way:

1. Access servers, such as dial-up network access servers, VPN servers, and wireless access points, receive connection requests from access clients.
2. The access server, configured to use RADIUS as the authentication, authorization, and accounting protocol, creates an Access-Request message and sends it to the NPS.
3. The NPS evaluates the Access-Request message.
4. If required, the NPS sends an Access-Challenge message to the access server. The access server processes the challenge and sends an updated Access-Request to the NPS.
5. The user credentials are checked and the dial-in properties of the user account are obtained by using a secure connection to a domain controller.
6. The connection attempt is authorized with both the dial-in properties of the user account and network policies.
7. If the connection attempt is both authenticated and authorized, the NPS sends an Access-Accept message to the access server. If the connection attempt is either not authenticated or not authorized, the NPS sends an Access-Reject message to the access server.
8. The access server completes the connection process with the access client and sends an Accounting-Request message to the NPS, where the message is logged.
9. The NPS sends an Accounting-Response to the access server.

NOTE

The access server also sends Accounting-Request messages during the time in which the connection is established, when the access client connection is closed, and when the access server is started and stopped.

NPS as a RADIUS proxy connection request processing

When NPS is used as a RADIUS proxy between a RADIUS client and a RADIUS server, RADIUS messages for network access connection attempts are forwarded in the following way:

1. Access servers, such as dial-up network access servers, virtual private network (VPN) servers, and wireless access points, receive connection requests from access clients.
2. The access server, configured to use RADIUS as the authentication, authorization, and accounting protocol, creates an Access-Request message and sends it to the NPS that is being used as the NPS RADIUS proxy.
3. The NPS RADIUS proxy receives the Access-Request message and, based on the locally configured connection request policies, determines where to forward the Access-Request message.
4. The NPS RADIUS proxy forwards the Access-Request message to the appropriate RADIUS server.
5. The RADIUS server evaluates the Access-Request message.
6. If required, the RADIUS server sends an Access-Challenge message to the NPS RADIUS proxy, where it is

forwarded to the access server. The access server processes the challenge with the access client and sends an updated Access-Request to the NPS RADIUS proxy, where it is forwarded to the RADIUS server.

7. The RADIUS server authenticates and authorizes the connection attempt.
8. If the connection attempt is both authenticated and authorized, the RADIUS server sends an Access-Accept message to the NPS RADIUS proxy, where it is forwarded to the access server. If the connection attempt is either not authenticated or not authorized, the RADIUS server sends an Access-Reject message to the NPS RADIUS proxy, where it is forwarded to the access server.
9. The access server completes the connection process with the access client and sends an Accounting-Request message to the NPS RADIUS proxy. The NPS RADIUS proxy logs the accounting data and forwards the message to the RADIUS server.
10. The RADIUS server sends an Accounting-Response to the NPS RADIUS proxy, where it is forwarded to the access server.

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Connection Request Policies

9/1/2018 • 15 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to use NPS connection request policies to configure the NPS as a RADIUS server, a RADIUS proxy, or both.

NOTE

In addition to this topic, the following connection request policy documentation is available.

- [Configure Connection Request Policies](#)
- [Configure Remote RADIUS Server Groups](#)

Connection request policies are sets of conditions and settings that allow network administrators to designate which Remote Authentication Dial-In User Service (RADIUS) servers perform the authentication and authorization of connection requests that the server running Network Policy Server (NPS) receives from RADIUS clients. Connection request policies can be configured to designate which RADIUS servers are used for RADIUS accounting.

You can create connection request policies so that some RADIUS request messages sent from RADIUS clients are processed locally (NPS is used as a RADIUS server) and other types of messages are forwarded to another RADIUS server (NPS is used as a RADIUS proxy).

With connection request policies, you can use NPS as a RADIUS server or as a RADIUS proxy, based on factors such as the following:

- The time of day and day of the week
- The realm name in the connection request
- The type of connection being requested
- The IP address of the RADIUS client

RADIUS Access-Request messages are processed or forwarded by NPS only if the settings of the incoming message match at least one of the connection request policies configured on the NPS.

If the policy settings match and the policy requires that the NPS process the message, NPS acts as a RADIUS server, authenticating and authorizing the connection request. If the policy settings match and the policy requires that the NPS forwards the message, NPS acts as a RADIUS proxy and forwards the connection request to a remote RADIUS server for processing.

If the settings of an incoming RADIUS Access-Request message do not match at least one of the connection request policies, an Access-Reject message is sent to the RADIUS client and the user or computer attempting to connect to the network is denied access.

Configuration examples

The following configuration examples demonstrate how you can use connection request policies.

NPS as a RADIUS server

The default connection request policy is the only configured policy. In this example, NPS is configured as a

RADIUS server and all connection requests are processed by the local NPS. The NPS can authenticate and authorize users whose accounts are in the domain of the NPS domain and in trusted domains.

NPS as a RADIUS proxy

The default connection request policy is deleted, and two new connection request policies are created to forward requests to two different domains. In this example, NPS is configured as a RADIUS proxy. NPS does not process any connection requests on the local server. Instead, it forwards connection requests to NPS or other RADIUS servers that are configured as members of remote RADIUS server groups.

NPS as both RADIUS server and RADIUS proxy

In addition to the default connection request policy, a new connection request policy is created that forwards connection requests to an NPS or other RADIUS server in an untrusted domain. In this example, the proxy policy appears first in the ordered list of policies. If the connection request matches the proxy policy, the connection request is forwarded to the RADIUS server in the remote RADIUS server group. If the connection request does not match the proxy policy but does match the default connection request policy, NPS processes the connection request on the local server. If the connection request does not match either policy, it is discarded.

NPS as RADIUS server with remote accounting servers

In this example, the local NPS is not configured to perform accounting and the default connection request policy is revised so that RADIUS accounting messages are forwarded to an NPS or other RADIUS server in a remote RADIUS server group. Although accounting messages are forwarded, authentication and authorization messages are not forwarded, and the local NPS performs these functions for the local domain and all trusted domains.

NPS with Remote RADIUS to Windows User Mapping

In this example, NPS acts as both a RADIUS server and as a RADIUS proxy for each individual connection request by forwarding the authentication request to a remote RADIUS server while using a local Windows user account for authorization. This configuration is implemented by configuring the Remote RADIUS to Windows User Mapping attribute as a condition of the connection request policy. (In addition, a user account must be created locally that has the same name as the remote user account against which authentication is performed by the remote RADIUS server.)

Connection request policy conditions

Connection request policy conditions are one or more RADIUS attributes that are compared to the attributes of the incoming RADIUS Access-Request message. If there are multiple conditions, then all of the conditions in the connection request message and in the connection request policy must match in order for the policy to be enforced by NPS.

Following are the available condition attributes that you can configure in connection request policies.

Connection Properties attribute group

The Connection Properties attribute group contains the following attributes.

- **Framed Protocol.** Used to designate the type of framing for incoming packets. Examples are Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), Frame Relay, and X.25.
- **Service Type.** Used to designate the type of service being requested. Examples include framed (for example, PPP connections) and login (for example, Telnet connections). For more information about RADIUS service types, see RFC 2865, "Remote Authentication Dial-in User Service (RADIUS)."
- **Tunnel Type.** Used to designate the type of tunnel that is being created by the requesting client. Tunnel types include the Point-to-Point Tunneling Protocol (PPTP) and the Layer Two Tunneling Protocol (L2TP).

Day and Time Restrictions attribute group

The Day and Time Restrictions attribute group contains the Day and Time Restrictions attribute. With this attribute, you can designate the day of the week and the time of day of the connection attempt. The day and time is relative

to the day and time of the NPS.

Gateway attribute group

The Gateway attribute group contains the following attributes.

- **Called Station ID.** Used to designate the phone number of the network access server. This attribute is a character string. You can use pattern-matching syntax to specify area codes.
- **NAS Identifier.** Used to designate the name of the network access server. This attribute is a character string. You can use pattern-matching syntax to specify NAS identifiers.
- **NAS IPv4 Address.** Used to designate the Internet Protocol version 4 (IPv4) address of the network access server (the RADIUS client). This attribute is a character string. You can use pattern-matching syntax to specify IP networks.
- **NAS IPv6 Address.** Used to designate the Internet Protocol version 6 (IPv6) address of the network access server (the RADIUS client). This attribute is a character string. You can use pattern-matching syntax to specify IP networks.
- **NAS Port Type.** Used to designate the type of media used by the access client. Examples are analog phone lines (known as async), Integrated Services Digital Network (ISDN), tunnels or virtual private networks (VPNs), IEEE 802.11 wireless, and Ethernet switches.

Machine Identity attribute group

The Machine Identity attribute group contains the Machine Identity attribute. By using this attribute, you can specify the method with which clients are identified in the policy.

RADIUS Client Properties attribute group

The RADIUS Client Properties attribute group contains the following attributes.

- **Calling Station ID.** Used to designate the phone number used by the caller (the access client). This attribute is a character string. You can use pattern-matching syntax to specify area codes. In 802.1x authentications the MAC Address is typically populated and can be matched from the client. This field is typically used for Mac Address Bypass scenarios when the connection request policy is configured for 'Accept users without validating credentials'.
- **Client Friendly Name.** Used to designate the name of the RADIUS client computer that is requesting authentication. This attribute is a character string. You can use pattern-matching syntax to specify client names.
- **Client IPv4 Address.** Used to designate the IPv4 address of the network access server (the RADIUS client). This attribute is a character string. You can use pattern-matching syntax to specify IP networks.
- **Client IPv6 Address.** Used to designate the IPv6 address of the network access server (the RADIUS client). This attribute is a character string. You can use pattern-matching syntax to specify IP networks.
- **Client Vendor.** Used to designate the vendor of the network access server that is requesting authentication. A computer running the Routing and Remote Access service is the Microsoft NAS manufacturer. You can use this attribute to configure separate policies for different NAS manufacturers. This attribute is a character string. You can use pattern-matching syntax.

User Name attribute group

The User Name attribute group contains the User Name attribute. By using this attribute, you can designate the user name, or a portion of the user name, that must match the user name supplied by the access client in the RADIUS message. This attribute is a character string that typically contains a realm name and a user account name. You can use pattern-matching syntax to specify user names.

Connection request policy settings

Connection request policy settings are a set of properties that are applied to an incoming RADIUS message. Settings consist of the following groups of properties.

- Authentication

- Accounting
- Attribute manipulation
- Forwarding request
- Advanced

The following sections provide additional detail about these settings.

Authentication

By using this setting, you can override the authentication settings that are configured in all network policies and you can designate the authentication methods and types that are required to connect to your network.

IMPORTANT

If you configure an authentication method in connection request policy that is less secure than the authentication method you configure in network policy, the more secure authentication method that you configure in network policy is overridden. For example, if you have one network policy that requires the use of Protected Extensible Authentication Protocol-Microsoft Challenge Handshake Authentication Protocol version 2 (PEAP-MS-CHAP v2), which is a password-based authentication method for secure wireless, and you also configure a connection request policy to allow unauthenticated access, the result is that no clients are required to authenticate by using PEAP-MS-CHAP v2. In this example, all clients connecting to your network are granted unauthenticated access.

Accounting

By using this setting, you can configure connection request policy to forward accounting information to an NPS or other RADIUS server in a remote RADIUS server group so that the remote RADIUS server group performs accounting.

NOTE

If you have multiple RADIUS servers and you want accounting information for all servers stored in one central RADIUS accounting database, you can use the connection request policy accounting setting in a policy on each RADIUS server to forward accounting data from all of the servers to one NPS or other RADIUS server that is designated as an accounting server.

Connection request policy accounting settings function independent of the accounting configuration of the local NPS. In other words, if you configure the local NPS to log RADIUS accounting information to a local file or to a Microsoft SQL Server database, it will do so regardless of whether you configure a connection request policy to forward accounting messages to a remote RADIUS server group.

If you want accounting information logged remotely but not locally, you must configure the local NPS to not perform accounting, while also configuring accounting in a connection request policy to forward accounting data to a remote RADIUS server group.

Attribute manipulation

You can configure a set of find-and-replace rules that manipulate the text strings of one of the following attributes.

- User Name
- Called Station ID
- Calling Station ID

Find-and-replace rule processing occurs for one of the preceding attributes before the RADIUS message is subject to authentication and accounting settings. Attribute manipulation rules apply only to a single attribute. You cannot configure attribute manipulation rules for each attribute. In addition, the list of attributes that you can manipulate is a static list; you cannot add to the list of attributes available for manipulation.

NOTE

If you are using the MS-CHAP v2 authentication protocol, you cannot manipulate the User Name attribute if the connection request policy is used to forward the RADIUS message. The only exception occurs when a backslash () character is used and the manipulation only affects the information to the left of it. A backslash character is typically used to indicate a domain name (the information to the left of the backslash character) and a user account name within the domain (the information to the right of the backslash character). In this case, only attribute manipulation rules that modify or replace the domain name are allowed.

For examples of how to manipulate the realm name in the User Name attribute, see the section "Examples for manipulation of the realm name in the User Name attribute" in the topic [Use Regular Expressions in NPS](#).

Forwarding request

You can set the following forwarding request options that are used for RADIUS Access-Request messages:

- **Authenticate requests on this server.** By using this setting, NPS uses a Windows NT 4.0 domain, Active Directory, or the local Security Accounts Manager (SAM) user accounts database to authenticate the connection request. This setting also specifies that the matching network policy configured in NPS, along with the dial-in properties of the user account, are used by NPS to authorize the connection request. In this case, the NPS is configured to perform as a RADIUS server.
- **Forward requests to the following remote RADIUS server group.** By using this setting, NPS forwards connection requests to the remote RADIUS server group that you specify. If the NPS receives a valid Access-Accept message that corresponds to the Access-Request message, the connection attempt is considered authenticated and authorized. In this case, the NPS acts as a RADIUS proxy.
- **Accept users without validating credentials.** By using this setting, NPS does not verify the identity of the user attempting to connect to the network and NPS does not attempt to verify that the user or computer has the right to connect to the network. When NPS is configured to allow unauthenticated access and it receives a connection request, NPS immediately sends an Access-Accept message to the RADIUS client and the user or computer is granted network access. This setting is used for some types of compulsory tunneling where the access client is tunneled before user credentials are authenticated.

NOTE

This authentication option cannot be used when the authentication protocol of the access client is MS-CHAP v2 or Extensible Authentication Protocol-Transport Layer Security (EAP-TLS), both of which provide mutual authentication. In mutual authentication, the access client proves that it is a valid access client to the authenticating server (the NPS), and the authenticating server proves that it is a valid authenticating server to the access client. When this authentication option is used, the Access-Accept message is returned. However, the authenticating server does not provide validation to the access client, and mutual authentication fails.

For examples of how to use regular expressions to create routing rules that forward RADIUS messages with a specified realm name to a remote RADIUS server group, see the section "Example for RADIUS message forwarding by a proxy server" in the topic [Use Regular Expressions in NPS](#).

Advanced

You can set advanced properties to specify the series of RADIUS attributes that are:

- Added to the RADIUS response message when the NPS is being used as a RADIUS authentication or accounting server. When there are attributes specified on both a network policy and the connection request policy, the attributes that are sent in the RADIUS response message are the combination of the two sets of attributes.
- Added to the RADIUS message when the NPS is being used as a RADIUS authentication or accounting proxy.

If the attribute already exists in the message that is forwarded, it is replaced with the value of the attribute specified in the connection request policy.

In addition, some attributes that are available for configuration on the connection request policy **Settings** tab in the **Advanced** category provide specialized functionality. For example, you can configure the **Remote RADIUS to Windows User Mapping** attribute when you want to split the authentication and authorization of a connection request between two user accounts databases.

The **Remote RADIUS to Windows User Mapping** attribute specifies that Windows authorization occurs for users who are authenticated by a remote RADIUS server. In other words, a remote RADIUS server performs authentication against a user account in a remote user accounts database, but the local NPS authorizes the connection request against a user account in a local user accounts database. This is useful when you want to allow visitors access to your network.

For example, visitors from partner organizations can be authenticated by their own partner organization RADIUS server, and then use a Windows user account at your organization to access a guest local area network (LAN) on your network.

Other attributes that provide specialized functionality are:

- **MS-Quarantine-IPFilter and MS-Quarantine-Session-Timeout.** These attributes are used when you deploy Network Access Quarantine Control (NAQC) with your Routing and Remote Access VPN deployment.
- **Passport-User-Mapping-UPN-Suffix.** This attribute allows you to authenticate connection requests with Windows Live™ ID user account credentials.
- **Tunnel-Tag.** This attribute designates the VLAN ID number to which the connection should be assigned by the NAS when you deploy virtual local area networks (VLANs).

Default connection request policy

A default connection request policy is created when you install NPS. This policy has the following configuration.

- Authentication is not configured.
- Accounting is not configured to forward accounting information to a remote RADIUS server group.
- Attribute is not configured with attribute manipulation rules that forward connection requests to remote RADIUS server groups.
- Forwarding Request is configured so that connection requests are authenticated and authorized on the local NPS.
- Advanced attributes are not configured.

The default connection request policy uses NPS as a RADIUS server. To configure a server running NPS to act as a RADIUS proxy, you must also configure a remote RADIUS server group. You can create a new remote RADIUS server group while you are creating a new connection request policy by using the New Connection Request Policy Wizard. You can either delete the default connection request policy or verify that the default connection request policy is the last policy processed by NPS by placing it last in the ordered list of policies.

NOTE

If NPS and the Remote Access service are installed on the same computer, and the Remote Access service is configured for Windows authentication and accounting, it is possible for Remote Access authentication and accounting requests to be forwarded to a RADIUS server. This can occur when Remote Access authentication and accounting requests match a connection request policy that is configured to forward them to a remote RADIUS server group.

Realm Names

9/1/2018 • 5 minutes to read • [Edit Online](#)

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

You can use this topic for an overview of using realm names in Network Policy Server connection request processing.

The User-Name RADIUS attribute is a character string that typically contains a user account location and a user account name. The user account location is also called the realm or realm name, and is synonymous with the concept of domain, including DNS domains, Active Directory® domains, and Windows NT 4.0 domains. For example, if a user account is located in the user accounts database for a domain named example.com, then example.com is the realm name.

In another example, if the User-Name RADIUS attribute contains the user name user1@example.com, user1 is the user account name and example.com is the realm name. Realm names can be presented in the user name as a prefix or as a suffix:

- **Example\user1.** In this example, the realm name **Example** is a prefix; and it is also the name of an Active Directory® Domain Services (AD DS) domain.
- **user1@example.com.** In this example, the realm name **example.com** is a suffix; and it is either a DNS domain name or the name of an AD DS domain.

You can use realm names configured in connection request policies while designing and deploying your RADIUS infrastructure to ensure that connection requests are routed from RADIUS clients, also called network access servers, to RADIUS servers that can authenticate and authorize the connection request.

When NPS is configured as a RADIUS server with the default connection request policy, NPS processes connection requests for the domain in which the NPS is a member and for trusted domains.

To configure NPS to act as a RADIUS proxy and forward connection requests to untrusted domains, you must create a new connection request policy. In the new connection request policy, you must configure the User Name attribute with the realm name that will be contained in the User-Name attribute of connection requests that you want to forward. You must also configure the connection request policy with a remote RADIUS server group. The connection request policy allows NPS to calculate which connection requests to forward to the remote RADIUS server group based on the realm portion of the User-Name attribute.

Acquiring the realm name

The realm name portion of the user name is provided when the user types password-based credentials during a connection attempt or when a Connection Manager (CM) profile on the user's computer is configured to provide the realm name automatically.

You can designate that users of your network provide their realm name when typing their credentials during network connection attempts.

For example, you can require users to type their user name, including the user account name and the realm name, in **User name** in the **Connect** dialog box when making a dial-up or virtual private network (VPN) connection.

In addition, if you create a custom dialing package with the Connection Manager Administration Kit (CMAK), you can assist users by adding the realm name automatically to the user account name in CM profiles that are installed on users' computers. For example, you can specify a realm name and user name syntax in the CM profile so that

the user only has to specify the user account name when typing credentials. In this circumstance, the user does not need to know or remember the domain where their user account is located.

During the authentication process, after users type their password-based credentials, the user name is passed from the access client to the network access server. The network access server constructs a connection request and includes the realm name within the User-Name RADIUS attribute in the Access-Request message that is sent to the RADIUS proxy or server.

If the RADIUS server is an NPS, the Access-Request message is evaluated against the set of configured connection request policies. Conditions on the connection request policy can include the specification of the contents of the User-Name attribute.

You can configure a set of connection request policies that are specific to the realm name within the User-Name attribute of incoming messages. This allows you to create routing rules that forward RADIUS messages with a specific realm name to a specific set of RADIUS servers when NPS is used as a RADIUS proxy.

Attribute manipulation rules

Before the RADIUS message is either processed locally (when NPS is being used as a RADIUS server) or forwarded to another RADIUS server (when NPS is being used as a RADIUS proxy), the User-Name attribute in the message can be modified by attribute manipulation rules. You can configure attribute manipulation rules for the User-Name attribute by selecting **User name** on the **Conditions** tab in the properties of a connection request policy. NPS attribute manipulation rules use regular expression syntax.

You can configure attribute manipulation rules for the User-Name attribute to change the following:

- Remove the realm name from the user name (also known as realm stripping). For example, the user name user1@example.com is changed to user1.
- Change the realm name but not its syntax. For example, the user name user1@example.com is changed to user1@wcoast.example.com.
- Change the syntax of the realm name. For example, the user name example\user1 is changed to user1@example.com.

After the User-Name attribute is modified according to the attribute manipulation rules that you configure, additional settings of the first matching connection request policy are used to determine whether:

- The NPS processes the Access-Request message locally (when NPS is being used as a RADIUS server).
- The NPS forwards the message to another RADIUS server (when NPS is being used as a RADIUS proxy).

Configuring the the NPS-supplied domain name

When the user name does not contain a domain name, NPS supplies one. By default, the NPS-supplied domain name is the domain of which the NPS is a member. You can specify the NPS-supplied domain name through the following registry setting:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\RasMan\PPP\ControlProtocols\BuiltIn\DefaultDomain
```

Caution

Incorrectly editing the registry can severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.

Some non-Microsoft network access servers delete or modify the domain name as specified by the user. As the result, the network access request is authenticated against the default domain, which might not be the domain for the user's account. To resolve this problem, configure your RADIUS servers to change the user name into the

correct format with the accurate domain name.

Remote RADIUS Server Groups

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

When you configure Network Policy Server (NPS) as a Remote Authentication Dial-In User Service (RADIUS) proxy, you use NPS to forward connection requests to RADIUS servers that are capable of processing the connection requests because they can perform authentication and authorization in the domain where the user or computer account is located. For example, if you want to forward connection requests to one or more RADIUS servers in untrusted domains, you can configure NPS as a RADIUS proxy to forward the requests to the remote RADIUS servers in the untrusted domain.

NOTE

Remote RADIUS server groups are unrelated to and separate from Windows groups.

To configure NPS as a RADIUS proxy, you must create a connection request policy that contains all of the information required for NPS to evaluate which messages to forward and where to send the messages.

When you configure a remote RADIUS server group in NPS and you configure a connection request policy with the group, you are designating the location where NPS is to forward connection requests.

Configuring RADIUS servers for a group

A remote RADIUS server group is a named group that contains one or more RADIUS servers. If you configure more than one server, you can specify load balancing settings to either determine the order in which the servers are used by the proxy or to distribute the flow of RADIUS messages across all servers in the group to prevent overloading one or more servers with too many connection requests.

Each server in the group has the following settings.

- **Name or address.** Each group member must have a unique name within the group. The name can be an IP address or a name that can be resolved to its IP address.
- **Authentication and accounting.** You can forward authentication requests, accounting requests, or both to each remote RADIUS server group member.
- **Load balancing.** A priority setting is used to indicate which member of the group is the primary server (the priority is set to 1). For group members that have the same priority, a weight setting is used to calculate how often RADIUS messages are sent to each server. You can use additional settings to configure the way in which the NPS detects when a group member first becomes unavailable and when it becomes available after it has been determined to be unavailable.

After you have configured a Remote RADIUS Server Group, you can specify the group in the authentication and accounting settings of a connection request policy. Because of this, you can configure a remote RADIUS server group first. Next, you can configure the connection request policy to use the newly configured remote RADIUS server group. Alternatively, you can use the New Connection Request Policy Wizard to create a new remote RADIUS server group while you are creating the connection request policy.

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Network Policies

9/1/2018 • 3 minutes to read • [Edit Online](#)

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

You can use this topic for an overview of network policies in NPS.

NOTE

In addition to this topic, the following network policy documentation is available.

- [Access Permission](#)
- [Configure Network Policies](#)

Network policies are sets of conditions, constraints, and settings that allow you to designate who is authorized to connect to the network and the circumstances under which they can or cannot connect.

When processing connection requests as a Remote Authentication Dial-In User Service (RADIUS) server, NPS performs both authentication and authorization for the connection request. During the authentication process, NPS verifies the identity of the user or computer that is connecting to the network. During the authorization process, NPS determines whether the user or computer is allowed to access the network.

To make these determinations, NPS uses network policies that are configured in the NPS console. NPS also examines the dial-in properties of the user account in Active Directory® Domain Services (AD DS) to perform authorization.

Network Policies - An Ordered Set of Rules

Network policies can be viewed as rules. Each rule has a set of conditions and settings. NPS compares the conditions of the rule to the properties of connection requests. If a match occurs between the rule and the connection request, the settings defined in the rule are applied to the connection.

When multiple network policies are configured in NPS, they are an ordered set of rules. NPS checks each connection request against the first rule in the list, then the second, and so on, until a match is found.

Each network policy has a **Policy State** setting that allows you to enable or disable the policy. When you disable a network policy, NPS does not evaluate the policy when authorizing connection requests.

NOTE

If you want NPS to evaluate a network policy when performing authorization for connection requests, you must configure the **Policy State** setting by selecting the Policy enabled check box.

Network policy properties

There are four categories of properties for each network policy:

Overview

These properties allow you to specify whether the policy is enabled, whether the policy grants or denies access, and whether a specific network connection method, or type of network access server (NAS), is required for

connection requests. Overview properties also allow you to specify whether the dial-in properties of user accounts in AD DS are ignored. If you select this option, only the settings in the network policy are used by NPS to determine whether the connection is authorized.

Conditions

These properties allow you to specify the conditions that the connection request must have in order to match the network policy; if the conditions configured in the policy match the connection request, NPS applies the settings designated in the network policy to the connection. For example, if you specify the NAS IPv4 address as a condition of the network policy and NPS receives a connection request from a NAS that has the specified IP address, the condition in the policy matches the connection request.

Constraints

Constraints are additional parameters of the network policy that are required to match the connection request. If a constraint is not matched by the connection request, NPS automatically rejects the request. Unlike the NPS response to unmatched conditions in the network policy, if a constraint is not matched, NPS denies the connection request without evaluating additional network policies.

Settings

These properties allow you to specify the settings that NPS applies to the connection request if all of the network policy conditions for the policy are matched.

When you add a new network policy by using the NPS console, you must use the New Network Policy Wizard. After you have created a network policy by using the wizard, you can customize the policy by double-clicking the policy in the NPS console to obtain the policy properties.

For examples of pattern-matching syntax to specify network policy attributes, see [Use Regular Expressions in NPS](#).

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Access Permission

9/1/2018 • 3 minutes to read • [Edit Online](#)

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

Access permission is configured on the **Overview** tab of each network policy in Network Policy Server (NPS).

This setting allows you to configure the policy to either grant or deny access to users if the conditions and constraints of the network policy are matched by the connection request.

Access permission settings have the following effect:

- **Grant access.** Access is granted if the connection request matches the conditions and constraints that are configured in the policy.
- **Deny access.** Access is denied if the connection request matches the conditions and constraints that are configured in the policy.

Access permission is also granted or denied based on your configuration of the dial-in properties of each user account.

NOTE

User accounts and their properties, such as dial-in properties, are configured in either the Active Directory Users and Computers or the Local Users and Groups Microsoft Management Console (MMC) snap-in, depending on whether you have Active Directory® Domain Services (AD DS) installed.

The user account setting **Network Access Permission**, which is configured on the dial-in properties of user accounts, overrides the network policy access permission setting. When network access permission on a user account is set to the **Control access through NPS Network Policy** option, the network policy access permission setting determines whether the user is granted or denied access.

NOTE

In Windows Server 2016, the default value of **Network Access Permission** in AD DS user account dial-in properties is **Control access through NPS Network Policy**.

When NPS evaluates connection requests against configured network policies, it performs the following actions:

- If the conditions of the first policy are not matched, NPS evaluates the next policy, and continues this process until either a match is found or all policies have been evaluated for a match.
- If the conditions and constraints of a policy are matched, NPS either grants or denies access, depending on the value of the Access Permission setting in the policy.
- If the conditions of a policy match but the constraints in the policy do not match, NPS rejects the connection request.
- If the conditions of all policies do not match, NPS rejects the connection request.

Ignore user account dial-in properties

You can configure NPS network policy to ignore the dial-in properties of user accounts by selecting or clearing the **Ignore user account dial-in properties** check box on the **Overview** tab of a network policy.

Normally when NPS performs authorization of a connection request, it checks the dial-in properties of the user account, where the network access permission setting value can affect whether the user is authorized to connect to the network. When you configure NPS to ignore the dial-in properties of user accounts during authorization, network policy settings determine whether the user is granted access to the network.

The dial-in properties of user accounts contain the following:

- Network access permission
- Caller-ID
- Callback options
- Static IP address
- Static routes

To support multiple types of connections for which NPS provides authentication and authorization, it might be necessary to disable the processing of user account dial-in properties. This can be done to support scenarios in which specific dial-in properties are not required.

For example, the caller-ID, callback, static IP address, and static routes properties are designed for a client that is dialing into a network access server (NAS), not for clients that are connecting to wireless access points. A wireless access point that receives these settings in a RADIUS message from NPS might not be able to process them, which can cause the wireless client to be disconnected.

When NPS provides authentication and authorization for users who are both dialing in and accessing your organization network through wireless access points, you must configure the dial-in properties to support either dial-in connections (by setting dial-in properties) or wireless connections (by not setting dial-in properties).

You can use NPS to enable dial-in properties processing for the user account in some scenarios (such as dial-in) and to disable dial-in properties processing in other scenarios (such as 802.1X wireless and authenticating switch).

You can also use **Ignore user account dial-in properties** to manage network access control through groups and the access permission setting on the network policy. When you select the **Ignore user account dial-in properties** check box, network access permission on the user account is ignored.

The only disadvantage to this configuration is that you cannot use the additional user account dial-in properties of caller-ID, callback, static IP address, and static routes.

For more information about NPS, see [Network Policy Server \(NPS\)](#).

NPS Templates

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

Network Policy Server (NPS) templates allow you to create configuration elements, such as Remote Authentication Dial-In User Service (RADIUS) clients or shared secrets, that you can reuse on the local NPS and export for use on other NPSs.

NPS templates are designed to reduce the amount of time and cost that it takes to configure NPS on one or more servers. The following NPS template types are available for configuration in Templates Management:

- Shared Secrets
- RADIUS Clients
- Remote RADIUS Servers
- IP Filters
- Remediation Server Groups

Configuring a template is different than configuring the NPS directly. Creating a template does not affect the NPS's functionality. It is only when you select the template in the appropriate location in the NPS console that the template affects the NPS functionality.

For example, if you configure a RADIUS client in the NPS console under RADIUS Clients and Servers, you have altered the NPS configuration and taken one step in configuring NPS to communicate with one of your network access servers (NAS's). (The next step would be to configure the NAS to communicate with NPS.) However, if you configure a new RADIUS Clients template in the NPS console under **Templates Management** rather than creating a new RADIUS client under **RADIUS Clients and Servers**, you have created a template, but you have not altered the NPS functionality yet. To alter the NPS functionality, you must select the template from the correct location in the NPS console.

Creating templates

To create a template, open the NPS console, right-click a template type, such as **IP Filters**, and then click **New**. A new template properties dialog box opens that allows you to configure your template.

Using templates locally

You can use a template that you've created in **Templates Management** by navigating to a location in the NPS console where the template can be applied. For example, if you create a new Shared Secrets template that you want to apply to a RADIUS client configuration, in **RADIUS Clients and Servers** and **RADIUS Clients**, open the RADIUS client properties. In **Select an existing Shared Secrets template**, select the template you previously created from the list of available templates.

For more information about NPS, see [Network Policy Server \(NPS\)](#).

RADIUS Clients

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

A network access server (NAS) is a device that provides some level of access to a larger network. A NAS using a RADIUS infrastructure is also a RADIUS client, sending connection requests and accounting messages to a RADIUS server for authentication, authorization, and accounting.

NOTE

Client computers, such as laptop computers and other computers running client operating systems, are not RADIUS clients. RADIUS clients are network access servers - such as wireless access points, 802.1X authenticating switches, virtual private network (VPN) servers, and dial-up servers - because they use the RADIUS protocol to communicate with RADIUS servers such as Network Policy Server (NPS) servers.

To deploy NPS as a RADIUS server or a RADIUS proxy, you must configure RADIUS clients in NPS.

RADIUS client examples

Examples of network access servers are:

- Network access servers that provide remote access connectivity to an organization network or the Internet. An example is a computer running the Windows Server 2016 operating system and the Remote Access service that provides either traditional dial-up or virtual private network (VPN) remote access services to an organization intranet.
- Wireless access points that provide physical layer access to an organization network using wireless-based transmission and reception technologies.
- Switches that provide physical layer access to an organization's network, using traditional LAN technologies, such as Ethernet.
- RADIUS proxies that forward connection requests to RADIUS servers that are members of a remote RADIUS server group that is configured on the RADIUS proxy.

RADIUS Access-Request messages

RADIUS clients either create RADIUS Access-Request messages and forward them to a RADIUS proxy or RADIUS server, or they forward Access-Request messages to a RADIUS server that they have received from another RADIUS client but have not created themselves.

RADIUS clients do not process Access-Request messages by performing authentication, authorization, and accounting. Only RADIUS servers perform these functions.

NPS, however, can be configured as both a RADIUS proxy and a RADIUS server simultaneously, so that it processes some Access-Request messages and forwards other messages.

NPS as a RADIUS client

NPS acts as a RADIUS client when you configure it as a RADIUS proxy to forward Access-Request messages to other RADIUS servers for processing. When you use NPS as a RADIUS proxy, the following general configuration steps are required:

1. Network access servers, such as wireless access points and VPN servers, are configured with the IP address of the NPS proxy as the designated RADIUS server or authenticating server. This allows the network access servers, which create Access-Request messages based on information they receive from access clients, to forward messages to the NPS proxy.
2. The NPS proxy is configured by adding each network access server as a RADIUS client. This configuration step allows the NPS proxy to receive messages from the network access servers and to communicate with them throughout authentication. In addition, connection request policies on the NPS proxy are configured to specify which Access-Request messages to forward to one or more RADIUS servers. These policies are also configured with a remote RADIUS server group, which tells NPS where to send the messages it receives from the network access servers.
3. The NPS or other RADIUS servers that are members of the remote RADIUS server group on the NPS proxy are configured to receive messages from the NPS proxy. This is accomplished by configuring the NPS proxy as a RADIUS client.

RADIUS client properties

When you add a RADIUS client to the NPS configuration through the NPS console or through the use of the netsh commands for NPS or Windows PowerShell commands, you are configuring NPS to receive RADIUS Access-Request messages from either a network access server or a RADIUS proxy.

When you configure a RADIUS client in NPS, you can designate the following properties:

Client name

A friendly name for the RADIUS client, which makes it easier to identify when using the NPS snap-in or netsh commands for NPS.

IP address

The Internet Protocol version 4 (IPv4) address or the Domain Name System (DNS) name of the RADIUS client.

Client-Vendor

The vendor of the RADIUS client. Otherwise, you can use the RADIUS standard value for Client-Vendor.

Shared secret

A text string that is used as a password between RADIUS clients, RADIUS servers, and RADIUS proxies. When the Message Authenticator attribute is used, the shared secret is also used as the key to encrypt RADIUS messages. This string must be configured on the RADIUS client and in the NPS snap-in.

Message Authenticator attribute

Described in RFC 2869, "RADIUS Extensions," a Message Digest 5 (MD5) hash of the entire RADIUS message. If the RADIUS Message Authenticator attribute is present, it is verified. If it fails verification, the RADIUS message is discarded. If the client settings require the Message Authenticator attribute and it is not present, the RADIUS message is discarded. Use of the Message Authenticator attribute is recommended.

NOTE

The Message Authenticator attribute is required and enabled by default when you use Extensible Authentication Protocol (EAP) authentication.

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Plan Network Policy Server

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

This topic provides links to information about planning NPS and proxy deployments.

NOTE

For additional Network Policy Server documentation, you can use the following library sections.

- [Getting Started with Network Policy Server](#)
- [Deploy Network Policy Server](#)
- [Manage Network Policy Server](#)

This section includes the following topics.

- [Plan NPS as a RADIUS server](#)
- [Plan NPS as a RADIUS proxy](#)

Plan NPS as a RADIUS server

9/1/2018 • 16 minutes to read • [Edit Online](#)

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

When you deploy Network Policy Server (NPS) as a Remote Authentication Dial-In User Service (RADIUS) server, NPS performs authentication, authorization, and accounting for connection requests for the local domain and for domains that trust the local domain. You can use these planning guidelines to simplify your RADIUS deployment.

These planning guidelines do not include circumstances in which you want to deploy NPS as a RADIUS proxy. When you deploy NPS as a RADIUS proxy, NPS forwards connection requests to a server running NPS or other RADIUS servers in remote domains, untrusted domains, or both.

Before you deploy NPS as a RADIUS server on your network, use the following guidelines to plan your deployment.

- Plan NPS configuration.
- Plan RADIUS clients.
- Plan the use of authentication methods.
- Plan network policies.
- Plan NPS accounting.

Plan NPS configuration

You must decide in which domain the NPS is a member. For multiple-domain environments, an NPS can authenticate credentials for user accounts in the domain of which it is a member and for all domains that trust the local domain of the NPS. To allow the NPS to read the dial-in properties of user accounts during the authorization process, you must add the computer account of the NPS to the RAS and NPSs group for each domain.

After you have determined the domain membership of the NPS, the server must be configured to communicate with RADIUS clients, also called network access servers, by using the RADIUS protocol. In addition, you can configure the types of events that NPS records in the event log and you can enter a description for the server.

Key steps

During the planning for NPS configuration, you can use the following steps.

- Determine the RADIUS ports that the NPS uses to receive RADIUS messages from RADIUS clients. The default ports are UDP ports 1812 and 1645 for RADIUS authentication messages and ports 1813 and 1646 for RADIUS accounting messages.
- If the NPS is configured with multiple network adapters, determine the adapters over which you want RADIUS traffic to be allowed.
- Determine the types of events that you want NPS to record in the Event Log. You can log rejected authentication requests, successful authentication requests, or both types of requests.
- Determine whether you are deploying more than one NPS. To provide fault tolerance for RADIUS-based authentication and accounting, use at least two NPSs. One NPS is used as the primary RADIUS server and the other is used as a backup. Each RADIUS client is then configured on both NPSs. If the primary NPS becomes unavailable, RADIUS clients then send Access-Request messages to the alternate NPS.

- Plan the script used to copy one NPS configuration to other NPSs to save on administrative overhead and to prevent the incorrect configuration of a server. NPS provides the Netsh commands that allow you to copy all or part of an NPS configuration for import onto another NPS. You can run the commands manually at the Netsh prompt. However, if you save your command sequence as a script, you can run the script at a later date if you decide to change your server configurations.

Plan RADIUS clients

RADIUS clients are network access servers, such as wireless access points, virtual private network (VPN) servers, 802.1X-capable switches, and dial-up servers. RADIUS proxies, which forward connection request messages to RADIUS servers, are also RADIUS clients. NPS supports all network access servers and RADIUS proxies that comply with the RADIUS protocol as described in RFC 2865, "Remote Authentication Dial-in User Service (RADIUS)," and RFC 2866, "RADIUS Accounting."

IMPORTANT

Access clients, such as client computers, are not RADIUS clients. Only network access servers and proxy servers that support the RADIUS protocol are RADIUS clients.

In addition, both wireless access points and switches must be capable of 802.1X authentication. If you want to deploy Extensible Authentication Protocol (EAP) or Protected Extensible Authentication Protocol (PEAP), access points and switches must support the use of EAP.

To test basic interoperability for PPP connections for wireless access points, configure the access point and the access client to use Password Authentication Protocol (PAP). Use additional PPP-based authentication protocols, such as PEAP, until you have tested the ones that you intend to use for network access.

Key steps

During the planning for RADIUS clients, you can use the following steps.

- Document the vendor-specific attributes (VSAs) you must configure in NPS. If your network access servers require VSAs, log the VSA information for later use when you configure your network policies in NPS.
- Document the IP addresses of RADIUS clients and your NPS to simplify the configuration of all devices. When you deploy your RADIUS clients, you must configure them to use the RADIUS protocol, with the NPS IP address entered as the authenticating server. And when you configure NPS to communicate with your RADIUS clients, you must enter the RADIUS client IP addresses into the NPS snap-in.
- Create shared secrets for configuration on the RADIUS clients and in the NPS snap-in. You must configure RADIUS clients with a shared secret, or password, that you will also enter into the NPS snap-in while configuring RADIUS clients in NPS.

Plan the use of authentication methods

NPS supports both password-based and certificate-based authentication methods. However, not all network access servers support the same authentication methods. In some cases, you might want to deploy a different authentication method based on the type of network access.

For example, you might want to deploy both wireless and VPN access for your organization, but use a different authentication method for each type of access: EAP-TLS for VPN connections, due to the strong security that EAP with Transport Layer Security (EAP-TLS) provides, and PEAP-MS-CHAP v2 for 802.1X wireless connections.

PEAP with Microsoft Challenge Handshake Authentication Protocol version 2 (PEAP-MS-CHAP v2) provides a feature named fast reconnect that is specifically designed for use with portable computers and other wireless devices. Fast reconnect enables wireless clients to move between wireless access points on the same network

without being reauthenticated each time they associate with a new access point. This provides a better experience for wireless users and allows them to move between access points without having to retype their credentials. Because of fast reconnect and the security that PEAP-MS-CHAP v2 provides, PEAP-MS-CHAP v2 is a logical choice as an authentication method for wireless connections.

For VPN connections, EAP-TLS is a certificate-based authentication method that provides strong security that protects network traffic even as it is transmitted across the Internet from home or mobile computers to your organization VPN servers.

Certificate-based authentication methods

Certificate-based authentication methods have the advantage of providing strong security; and they have the disadvantage of being more difficult to deploy than password-based authentication methods.

Both PEAP-MS-CHAP v2 and EAP-TLS are certificate-based authentication methods, but there are many differences between them and the way in which they are deployed.

EAP-TLS

EAP-TLS uses certificates for both client and server authentication, and requires that you deploy a public key infrastructure (PKI) in your organization. Deploying a PKI can be complex, and requires a planning phase that is independent of planning for the use of NPS as a RADIUS server.

With EAP-TLS, the NPS enrolls a server certificate from a certification authority (CA), and the certificate is saved on the local computer in the certificate store. During the authentication process, server authentication occurs when the NPS sends its server certificate to the access client to prove its identity to the access client. The access client examines various certificate properties to determine whether the certificate is valid and is appropriate for use during server authentication. If the server certificate meets the minimum server certificate requirements and is issued by a CA that the access client trusts, the NPS is successfully authenticated by the client.

Similarly, client authentication occurs during the authentication process when the client sends its client certificate to the NPS to prove its identity to the NPS. The NPS examines the certificate, and if the client certificate meets the minimum client certificate requirements and is issued by a CA that the NPS trusts, the access client is successfully authenticated by the NPS.

Although it is required that the server certificate is stored in the certificate store on the NPS, the client or user certificate can be stored in either the certificate store on the client or on a smart card.

For this authentication process to succeed, it is required that all computers have your organization's CA certificate in the Trusted Root Certification Authorities certificate store for the Local Computer and the Current User.

PEAP-MS-CHAP v2

PEAP-MS-CHAP v2 uses a certificate for server authentication and password-based credentials for user authentication. Because certificates are used only for server authentication, you are not required to deploy a PKI in order to use PEAP-MS-CHAP v2. When you deploy PEAP-MS-CHAP v2, you can obtain a server certificate for the NPS in one of the following two ways:

- You can install Active Directory Certificate Services (AD CS), and then autoenroll certificates to NPSs. If you use this method, you must also enroll the CA certificate to client computers connecting to your network so that they trust the certificate issued to the NPS.
- You can purchase a server certificate from a public CA such as VeriSign. If you use this method, make sure that you select a CA that is already trusted by client computers. To determine whether client computers trust a CA, open the Certificates Microsoft Management Console (MMC) snap-in on a client computer, and then view the Trusted Root Certification Authorities store for the Local Computer and for the Current User. If there is a certificate from the CA in these certificate stores, the client computer trusts the CA and will therefore trust any certificate issued by the CA.

During the authentication process with PEAP-MS-CHAP v2, server authentication occurs when the NPS sends its

server certificate to the client computer. The access client examines various certificate properties to determine whether the certificate is valid and is appropriate for use during server authentication. If the server certificate meets the minimum server certificate requirements and is issued by a CA that the access client trusts, the NPS is successfully authenticated by the client.

User authentication occurs when a user attempting to connect to the network types password-based credentials and tries to log on. NPS receives the credentials and performs authentication and authorization. If the user is authenticated and authorized successfully, and if the client computer successfully authenticated the NPS, the connection request is granted.

Key steps

During the planning for the use of authentication methods, you can use the following steps.

- Identify the types of network access you plan to offer, such as wireless, VPN, 802.1X-capable switch, and dial-up access.
- Determine the authentication method or methods that you want to use for each type of access. It is recommended that you use the certificate-based authentication methods that provide strong security; however, it might not be practical for you to deploy a PKI, so other authentication methods might provide a better balance of what you need for your network.
- If you are deploying EAP-TLS, plan your PKI deployment. This includes planning the certificate templates you are going to use for server certificates and client computer certificates. It also includes determining how to enroll certificates to domain member and non-domain member computers, and determining whether you want to use smart cards.
- If you are deploying PEAP-MS-CHAP v2, determine whether you want to install AD CS to issue server certificates to your NPSs or whether you want to purchase server certificates from a public CA, such as VeriSign.

Plan network policies

Network policies are used by NPS to determine whether connection requests received from RADIUS clients are authorized. NPS also uses the dial-in properties of the user account to make an authorization determination.

Because network policies are processed in the order in which they appear in the NPS snap-in, plan to place your most restrictive policies first in the list of policies. For each connection request, NPS attempts to match the conditions of the policy with the connection request properties. NPS examines each network policy in order until it finds a match. If it does not find a match, the connection request is rejected.

Key steps

During the planning for network policies, you can use the following steps.

- Determine the preferred NPS processing order of network policies, from most restrictive to least restrictive.
- Determine the policy state. The policy state can have the value of enabled or disabled. If the policy is enabled, NPS evaluates the policy while performing authorization. If the policy is not enabled, it is not evaluated.
- Determine the policy type. You must determine whether the policy is designed to grant access when the conditions of the policy are matched by the connection request or whether the policy is designed to deny access when the conditions of the policy are matched by the connection request. For example, if you want to explicitly deny wireless access to the members of a Windows group, you can create a network policy that specifies the group, the wireless connection method, and that has a policy type setting of Deny access.
- Determine whether you want NPS to ignore the dial-in properties of user accounts that are members of the group on which the policy is based. When this setting is not enabled, the dial-in properties of user accounts override settings that are configured in network policies. For example, if a network policy is configured that

grants access to a user but the dial-in properties of the user account for that user are set to deny access, the user is denied access. But if you enable the policy type setting Ignore user account dial-in properties, the same user is granted access to the network.

- Determine whether the policy uses the policy source setting. This setting allows you to easily specify a source for all access requests. Possible sources are a Terminal Services Gateway (TS Gateway), a remote access server (VPN or dial-up), a DHCP server, a wireless access point, and a Health Registration Authority server. Alternatively, you can specify a vendor-specific source.
- Determine the conditions that must be matched in order for the network policy to be applied.
- Determine the settings that are applied if the conditions of the network policy are matched by the connection request.
- Determine whether you want to use, modify, or delete the default network policies.

Plan NPS accounting

NPS provides the ability to log RADIUS accounting data, such as user authentication and accounting requests, in three formats: IAS format, database-compatible format, and Microsoft SQL Server logging.

IAS format and database-compatible format create log files on the local NPS in text file format.

SQL Server logging provides the ability to log to a SQL Server 2000 or SQL Server 2005 XML-compliant database, extending RADIUS accounting to leverage the advantages of logging to a relational database.

Key steps

During the planning for NPS accounting, you can use the following steps.

- Determine whether you want to store NPS accounting data in log files or in a SQL Server database.

NPS accounting using local log files

Recording user authentication and accounting requests in log files is used primarily for connection analysis and billing purposes, and is also useful as a security investigation tool, providing you with a method for tracking the activity of a malicious user after an attack.

Key steps

During the planning for NPS accounting using local log files, you can use the following steps.

- Determine the text file format that you want to use for your NPS log files.
- Choose the type of information that you want to log. You can log accounting requests, authentication requests, and periodic status.
- Determine the hard disk location where you want to store your log files.
- Design your log file backup solution. The hard disk location where you store your log files should be a location that allows you to easily back up your data. In addition, the hard disk location should be protected by configuring the access control list (ACL) for the folder where the log files are stored.
- Determine the frequency at which you want new log files to be created. If you want log files to be created based on the file size, determine the maximum file size allowed before a new log file is created by NPS.
- Determine whether you want NPS to delete older log files if the hard disk runs out of storage space.
- Determine the application or applications that you want to use to view accounting data and produce reports.

NPS SQL Server logging

NPS SQL Server logging is used when you need session state information, for report creation and data analysis

purposes, and to centralize and simplify management of your accounting data.

NPS provides the ability to use SQL Server logging to record user authentication and accounting requests received from one or more network access servers to a data source on a computer running the Microsoft SQL Server Desktop Engine (MSDE 2000), or any version of SQL Server later than SQL Server 2000.

Accounting data is passed from NPS in XML format to a stored procedure in the database, which supports both structured query language (SQL) and XML (SQLXML). Recording user authentication and accounting requests in an XML-compliant SQL Server database enables multiple NPSs to have one data source.

Key steps

During the planning for NPS accounting by using NPS SQL Server logging, you can use the following steps.

- Determine whether you or another member of your organization has SQL Server 2000 or SQL Server 2005 relational database development experience and you understand how to use these products to create, modify, administer, and manage SQL Server databases.
- Determine whether SQL Server is installed on the NPS or on a remote computer.
- Design the stored procedure that you will use in your SQL Server database to process incoming XML files that contain NPS accounting data.
- Design the SQL Server database replication structure and flow.
- Determine the application or applications that you want to use to view accounting data and produce reports.
- Plan to use network access servers that send the Class attribute in all accounting-requests. The Class attribute is sent to the RADIUS client in an Access-Accept message, and is useful for correlating Accounting-Request messages with authentication sessions. If the Class attribute is sent by the network access server in the accounting request messages, it can be used to match the accounting and authentication records. The combination of the attributes Unique-Serial-Number, Service-Reboot-Time, and Server-Address must be a unique identification for each authentication that the server accepts.
- Plan to use network access servers that support interim accounting.
- Plan to use network access servers that send Accounting-on and Accounting-off messages.
- Plan to use network access servers that support the storing and forwarding of accounting data. Network access servers that support this feature can store accounting data when the network access server cannot communicate with the NPS. When the NPS is available, the network access server forwards the stored records to the NPS, providing increased reliability in accounting over network access servers that do not provide this feature.
- Plan to always configure the Acct-Interim-Interval attribute in network policies. The Acct-Interim-Interval attribute sets the interval (in seconds) between each interim update that the network access server sends. According to RFC 2869, the value of the Acct-Interim-Interval attribute must not be smaller than 60 seconds, or one minute, and should not be smaller than 600 seconds, or 10 minutes. For more information, see RFC 2869, "RADIUS Extensions."
- Ensure that logging of periodic status is enabled on your NPSs.

Plan NPS as a RADIUS proxy

9/1/2018 • 9 minutes to read • [Edit Online](#)

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

When you deploy Network Policy Server (NPS) as a Remote Authentication Dial-In User Service (RADIUS) proxy, NPS receives connection requests from RADIUS clients, such as network access servers or other RADIUS proxies, and then forwards these connection requests to servers running NPS or other RADIUS servers. You can use these planning guidelines to simplify your RADIUS deployment.

These planning guidelines do not include circumstances in which you want to deploy NPS as a RADIUS server. When you deploy NPS as a RADIUS server, NPS performs authentication, authorization, and accounting for connection requests for the local domain and for domains that trust the local domain.

Before you deploy NPS as a RADIUS proxy on your network, use the following guidelines to plan your deployment.

- Plan NPS configuration.
- Plan RADIUS clients.
- Plan remote RADIUS server groups.
- Plan attribute manipulation rules for message forwarding.
- Plan connection request policies.
- Plan NPS accounting.

Plan NPS configuration

When you use NPS as a RADIUS proxy, NPS forwards connection requests to an NPS or other RADIUS servers for processing. Because of this, the domain membership of the NPS proxy is irrelevant. The proxy does not need to be registered in Active Directory Domain Services (AD DS) because it does not need access to the dial-in properties of user accounts. In addition, you do not need to configure network policies on an NPS proxy because the proxy does not perform authorization for connection requests. The NPS proxy can be a domain member or it can be a stand-alone server with no domain membership.

NPS must be configured to communicate with RADIUS clients, also called network access servers, by using the RADIUS protocol. In addition, you can configure the types of events that NPS records in the event log and you can enter a description for the server.

Key steps

During the planning for NPS proxy configuration, you can use the following steps.

- Determine the RADIUS ports that the NPS proxy uses to receive RADIUS messages from RADIUS clients and to send RADIUS messages to members of remote RADIUS server groups. The default User Datagram Protocol (UDP) ports are 1812 and 1645 for RADIUS authentication messages and UDP ports 1813 and 1646 for RADIUS accounting messages.
- If the NPS proxy is configured with multiple network adapters, determine the adapters over which you want RADIUS traffic to be allowed.
- Determine the types of events that you want NPS to record in the Event Log. You can log rejected

connection requests, successful connection requests, or both.

- Determine whether you are deploying more than one NPS proxy. To provide fault tolerance, use at least two NPS proxies. One NPS proxy is used as the primary RADIUS proxy and the other is used as a backup. Each RADIUS client is then configured on both NPS proxies. If the primary NPS proxy becomes unavailable, RADIUS clients then send Access-Request messages to the alternate NPS proxy.
- Plan the script used to copy one NPS proxy configuration to other NPS proxies to save on administrative overhead and to prevent the incorrect configuration of a server. NPS provides the Netsh commands that allow you to copy all or part of an NPS proxy configuration for import onto another NPS proxy. You can run the commands manually at the Netsh prompt. However, if you save your command sequence as a script, you can run the script at a later date if you decide to change your proxy configurations.

Plan RADIUS clients

RADIUS clients are network access servers, such as wireless access points, virtual private network (VPN) servers, 802.1X-capable switches, and dial-up servers. RADIUS proxies, which forward connection request messages to RADIUS servers, are also RADIUS clients. NPS supports all network access servers and RADIUS proxies that comply with the RADIUS protocol, as described in RFC 2865, "Remote Authentication Dial-in User Service (RADIUS)," and RFC 2866, "RADIUS Accounting."

In addition, both wireless access points and switches must be capable of 802.1X authentication. If you want to deploy Extensible Authentication Protocol (EAP) or Protected Extensible Authentication Protocol (PEAP), access points and switches must support the use of EAP.

To test basic interoperability for PPP connections for wireless access points, configure the access point and the access client to use Password Authentication Protocol (PAP). Use additional PPP-based authentication protocols, such as PEAP, until you have tested the ones that you intend to use for network access.

Key steps

During the planning for RADIUS clients, you can use the following steps.

- Document the vendor-specific attributes (VSAs) you must configure in NPS. If your NASs require VSAs, log the VSA information for later use when you configure your network policies in NPS.
- Document the IP addresses of RADIUS clients and your NPS proxy to simplify the configuration of all devices. When you deploy your RADIUS clients, you must configure them to use the RADIUS protocol, with the NPS proxy IP address entered as the authenticating server. And when you configure NPS to communicate with your RADIUS clients, you must enter the RADIUS client IP addresses into the NPS snap-in.
- Create shared secrets for configuration on the RADIUS clients and in the NPS snap-in. You must configure RADIUS clients with a shared secret, or password, that you will also enter into the NPS snap-in while configuring RADIUS clients in NPS.

Plan remote RADIUS server groups

When you configure a remote RADIUS server group on an NPS proxy, you are telling the NPS proxy where to send some or all connection request messages that it receives from network access servers and NPS proxies or other RADIUS proxies.

You can use NPS as a RADIUS proxy to forward connection requests to one or more remote RADIUS server groups, and each group can contain one or more RADIUS servers. When you want the NPS proxy to forward messages to multiple groups, configure one connection request policy per group. The connection request policy contains additional information, such as attribute manipulation rules, that tell the NPS proxy which messages to send to the remote RADIUS server group specified in the policy.

You can configure remote RADIUS server groups by using the Netsh commands for NPS, by configuring groups directly in the NPS snap-in under Remote RADIUS Server Groups, or by running the New Connection Request Policy wizard.

Key steps

During the planning for remote RADIUS server groups, you can use the following steps.

- Determine the domains that contain the RADIUS servers to which you want the NPS proxy to forward connection requests. These domains contain the user accounts for users that connect to the network through the RADIUS clients you deploy.
- Determine whether you need to add new RADIUS servers in domains where RADIUS is not already deployed.
- Document the IP addresses of RADIUS servers that you want to add to remote RADIUS server groups.
- Determine how many remote RADIUS server groups you need to create. In some cases, it is best to create one remote RADIUS server group per domain, and then add the RADIUS servers for the domain to the group. However, there might be cases in which you have a large amount of resources in one domain, including a large number of users with user accounts in the domain, a large number of domain controllers, and a large number of RADIUS servers. Or your domain might cover a large geographical area, causing you to have network access servers and RADIUS servers in locations that are distant from each other. In these and possibly other cases, you can create multiple remote RADIUS server groups per domain.
- Create shared secrets for configuration on the NPS proxy and on the remote RADIUS servers.

Plan attribute manipulation rules for message forwarding

Attribute manipulation rules, which are configured in connection request policies, allow you to identify the Access-Request messages that you want to forward to a specific remote RADIUS server group.

You can configure NPS to forward all connection requests to one remote RADIUS server group without using attribute manipulation rules.

If you have more than one location to which you want to forward connection requests, however, you must create a connection request policy for each location, then configure the policy with the remote RADIUS server group to which you want to forward messages as well as with the attribute manipulation rules that tell NPS which messages to forward.

You can create rules for the following attributes.

- Called-Station-ID. The phone number of the network access server (NAS). The value of this attribute is a character string. You can use pattern-matching syntax to specify area codes.
- Calling-Station-ID. The phone number used by the caller. The value of this attribute is a character string. You can use pattern-matching syntax to specify area codes.
- User-Name. The user name that is provided by the access client and that is included by the NAS in the RADIUS Access-Request message. The value of this attribute is a character string that typically contains a realm name and a user account name.

To correctly replace or convert realm names in the user name of a connection request, you must configure attribute manipulation rules for the User-Name attribute on the appropriate connection request policy.

Key steps

During the planning for attribute manipulation rules, you can use the following steps.

- Plan message routing from the NAS through the proxy to the remote RADIUS servers to verify that you

have a logical path with which to forward messages to the RADIUS servers.

- Determine one or more attributes that you want to use for each connection request policy.
- Document the attribute manipulation rules that you plan to use for each connection request policy, and match the rules to the remote RADIUS server group to which messages are forwarded.

Plan connection request policies

The default connection request policy is configured for NPS when it is used as a RADIUS server. Additional connection request policies can be used to define more specific conditions, create attribute manipulation rules that tell NPS which messages to forward to remote RADIUS server groups, and to specify advanced attributes. Use the New Connection Request Policy Wizard to create either common or custom connection request policies.

Key steps

During the planning for connection request policies, you can use the following steps.

- Delete the default connection request policy on each server running NPS that functions solely as a RADIUS proxy.
- Plan additional conditions and settings that are required for each policy, combining this information with the remote RADIUS server group and the attribute manipulation rules planned for the policy.
- Design the plan to distribute common connection request policies to all NPS proxies. Create policies common to multiple NPS proxies on one NPS, and then use the Netsh commands for NPS to import the connection request policies and server configuration on all other proxies.

Plan NPS accounting

When you configure NPS as a RADIUS proxy, you can configure it to perform RADIUS accounting by using NPS format log files, database-compatible format log files, or NPS SQL Server logging.

You can also forward accounting messages to a remote RADIUS server group that performs accounting by using one of these logging formats.

Key steps

During the planning for NPS accounting, you can use the following steps.

- Determine whether you want the NPS proxy to perform accounting services or to forward accounting messages to a remote RADIUS server group for accounting.
- Plan to disable local NPS proxy accounting if you plan to forward accounting messages to other servers.
- Plan connection request policy configuration steps if you plan to forward accounting messages to other servers. If you disable local accounting for the NPS proxy, each connection request policy that you configure on that proxy must have accounting message forwarding enabled and configured properly.
- Determine the logging format that you want to use: IAS format log files, database-compatible format log files, or NPS SQL Server logging.

To configure load balancing for NPS as a RADIUS proxy, see [NPS Proxy Server Load Balancing](#).

Deploy Network Policy Server

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic for information about deploying Network Policy Server.

NOTE

For additional Network Policy Server documentation, you can use the following library sections.

- [Getting Started with Network Policy Server](#)
- [Plan Network Policy Server](#)
- [Manage Network Policy Server](#)

The Windows Server 2016 Core Network Guide includes a section on planning and installing Network Policy Server (NPS), and the technologies presented in the guide serve as prerequisites for deploying NPS in an Active Directory domain. For more information, see the section "Deploy NPS1" in the Windows Server 2016 [Core Network Guide](#).

Deploy NPS Certificates for VPN and 802.1X Access

If you want to deploy authentication methods like Extensible Authentication Protocol (EAP) and Protected EAP that require the use of server certificates on your NPS, you can deploy NPS certificates with the guide [Deploy Server Certificates for 802.1X Wired and Wireless Deployments](#).

Deploy NPS for 802.1X Wireless Access

To deploy NPS for wireless access, you can use the guide [Deploy Password-Based 802.1X Authenticated Wireless Access](#).

Deploy NPS for Windows 10 VPN Access

You can use NPS to process connection requests for Always On Virtual Private Network (VPN) connections for remote employees that are using computers and devices running Windows 10.

For more information, see the [Remote Access Always On VPN Deployment Guide for Windows Server 2016 and Windows 10](#).

Manage Network Policy Server (NPS)

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the topics in this section to manage Network Policy Server.

NOTE

For additional Network Policy Server documentation, you can use the following library sections.

- [Getting Started with Network Policy Server](#)
- [Plan Network Policy Server](#)
- [Deploy Network Policy Server](#)

This section contains the following topics.

- [Network Policy Server Management with Administration Tools](#)
- [Configure Connection Request Policies](#)
- [Configure Firewalls for RADIUS Traffic](#)
- [Configure Network Policies](#)
- [Configure Network Policy Server Accounting](#)
- [Configure RADIUS Clients](#)
- [Configure Remote RADIUS Server Groups](#)
- [Manage Certificates Used with NPS](#)
- [Manage NPSs](#)
- [Manage NPS Templates](#)

Network Policy Server Management with Administration Tools

9/1/2018 • 7 minutes to read • [Edit Online](#)

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

You can use this topic to learn about the tools that you can use to manage your NPSs.

After you install NPS, you can administer NPSs:

- Locally, by using the NPS Microsoft Management Console (MMC) snap-in, the static NPS console in Administrative Tools, Windows PowerShell commands, or the Network Shell (Netsh) commands for NPS.
- From a remote NPS, by using the NPS MMC snap-in, the Netsh commands for NPS, the Windows PowerShell commands for NPS, or Remote Desktop Connection.
- From a remote workstation, by using Remote Desktop Connection in combination with other tools, such as the NPS MMC or Windows PowerShell.

NOTE

In Windows Server 2016, you can manage the local NPS by using the NPS console. To manage both remote and local NPSs, you must use the NPS MMC snap-in.

The following sections provide instructions on how to manage your local and remote NPSs.

Configure the Local NPS by Using the NPS Console

After you have installed NPS, you can use this procedure to manage the local NPS by using the NPS MMC.

Administrative Credentials

To complete this procedure, you must be a member of the Administrators group.

To configure the local NPS by using the NPS console

1. In Server Manager, click **Tools**, and then click **Network Policy Server**. The NPS console opens.
2. In the NPS console, click NPS (Local). In the details pane, choose either **Standard Configuration** or **Advanced Configuration**, and then do one of the following based upon your selection:
 - If you choose **Standard Configuration**, select a scenario from the list, and then follow the instructions to start a configuration wizard.
 - If you choose **Advanced Configuration**, click the arrow to expand **Advanced Configuration options**, and then review and configure the available options based on the NPS functionality that you want - RADIUS server, RADIUS proxy, or both.

Manage Multiple NPSs by Using the NPS MMC Snap-in

You can use this procedure to manage the local NPS and multiple remote NPSs by using the NPS MMC snap-in.

Before performing the procedure below, you must install NPS on the local computer and on remote computers.

Depending on network conditions and the number of NPSs you manage by using the NPS MMC snap-in,

response of the MMC snap-in might be slow. In addition, NPS configuration traffic is sent over the network during a remote administration session by using the NPS snap-in. Ensure that your network is physically secure and that malicious users do not have access to this network traffic.

Administrative Credentials

To complete this procedure, you must be a member of the Administrators group.

To manage multiple NPSs by using the NPS snap-in

1. To open the MMC, run Windows PowerShell as an Administrator. In Windows PowerShell, type **mmc**, and then press ENTER. The Microsoft Management Console opens.
2. In the MMC, on the **File** menu, click **Add/Remove Snap-in**. The **Add or Remove Snap-ins** dialog box opens.
3. In **Add or Remove Snap-ins**, in **Available snap-ins**, scroll down the list, click **Network Policy Server**, and then click **Add**. The **Select Computer** dialog box opens.
4. In **Select Computer**, verify that **Local computer (the computer on which this console is running)** is selected, and then click **OK**. The snap-in for the local NPS is added to the list in **Selected snap-ins**.
5. In **Add or Remove Snap-ins**, in **Available snap-ins**, ensure that **Network Policy Server** is still selected, and then click **Add**. The **Select Computer** dialog box opens again.
6. In **Select Computer**, click **Another computer**, and then type the IP address or fully qualified domain name (FQDN) of the remote NPS that you want to manage by using the NPS snap-in. Optionally, you can click **Browse** to peruse the directory for the computer that you want to add. Click **OK**.
7. Repeat steps 5 and 6 to add more NPSs to the NPS snap-in. When you have added all the NPSs you want to manage, click **OK**.
8. To save the NPS snap-in for later use, click **File**, and then click **Save**. In the **Save As** dialog box, browse to the hard disk location where you want to save the file, type a name for your Microsoft Management Console (.msc) file, and then click **Save**.

Manage an NPS by Using Remote Desktop Connection

You can use this procedure to manage a remote NPS by using Remote Desktop Connection.

By using Remote Desktop Connection, you can remotely manage your NPSs running Windows Server 2016. You can also remotely manage NPSs from a computer running Windows 10 or earlier Windows client operating systems.

You can use Remote Desktop connection to manage multiple NPSs by using one of two methods.

1. Create a Remote Desktop connection to each of your NPSs individually.
2. Use Remote Desktop to connect to one NPS, and then use the NPS MMC on that server to manage other remote servers. For more information, see the previous section **Manage Multiple NPSs by Using the NPS MMC Snap-in**.

Administrative Credentials

To complete this procedure, you must be a member of the Administrators group on the NPS.

To manage an NPS by using Remote Desktop Connection

1. On each NPS that you want to manage remotely, in Server Manager, select **Local Server**. In the Server Manager details pane, view the **Remote Desktop** setting, and do one of the following.
 - a. If the value of the **Remote Desktop** setting is **Enabled**, you do not need to perform some of the steps in this procedure. Skip down to Step 4 to start configuring Remote Desktop User permissions.
 - b. If the **Remote Desktop** setting is **Disabled**, click the word **Disabled**. The **System Properties** dialog box opens on the **Remote** tab.
2. In **Remote Desktop**, click **Allow remote connections to this computer**. The **Remote Desktop**

Connection dialog box opens. Do one of the following.

- a. To customize the network connections that are allowed, click **Windows Firewall with Advanced Security**, and then configure the settings that you want to allow.
 - b. To enable Remote Desktop Connection for all network connections on the computer, click **OK**.
3. In **System Properties**, in **Remote Desktop**, decide whether to enable **Allow connections only from computers running Remote Desktop with Network Level Authentication**, and make your selection.
 4. Click **Select Users**. The **Remote Desktop Users** dialog box opens.
 5. In **Remote Desktop Users**, to grant permission to a user to connect remotely to the NPS, click **Add**, and then type the user name for the user's account. Click **OK**.
 6. Repeat step 5 for each user for whom you want to grant remote access permission to the NPS. When you're done adding users, click **OK** to close the **Remote Desktop Users** dialog box and **OK** again to close the **System Properties** dialog box.
 7. To connect to a remote NPS that you have configured by using the previous steps, click **Start**, scroll down the alphabetical list and then click **Windows Accessories**, and click **Remote Desktop Connection**. The **Remote Desktop Connection** dialog box opens.
 8. In the **Remote Desktop Connection** dialog box, in **Computer**, type the NPS name or IP address. If you prefer, click **Options**, configure additional connection options, and then click **Save** to save the connection for repeated use.
 9. Click **Connect**, and when prompted provide user account credentials for an account that has permissions to log on to and configure the NPS.

Use Netsh NPS commands to manage an NPS

You can use commands in the Netsh NPS context to show and set the configuration of the authentication, authorization, accounting, and auditing database used both by NPS and the Remote Access service. Use commands in the Netsh NPS context to:

- Configure or reconfigure an NPS, including all aspects of NPS that are also available for configuration by using the NPS console in the Windows interface.
- Export the configuration of one NPS (the source server), including registry keys and the NPS configuration store, as a Netsh script.
- Import the configuration to another NPS by using a Netsh script and the exported configuration file from the source NPS.

You can run these commands from the Windows Server 2016 Command Prompt or from Windows PowerShell. You can also run netsh nps commands in scripts and batch files.

Administrative Credentials

To perform this procedure, you must be a member of the Administrators group on the local computer.

To enter the Netsh NPS context on an NPS

1. Open Command Prompt or Windows PowerShell.
2. Type **netsh**, and then press ENTER.
3. Type **nps**, and then press ENTER.
4. To view a list of available commands, type a question mark (?) and press ENTER.

For more information about Netsh NPS commands, see [Netsh Commands for Network Policy Server in Windows Server 2008](#), or download the entire [Netsh Technical Reference](#) from TechNet Gallery. This download is the full Network Shell Technical Reference for Windows Server 2008 and Windows Server 2008 R2. The format is Windows Help (*.chm) in a zip file. These commands are still present in Windows Server 2016 and Windows 10, so you can use netsh in these environments, although using Windows PowerShell is recommended.

Use Windows PowerShell to manage NPSs

You can use Windows PowerShell commands to manage NPSs. For more information, see the following Windows PowerShell command reference topics.

- [Network Policy Server \(NPS\) Cmdlets in Windows PowerShell](#). You can use these netsh commands in Windows Server 2012 R2 or later operating systems.
- [NPS Module](#). You can use these netsh commands in Windows Server 2016.

For more information about NPS administration, see [Manage Network Policy Server \(NPS\)](#).

Configure Connection Request Policies

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to create and configure connection request policies that designate whether the local NPS processes connection requests or forwards them to remote RADIUS server for processing.

Connection request policies are sets of conditions and settings that allow network administrators to designate which Remote Authentication Dial-In User Service (RADIUS) servers perform the authentication and authorization of connection requests that the server running Network Policy Server (NPS) receives from RADIUS clients.

The default connection request policy uses NPS as a RADIUS server and processes all authentication requests locally.

To configure a server running NPS to act as a RADIUS proxy and forward connection requests to other NPS or RADIUS servers, you must configure a remote RADIUS server group in addition to adding a new connection request policy that specifies conditions and settings that the connection requests must match.

You can create a new remote RADIUS server group while you are creating a new connection request policy with the New Connection Request Policy Wizard.

If you do not want the NPS to act as a RADIUS server and process connection requests locally, you can delete the default connection request policy.

If you want the NPS to act as both a RADIUS server, processing connection requests locally, and as a RADIUS proxy, forwarding some connection requests to a remote RADIUS server group, add a new policy using the following procedure and then verify that the default connection request policy is the last policy processed by placing it last in the list of policies.

Add a Connection Request Policy

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

To add a new connection request policy

1. In Server Manager, click **Tools**, and then click **Network Policy Server** to open the NPS console.
2. In the console tree, double-click **Policies**.
3. Right-click **Connection Request Policies**, and then click **New Connection Request Policy**.
4. Use the New Connection Request Policy Wizard to configure your connection request policy and, if not previously configured, a remote RADIUS server group.

For more information about managing NPS, see [Manage Network Policy Server](#).

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Configure Firewalls for RADIUS Traffic

9/1/2018 • 8 minutes to read • [Edit Online](#)

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

Firewalls can be configured to allow or block types of IP traffic to and from the computer or device on which the firewall is running. If firewalls are not properly configured to allow RADIUS traffic between RADIUS clients, RADIUS proxies, and RADIUS servers, network access authentication can fail, preventing users from accessing network resources.

You might need to configure two types of firewalls to allow RADIUS traffic:

- Windows Defender Firewall with Advanced Security on the local server running Network Policy Server (NPS).
- Firewalls running on other computers or hardware devices.

Windows Firewall on the local NPS

By default, NPS sends and receives RADIUS traffic by using User Datagram Protocol (UDP) ports 1812, 1813, 1645, and 1646. Windows Defender Firewall on the NPS is automatically configured with exceptions, during the installation of NPS, to allow this RADIUS traffic to be sent and received.

Therefore, if you are using the default UDP ports, you do not need to change the Windows Defender Firewall configuration to allow RADIUS traffic to and from NPSs.

In some cases, you might want to change the ports that NPS uses for RADIUS traffic. If you configure NPS and your network access servers to send and receive RADIUS traffic on ports other than the defaults, you must do the following:

- Remove the exceptions that allow RADIUS traffic on the default ports.
- Create new exceptions that allow RADIUS traffic on the new ports.

For more information, see [Configure NPS UDP Port Information](#).

Other firewalls

In the most common configuration, the firewall is connected to the Internet and the NPS is an intranet resource that is connected to the perimeter network.

To reach the domain controller within the intranet, the NPS might have:

- An interface on the perimeter network and an interface on the intranet (IP routing is not enabled).
- A single interface on the perimeter network. In this configuration, NPS communicates with domain controllers through another firewall that connects the perimeter network to the intranet.

Configuring the Internet firewall

The firewall that is connected to the Internet must be configured with input and output filters on its Internet interface (and, optionally, its network perimeter interface), to allow the forwarding of RADIUS messages between the NPS and RADIUS clients or proxies on the Internet. Additional filters can be used to allow the passing of traffic to Web servers, VPN servers, and other types of servers on the perimeter network.

Separate input and output packet filters can be configured on the Internet interface and the perimeter network

interface.

Configure Input Filters on the Internet Interface

Configure the following input packet filters on the Internet interface of the firewall to allow the following types of traffic:

- Destination IP address of the perimeter network interface and UDP destination port of 1812 (0x714) of the NPS. This filter allows RADIUS authentication traffic from Internet-based RADIUS clients to the NPS. This is the default UDP port that is used by NPS, as defined in RFC 2865. If you are using a different port, substitute that port number for 1812.
- Destination IP address of the perimeter network interface and UDP destination port of 1813 (0x715) of the NPS. This filter allows RADIUS accounting traffic from Internet-based RADIUS clients to the NPS. This is the default UDP port that is used by NPS, as defined in RFC 2866. If you are using a different port, substitute that port number for 1813.
- (Optional) Destination IP address of the perimeter network interface and UDP destination port of 1645 (0x66D) of the NPS. This filter allows RADIUS authentication traffic from Internet-based RADIUS clients to the NPS. This is the UDP port that is used by older RADIUS clients.
- (Optional) Destination IP address of the perimeter network interface and UDP destination port of 1646 (0x66E) of the NPS. This filter allows RADIUS accounting traffic from Internet-based RADIUS clients to the NPS. This is the UDP port that is used by older RADIUS clients.

Configure Output Filters on the Internet Interface

Configure the following output filters on the Internet interface of the firewall to allow the following types of traffic:

- Source IP address of the perimeter network interface and UDP source port of 1812 (0x714) of the NPS. This filter allows RADIUS authentication traffic from the NPS to Internet-based RADIUS clients. This is the default UDP port that is used by NPS, as defined in RFC 2865. If you are using a different port, substitute that port number for 1812.
- Source IP address of the perimeter network interface and UDP source port of 1813 (0x715) of the NPS. This filter allows RADIUS accounting traffic from the NPS to Internet-based RADIUS clients. This is the default UDP port that is used by NPS, as defined in RFC 2866. If you are using a different port, substitute that port number for 1813.
- (Optional) Source IP address of the perimeter network interface and UDP source port of 1645 (0x66D) of the NPS. This filter allows RADIUS authentication traffic from the NPS to Internet-based RADIUS clients. This is the UDP port that is used by older RADIUS clients.
- (Optional) Source IP address of the perimeter network interface and UDP source port of 1646 (0x66E) of the NPS. This filter allows RADIUS accounting traffic from the NPS to Internet-based RADIUS clients. This is the UDP port that is used by older RADIUS clients.

Configure Input Filters on the Perimeter Network Interface

Configure the following input filters on the perimeter network interface of the firewall to allow the following types of traffic:

- Source IP address of the perimeter network interface and UDP source port of 1812 (0x714) of the NPS. This filter allows RADIUS authentication traffic from the NPS to Internet-based RADIUS clients. This is the default UDP port that is used by NPS, as defined in RFC 2865. If you are using a different port, substitute that port number for 1812.
- Source IP address of the perimeter network interface and UDP source port of 1813 (0x715) of the NPS. This filter allows RADIUS accounting traffic from the NPS to Internet-based RADIUS clients. This is the default UDP port that is used by NPS, as defined in RFC 2866. If you are using a different port, substitute that port number for 1813.
- (Optional) Source IP address of the perimeter network interface and UDP source port of 1645 (0x66D) of the NPS. This filter allows RADIUS authentication traffic from the NPS to Internet-based RADIUS clients. This is

the UDP port that is used by older RADIUS clients.

- (Optional) Source IP address of the perimeter network interface and UDP source port of 1646 (0x66E) of the NPS. This filter allows RADIUS accounting traffic from the NPS to Internet-based RADIUS clients. This is the UDP port that is used by older RADIUS clients.

Configure Output Filters on the Perimeter Network Interface

Configure the following output packet filters on the perimeter network interface of the firewall to allow the following types of traffic:

- Destination IP address of the perimeter network interface and UDP destination port of 1812 (0x714) of the NPS. This filter allows RADIUS authentication traffic from Internet-based RADIUS clients to the NPS. This is the default UDP port that is used by NPS, as defined in RFC 2865. If you are using a different port, substitute that port number for 1812.
- Destination IP address of the perimeter network interface and UDP destination port of 1813 (0x715) of the NPS. This filter allows RADIUS accounting traffic from Internet-based RADIUS clients to the NPS. This is the default UDP port that is used by NPS, as defined in RFC 2866. If you are using a different port, substitute that port number for 1813.
- (Optional) Destination IP address of the perimeter network interface and UDP destination port of 1645 (0x66D) of the NPS. This filter allows RADIUS authentication traffic from Internet-based RADIUS clients to the NPS. This is the UDP port that is used by older RADIUS clients.
- (Optional) Destination IP address of the perimeter network interface and UDP destination port of 1646 (0x66E) of the NPS. This filter allows RADIUS accounting traffic from Internet-based RADIUS clients to the NPS. This is the UDP port that is used by older RADIUS clients.

For added security, you can use the IP addresses of each RADIUS client that sends the packets through the firewall to define filters for traffic between the client and the IP address of the NPS on the perimeter network.

Filters on the perimeter network interface

Configure the following input packet filters on the perimeter network interface of the intranet firewall to allow the following types of traffic:

- Source IP address of the perimeter network interface of the NPS. This filter allows traffic from the NPS on the perimeter network.

Configure the following output filters on the perimeter network interface of the intranet firewall to allow the following types of traffic:

- Destination IP address of the perimeter network interface of the NPS. This filter allows traffic to the NPS on the perimeter network.

Filters on the intranet interface

Configure the following input filters on the intranet interface of the firewall to allow the following types of traffic:

- Destination IP address of the perimeter network interface of the NPS. This filter allows traffic to the NPS on the perimeter network.

Configure the following output packet filters on the intranet interface of the firewall to allow the following types of traffic:

- Source IP address of the perimeter network interface of the NPS. This filter allows traffic from the NPS on the perimeter network.

For more information about managing NPS, see [Manage Network Policy Server](#).

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Configure Network Policies

9/1/2018 • 11 minutes to read • [Edit Online](#)

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

You can use this topic to configure network policies in NPS.

Add a Network Policy

Network Policy Server (NPS) uses network policies and the dial-in properties of user accounts to determine whether a connection request is authorized to connect to the network.

You can use this procedure to configure a new network policy in either the NPS console or the Remote Access console.

Performing authorization

When NPS performs the authorization of a connection request, it compares the request with each network policy in the ordered list of policies, starting with the first policy, and then moving down the list of configured policies. If NPS finds a policy whose conditions match the connection request, NPS uses the matching policy and the dial-in properties of the user account to perform authorization. If the dial-in properties of the user account are configured to grant access or control access through network policy and the connection request is authorized, NPS applies the settings that are configured in the network policy to the connection.

If NPS does not find a network policy that matches the connection request, the connection request is rejected unless the dial-in properties on the user account are set to grant access.

If the dial-in properties of the user account are set to deny access, the connection request is rejected by NPS.

Key settings

When you use the New Network Policy wizard to create a network policy, the value that you specify in **Network connection method** is used to automatically configure the **Policy Type** condition:

- If you keep the default value of **Unspecified**, the network policy that you create is evaluated by NPS for all network connection types that are using any kind of network access server (NAS).
- If you specify a network connection method, NPS evaluates the network policy only if the connection request originates from the type of network access server that you specify.

On the **Access Permission** page, you must select **Access granted** if you want the policy to allow users to connect to your network. If you want the policy to prevent users from connecting to your network, select **Access denied**.

If you want access permission to be determined by user account dial-in properties in Active Directory® Domain Services (AD DS), you can select the **Access is determined by User Dial-in properties** check box.

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

To add a network policy

1. Open the NPS console, and then double-click **Policies**.
2. In the console tree, right-click **Network Policies**, and click **New**. The New Network Policy wizard opens.
3. Use the New Network Policy wizard to create a policy.

Create Network Policies for Dial-Up or VPN with a Wizard

You can use this procedure to create the connection request policies and network policies required to deploy either dial-up servers or virtual private network (VPN) servers as Remote Authentication Dial-In User Service (RADIUS) clients to the NPS RADIUS server.

NOTE

Client computers, such as laptop computers and other computers running client operating systems, are not RADIUS clients. RADIUS clients are network access servers — such as wireless access points, 802.1X authenticating switches, virtual private network (VPN) servers, and dial-up servers — because these devices use the RADIUS protocol to communicate with RADIUS servers such as NPSs.

This procedure explains how to open the New Dial-up or Virtual Private Network Connections wizard in NPS.

After you run the wizard, the following policies are created:

- One connection request policy
- One network policy

You can run the New Dial-up or Virtual Private Network Connections wizard every time you need to create new policies for dial-up servers and VPN servers.

Running the New Dial-up or Virtual Private Network Connections wizard is not the only step required to deploy dial-up or VPN servers as RADIUS clients to the NPS. Both network access methods require that you deploy additional hardware and software components.

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

To create policies for dial-up or VPN with a wizard

1. Open the NPS console. If it is not already selected, click **NPS (Local)**. If you want to create policies on a remote NPS, select the server.
2. In **Getting Started** and **Standard Configuration**, select **RADIUS server for Dial-Up or VPN Connections**. The text and links under the text change to reflect your selection.
3. Click **Configure VPN or Dial-Up with a wizard**. The New Dial-up or Virtual Private Network Connections wizard opens.
4. Follow the instructions in the wizard to complete creation of your new policies.

Create Network Policies for 802.1X Wired or Wireless with a Wizard

You can use this procedure to create the connection request policy and network policy that are required to deploy either 802.1X authenticating switches or 802.1X wireless access points as Remote Authentication Dial-In User Service (RADIUS) clients to the NPS RADIUS server.

This procedure explains how to start the New IEEE 802.1X Secure Wired and Wireless Connections wizard in NPS.

After you run the wizard, the following policies are created:

- One connection request policy
- One network policy

You can run the New IEEE 802.1X Secure Wired and Wireless Connections wizard every time you need to create new policies for 802.1X access.

Running the New IEEE 802.1X Secure Wired and Wireless Connections wizard is not the only step required to deploy 802.1X authenticating switches and wireless access points as RADIUS clients to the NPS. Both network access methods require that you deploy additional hardware and software components.

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

To create policies for 802.1X wired or wireless with a wizard

1. On the NPS, in Server Manager, click **Tools**, and then click **Network Policy Server**. The NPS console opens.
2. If it is not already selected, click **NPS (Local)**. If you want to create policies on a remote NPS, select the server.
3. In **Getting Started** and **Standard Configuration**, select **RADIUS server for 802.1X Wireless or Wired Connections**. The text and links under the text change to reflect your selection.
4. Click **Configure 802.1X using a wizard**. The New IEEE 802.1X Secure Wired and Wireless Connections wizard opens.
5. Follow the instructions in the wizard to complete creation of your new policies.

Configure NPS to Ignore User Account Dial-in Properties

Use this procedure to configure an NPS network policy to ignore the dial-in properties of user accounts in Active Directory during the authorization process. User accounts in Active Directory Users and Computers have dial-in properties that NPS evaluates during the authorization process unless the **Network Access Permission** property of the user account is set to **Control access through NPS Network Policy**.

There are two circumstances where you might want to configure NPS to ignore the dial-in properties of user accounts in Active Directory:

- When you want to simplify NPS authorization by using network policy, but not all of your user accounts have the **Network Access Permission** property set to **Control access through NPS Network Policy**. For example, some user accounts might have the **Network Access Permission** property of the user account set to **Deny access** or **Allow access**.
- When other dial-in properties of user accounts are not applicable to the connection type that is configured in the network policy. For example, properties other than the **Network Access Permission** setting are applicable only to dial-in or VPN connections, but the network policy you are creating is for wireless or authenticating switch connections.

You can use this procedure to configure NPS to ignore user account dial-in properties. If a connection request matches the network policy where this check box is selected, NPS does not use the dial-in properties of the user account to determine whether the user or computer is authorized to access the network; only the settings in the network policy are used to determine authorization.

Membership in **Administrators**, or equivalent, is the minimum required to complete this procedure.

1. On the NPS, in Server Manager, click **Tools**, and then click **Network Policy Server**. The NPS console opens.
2. Double-click **Policies**, click **Network Policies**, and then in the details pane double-click the policy that you want to configure.
3. In the policy **Properties** dialog box, on the **Overview** tab, in **Access Permission**, select the **Ignore user account dial-in properties** check box, and then click **OK**.

To configure NPS to ignore user account dial-in properties

Configure NPS for VLANs

By using VLAN-aware network access servers and NPS in Windows Server 2016, you can provide groups of users with access only to the network resources that are appropriate for their security permissions. For example, you can provide visitors with wireless access to the Internet without allowing them access to your organization network.

In addition, VLANs allow you to logically group network resources that exist in different physical locations or on different physical subnets. For example, members of your sales department and their network resources, such as client computers, servers, and printers, might be located in several different buildings at your organization, but you can place all of these resources on one VLAN that uses the same IP address range. The VLAN then functions, from the end-user perspective, as a single subnet.

You can also use VLANs when you want to segregate a network between different groups of users. After you have determined how you want to define your groups, you can create security groups in the Active Directory Users and Computers snap-in, and then add members to the groups.

Configure a Network Policy for VLANs

You can use this procedure to configure a network policy that assigns users to a VLAN. When you use VLAN-aware network hardware, such as routers, switches, and access controllers, you can configure network policy to instruct the access servers to place members of specific Active Directory groups on specific VLANs. This ability to group network resources logically with VLANs provides flexibility when designing and implementing network solutions.

When you configure the settings of an NPS network policy for use with VLANs, you must configure the attributes **Tunnel-Medium-Type**, **Tunnel-Pvt-Group-ID**, **Tunnel-Type**, and **Tunnel-Tag**.

This procedure is provided as a guideline; your network configuration might require different settings than those described below.

Membership in **Administrators**, or equivalent, is the minimum required to complete this procedure.

To configure a network policy for VLANs

1. On the NPS, in Server Manager, click **Tools**, and then click **Network Policy Server**. The NPS console opens.
2. Double-click **Policies**, click **Network Policies**, and then in the details pane double-click the policy that you want to configure.
3. In the policy **Properties** dialog box, click the **Settings** tab.
4. In policy **Properties**, in **Settings**, in **RADIUS Attributes**, ensure that **Standard** is selected.
5. In the details pane, in **Attributes**, the **Service-Type** attribute is configured with a default value of **Framed**. By default, for policies with access methods of VPN and dial-up, the **Framed-Protocol** attribute is configured with a value of **PPP**. To specify additional connection attributes required for VLANs, click **Add**. The **Add Standard RADIUS Attribute** dialog box opens.
6. In **Add Standard RADIUS Attribute**, in Attributes, scroll down to and add the following attributes:
 - **Tunnel-Medium-Type**. Select a value appropriate to the previous selections you have made for the policy. For example, if the network policy you are configuring is a wireless policy, select **Value: 802 (Includes all 802 media plus Ethernet canonical format)**.
 - **Tunnel-Pvt-Group-ID**. Enter the integer that represents the VLAN number to which group members will be assigned.
 - **Tunnel-Type**. Select **Virtual LANs (VLAN)**.

7. In **Add Standard RADIUS Attribute**, click **Close**.
8. If your network access server (NAS) requires use of the **Tunnel-Tag** attribute, use the following steps to add the **Tunnel-Tag** attribute to the network policy. If your NAS documentation does not mention this attribute, do not add it to the policy. If required, add the attributes as follows:
 - In policy **Properties**, in **Settings**, in **RADIUS Attributes**, click **Vendor Specific**.
 - In the details pane, click **Add**. The **Add Vendor Specific Attribute** dialog box opens.
 - In **Attributes**, scroll down to and select **Tunnel-Tag**, and then click **Add**. The **Attribute Information** dialog box opens.
 - In **Attribute value**, type the value that you obtained from your hardware documentation.

Configure the EAP Payload Size

In some cases, routers or firewalls drop packets because they are configured to discard packets that require fragmentation.

When you deploy NPS with network policies that use the Extensible Authentication Protocol (EAP) with Transport Layer Security (TLS), or EAP-TLS, as an authentication method, the default maximum transmission unit (MTU) that NPS uses for EAP payloads is 1500 bytes.

This maximum size for the EAP payload can create RADIUS messages that require fragmentation by a router or firewall between the NPS and a RADIUS client. If this is the case, a router or firewall positioned between the RADIUS client and the NPS might silently discard some fragments, resulting in authentication failure and the inability of the access client to connect to the network.

Use the following procedure to lower the maximum size that NPS uses for EAP payloads by adjusting the **Framed-MTU** attribute in a network policy to a value no greater than 1344.

Membership in **Administrators**, or equivalent, is the minimum required to complete this procedure.

To configure the **Framed-MTU** attribute

1. On the NPS, in Server Manager, click **Tools**, and then click **Network Policy Server**. The NPS console opens.
2. Double-click **Policies**, click **Network Policies**, and then in the details pane double-click the policy that you want to configure.
3. In the policy **Properties** dialog box, click the **Settings** tab.
4. In **Settings**, in **RADIUS Attributes**, click **Standard**. In the details pane, click **Add**. The **Add Standard RADIUS Attribute** dialog box opens.
5. In **Attributes**, scroll down to and click **Framed-MTU**, and then click **Add**. The **Attribute Information** dialog box opens.
6. In **Attribute Value**, type a value equal to or less than **1344**. Click **OK**, click **Close**, and then click **OK**.

For more information about network policies, see [Network Policies](#).

For examples of pattern-matching syntax to specify network policy attributes, see [Use Regular Expressions in NPS](#).

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Configure Network Policy Server Accounting

9/21/2018 • 9 minutes to read • [Edit Online](#)

There are three types of logging for Network Policy Server (NPS):

- **Event logging.** Used primarily for auditing and troubleshooting connection attempts. You can configure NPS event logging by obtaining the NPS properties in the NPS console.
- **Logging user authentication and accounting requests to a local file.** Used primarily for connection analysis and billing purposes. Also useful as a security investigation tool because it provides you with a method of tracking the activity of a malicious user after an attack. You can configure local file logging using the Accounting Configuration wizard.
- **Logging user authentication and accounting requests to a Microsoft SQL Server XML-compliant database.** Used to allow multiple servers running NPS to have one data source. Also provides the advantages of using a relational database. You can configure SQL Server logging by using the Accounting Configuration wizard.

Use the Accounting Configuration wizard

By using the Accounting Configuration wizard, you can configure the following four accounting settings:

- **SQL logging only.** By using this setting, you can configure a data link to a SQL Server that allows NPS to connect to and send accounting data to the SQL server. In addition, the wizard can configure the database on the SQL Server to ensure that the database is compatible with NPS SQL server logging.
- **Text logging only.** By using this setting, you can configure NPS to log accounting data to a text file.
- **Parallel logging.** By using this setting, you can configure the SQL Server data link and database. You can also configure text file logging so that NPS logs simultaneously to the text file and the SQL Server database.
- **SQL logging with backup.** By using this setting, you can configure the SQL Server data link and database. In addition, you can configure text file logging that NPS uses if SQL Server logging fails.

In addition to these settings, both SQL Server logging and text logging allow you to specify whether NPS continues to process connection requests if logging fails. You can specify this in the **Logging failure action section** in local file logging properties, in SQL server logging properties, and while you are running the Accounting Configuration Wizard.

To run the Accounting Configuration Wizard

To run the Accounting Configuration Wizard, complete the following steps:

1. Open the NPS console or the NPS Microsoft Management Console (MMC) snap-in.
2. In the console tree, click **Accounting**.
3. In the details pane, in **Accounting**, click **Configure Accounting**.

Configure NPS Log File Properties

You can configure Network Policy Server (NPS) to perform Remote Authentication Dial-In User Service (RADIUS) accounting for user authentication requests, Access-Accept messages, Access-Reject messages, accounting requests and responses, and periodic status updates. You can use this procedure to configure the log files in which you want to store the accounting data.

For more information about interpreting log files, see [Interpret NPS Database Format Log Files](#).

To prevent the log files from filling the hard drive, it is strongly recommended that you keep them on a partition that is separate from the system partition. The following provides more information about configuring accounting for NPS:

- To send the log file data for collection by another process, you can configure NPS to write to a named pipe. To use named pipes, set the log file folder to `\.\pipe` or `\ComputerName\pipe`. The named pipe server program creates a named pipe called `\.\pipe\iaslog.log` to accept the data. In the Local file properties dialog box, in Create a new log file, select Never (unlimited file size) when you use named pipes.
- The log file directory can be created by using system environment variables (instead of user variables), such as `%systemdrive%`, `%systemroot%`, and `%windir%`. For example, the following path, using the environment variable `%windir%`, locates the log file at the system directory in the subfolder `\System32\Logs` (that is, `%windir%\System32\Logs`).
- Switching log file formats does not cause a new log to be created. If you change log file formats, the file that is active at the time of the change will contain a mixture of the two formats (records at the start of the log will have the previous format, and records at the end of the log will have the new format).
- If RADIUS accounting fails due to a full hard disk drive or other causes, NPS stops processing connection requests, preventing users from accessing network resources.
- NPS provides the ability to log to a Microsoft® SQL Server™ database in addition to, or instead of, logging to a local file.

Membership in the **Domain Admins** group is the minimum required to perform this procedure.

To configure NPS log file properties

1. Open the NPS console or the NPS Microsoft Management Console (MMC) snap-in.
2. In the console tree, click **Accounting**.
3. In the details pane, in **Log File Properties**, click **Change Log File Properties**. The **Log File Properties** dialog box opens.
4. In **Log File Properties**, on the **Settings** tab, in **Log the following information**, ensure that you choose to log enough information to achieve your accounting goals. For example, if your logs need to accomplish session correlation, select all check boxes.
5. In **Logging failure action**, select **If logging fails, discard connection requests** if you want NPS to stop processing Access-Request messages when log files are full or unavailable for some reason. If you want NPS to continue processing connection requests if logging fails, do not select this check box.
6. In the **Log File Properties** dialog box, click the **Log File** tab.
7. On the **Log File** tab, in **Directory**, type the location where you want to store NPS log files. The default location is the `systemroot\System32\LogFiles` folder.
If you do not supply a full path statement in **Log File Directory**, the default path is used. For example, if you type **NPSLogFile** in **Log File Directory**, the file is located at `%systemroot%\System32\NPSLogFile`.
8. In **Format**, click **DTS Compliant**. If you prefer, you can instead select a legacy file format, such as **ODBC (Legacy)** or **IAS (Legacy)**.
ODBC and **IAS** legacy file types contain a subset of the information that NPS sends to its SQL Server database. The **DTS Compliant** file type's XML format is identical to the XML format that NPS uses to import data into its SQL Server database. Therefore, the **DTS Compliant** file format provides a more efficient and complete transfer of data into the standard SQL Server database for NPS.
9. In **Create a new log file**, to configure NPS to start new log files at specified intervals, click the interval that you want to use:
 - For heavy transaction volume and logging activity, click **Daily**.
 - For lesser transaction volumes and logging activity, click **Weekly** or **Monthly**.
 - To store all transactions in one log file, click **Never (unlimited file size)**.

- To limit the size of each log file, click **When log file reaches this size**, and then type a file size, after which a new log is created. The default size is 10 megabytes (MB).
10. If you want NPS to delete old log files to create disk space for new log files when the hard disk is near capacity, ensure that **When disk is full delete older log files** is selected. This option is not available, however, if the value of **Create a new log file** is **Never (unlimited file size)**. Also, if the oldest log file is the current log file, it is not deleted.

Configure NPS SQL Server Logging

You can use this procedure to log RADIUS accounting data to a local or remote database running Microsoft SQL Server.

NOTE

NPS formats accounting data as an XML document that it sends to the **report_event** stored procedure in the SQL Server database that you designate in NPS. For SQL Server logging to function properly, you must have a stored procedure named **report_event** in the SQL Server database that can receive and parse the XML documents from NPS.

Membership in Domain Admins, or equivalent, is the minimum required to complete this procedure.

To configure SQL Server logging in NPS

1. Open the NPS console or the NPS Microsoft Management Console (MMC) snap-in.
2. In the console tree, click **Accounting**.
3. In the details pane, in **SQL Server Logging Properties**, click **Change SQL Server Logging Properties**. The **SQL Server Logging Properties** dialog box opens.
4. In **Log the following information**, select the information that you want to log:
 - To log all accounting requests, click **Accounting requests**.
 - To log authentication requests, click **Authentication requests**.
 - To log periodic accounting status, click **Periodic accounting status**.
 - To log periodic status, such as interim accounting requests, click **Periodic status**.
5. To configure the number of concurrent sessions allowed between the server running NPS and the SQL Server, type a number in **Maximum number of concurrent sessions**.
6. To configure the SQL Server data source, in **SQL Server Logging**, click **Configure**. The **Data Link Properties** dialog box opens. On the **Connection** tab, specify the following:
 - To specify the name of the server on which the database is stored, type or select a name in **Select or enter a server name**.
 - To specify the authentication method with which to log on to the server, click **Use Windows NT integrated security**. Or, click **Use a specific user name and password**, and then type credentials in **User name** and **Password**.
 - To allow a blank password, click **Blank password**.
 - To store the password, click **Allow saving password**.
 - To specify which database to connect to on the computer running SQL Server, click **Select the database on the server**, and then select a database name from the list.
7. To test the connection between NPS and SQL Server, click **Test Connection**. Click **OK** to close **Data Link Properties**.
8. In **Logging failure action**, select **Enable text file logging for failover** if you want NPS to continue with text file logging if SQL Server logging fails.
9. In **Logging failure action**, select **If logging fails, discard connection requests** if you want NPS to stop processing Access-Request messages when log files are full or unavailable for some reason. If you want NPS to continue processing connection requests if logging fails, do not select this check box.

Ping user-name

Some RADIUS proxy servers and network access servers periodically send authentication and accounting requests (known as ping requests) to verify that the NPS is present on the network. These ping requests include fictional user names. When NPS processes these requests, the event and accounting logs become filled with access reject records, making it more difficult to keep track of valid records.

When you configure a registry entry for **ping user-name**, NPS matches the registry entry value against the user name value in ping requests by other servers. A **ping user-name** registry entry specifies the fictional user name (or a user name pattern, with variables, that matches the fictional user name) sent by RADIUS proxy servers and network access servers. When NPS receives ping requests that match the **ping user-name** registry entry value, NPS rejects the authentication requests without processing the request. NPS does not record transactions involving the fictional user name in any log files, which makes the event log easier to interpret.

Ping user-name is not installed by default. You must add **ping user-name** to the registry. You can add an entry to the registry using Registry Editor.

Caution

Incorrectly editing the registry might severely damage your system. Before making changes to the registry, you should back up any valued data on the computer.

To add ping user-name to the registry

Ping user-name can be added to the following registry key as a string value by a member of the local Administrators group:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\IAS\Parameters

- **Name:** `ping user-name`
- **Type:** `REG_SZ`
- **Data:** *User name*

TIP

To indicate more than one user name for a **ping user-name** value, enter a name pattern, such as a DNS name, including wildcard characters, in **Data**.

Configure RADIUS Clients

9/1/2018 • 6 minutes to read • [Edit Online](#)

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

You can use this topic to configure network access servers as RADIUS Clients in NPS.

When you add a new network access server (VPN server, wireless access point, authenticating switch, or dial-up server) to your network, you must add the server as a RADIUS client in NPS, and then configure the RADIUS client to communicate with the NPS.

IMPORTANT

Client computers and devices, such as laptop computers, tablets, phones, and other computers running client operating systems, are not RADIUS clients. RADIUS clients are network access servers - such as wireless access points, 802.1X-capable switches, virtual private network (VPN) servers, and dial-up servers - because they use the RADIUS protocol to communicate with RADIUS servers, such as Network Policy Server (NPS) servers.

This step is also necessary when your NPS is a member of a remote RADIUS server group that is configured on an NPS proxy. In this circumstance, in addition to performing the steps in this task on the NPS proxy, you must do the following:

- On the NPS proxy, configure a remote RADIUS server group that contains the NPS.
- On the remote NPS, configure the NPS proxy as a RADIUS client.

To perform the procedures in this topic, you must have at least one network access server (VPN server, wireless access point, authenticating switch, or dial-up server) or NPS proxy physically installed on your network.

Configure the Network Access Server

Use this procedure to configure network access servers for use with NPS. When you deploy network access servers (NASs) as RADIUS clients, you must configure the clients to communicate with the NPSs where the NASs are configured as clients.

This procedure provides general guidelines about the settings you should use to configure your NASs; for specific instructions on how to configure the device you are deploying on your network, see your NAS product documentation.

To configure the network access server

1. On the NAS, in **RADIUS settings**, select **RADIUS authentication** on User Datagram Protocol (UDP) port **1812** and **RADIUS accounting** on UDP port **1813**.
2. In **Authentication server** or **RADIUS server**, specify your NPS by IP address or fully qualified domain name (FQDN), depending on the requirements of the NAS.
3. In **Secret** or **Shared secret**, type a strong password. When you configure the NAS as a RADIUS client in NPS, you will use the same password, so do not forget it.
4. If you are using PEAP or EAP as an authentication method, configure the NAS to use EAP authentication.
5. If you are configuring a wireless access point, in **SSID**, specify a Service Set Identifier (SSID), which is an alphanumeric string that serves as the network name. This name is broadcast by access points to wireless clients and is visible to users at your wireless fidelity (Wi-Fi) hotspots.
6. If you are configuring a wireless access point, in **802.1X and WPA**, enable **IEEE 802.1X authentication** if you

want to deploy PEAP-MS-CHAP v2, PEAP-TLS, or EAP-TLS.

Add the Network Access Server as a RADIUS Client in NPS

Use this procedure to add a network access server as a RADIUS client in NPS. You can use this procedure to configure a NAS as a RADIUS client by using the NPS console.

To complete this procedure, you must be a member of the **Administrators** group.

To add a network access server as a RADIUS client in NPS

1. On the NPS, in Server Manager, click **Tools**, and then click **Network Policy Server**. The NPS console opens.
2. In the NPS console, double-click **RADIUS Clients and Servers**. Right-click **RADIUS Clients**, and then click **New RADIUS Client**.
3. In **New RADIUS Client**, verify that the **Enable this RADIUS client** check box is selected.
4. In **New RADIUS Client**, in **Friendly name**, type a display name for the NAS. In **Address (IP or DNS)**, type the NAS IP address or fully qualified domain name (FQDN). If you enter the FQDN, click **Verify** if you want to verify that the name is correct and maps to a valid IP address.
5. In **New RADIUS Client**, in **Vendor**, specify the NAS manufacturer name. If you are not sure of the NAS manufacturer name, select **RADIUS standard**.
6. In **New RADIUS Client**, in **Shared secret**, do one of the following:
 - Ensure that **Manual** is selected, and then in **Shared secret**, type the strong password that is also entered on the NAS. Retype the shared secret in **Confirm shared secret**.
 - Select **Generate**, and then click **Generate** to automatically generate a shared secret. Save the generated shared secret for configuration on the NAS so that it can communicate with the NPS.
7. In **New RADIUS Client**, in **Additional Options**, if you are using any authentication methods other than EAP and PEAP, and if your NAS supports use of the message authenticator attribute, select **Access Request messages must contain the Message Authenticator attribute**.
8. Click **OK**. Your NAS appears in the list of RADIUS clients configured on the NPS.

Configure RADIUS Clients by IP Address Range in Windows Server 2016 Datacenter

If you are running Windows Server 2016 Datacenter, you can configure RADIUS clients in NPS by IP address range. This allows you to add a large number of RADIUS clients (such as wireless access points) to the NPS console at one time, rather than adding each RADIUS client individually.

You cannot configure RADIUS clients by IP address range if you are running NPS on Windows Server 2016 Standard.

Use this procedure to add a group of network access servers (NASs) as RADIUS clients that are all configured with IP addresses from the same IP address range.

All of the RADIUS clients in the range must use the same configuration and shared secret.

To complete this procedure, you must be a member of the **Administrators** group.

To set up RADIUS clients by IP address range

1. On the NPS, in Server Manager, click **Tools**, and then click **Network Policy Server**. The NPS console opens.
2. In the NPS console, double-click **RADIUS Clients and Servers**. Right-click **RADIUS Clients**, and then click **New RADIUS Client**.
3. In **New RADIUS Client**, in **Friendly name**, type a display name for the collection of NASs.
4. In **Address (IP or DNS)**, type the IP address range for the RADIUS clients by using Classless Inter-Domain Routing (CIDR) notation. For example, if the IP address range for the NASs is 10.10.0.0, type **10.10.0.0/16**.

5. In **New RADIUS Client**, in **Vendor**, specify the NAS manufacturer name. If you are not sure of the NAS manufacturer name, select **RADIUS standard**.
6. In **New RADIUS Client**, in **Shared secret**, do one of the following:
 - Ensure that **Manual** is selected, and then in **Shared secret**, type the strong password that is also entered on the NAS. Retype the shared secret in **Confirm shared secret**.
 - Select **Generate**, and then click **Generate** to automatically generate a shared secret. Save the generated shared secret for configuration on the NAS so that it can communicate with the NPS.
7. In **New RADIUS Client**, in **Additional Options**, if you are using any authentication methods other than EAP and PEAP, and if all of your NASs support use of the message authenticator attribute, select **Access Request messages must contain the Message Authenticator attribute**.
8. Click **OK**. Your NASs appear in the list of RADIUS clients configured on the NPS.

For more information, see [RADIUS Clients](#).

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Configure Remote RADIUS Server Groups

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to configure remote RADIUS server groups when you want to configure NPS to act as a proxy server and forward connection requests to other NPSs for processing.

Add a Remote RADIUS Server Group

You can use this procedure to add a new remote RADIUS server group in the Network Policy Server (NPS) snap-in.

When you configure NPS as a RADIUS proxy, you create a new connection request policy that NPS uses to determine which connection requests to forward to other RADIUS servers. In addition, the connection request policy is configured by specifying a remote RADIUS server group that contains one or more RADIUS servers, which tells NPS where to send the connection requests that match the connection request policy.

NOTE

You can also configure a new remote RADIUS server group during the process of creating a new connection request policy.

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

To add a remote RADIUS server group

1. In Server Manager, click **Tools**, and then click **Network Policy Server** to open the NPS console.
2. In the console tree, double-click **RADIUS Clients and Servers**, right-click **Remote RADIUS Server Groups**, and then click **New**.
3. The **New Remote RADIUS Server Group** dialog box opens. In **Group name**, type a name for the remote RADIUS server group.
4. In **RADIUS Servers**, click **Add**. The **Add RADIUS Servers** dialog box opens. Type the IP address of the RADIUS server that you want to add to the group, or type the Fully Qualified Domain Name (FQDN) of the RADIUS server, and then click **Verify**.
5. In **Add RADIUS Servers**, click the **Authentication/Accounting** tab. In **Shared secret** and **Confirm shared secret**, type the shared secret. You must use the same shared secret when you configure the local computer as a RADIUS client on the remote RADIUS server.
6. If you are not using Extensible Authentication Protocol (EAP) for authentication, click **Request must contain the message authenticator attribute**. EAP uses the Message-Authenticator attribute by default.
7. Verify that the authentication and accounting port numbers are correct for your deployment.
8. If you use a different shared secret for accounting, in **Accounting**, clear the **Use the same shared secret for authentication and accounting** check box, and then type the accounting shared secret in **Shared secret** and **Confirm shared secret**.
9. If you do not want to forward network access server start and stop messages to the remote RADIUS server, clear the **Forward network access server start and stop notifications to this server** check box.

For more information about managing NPS, see [Manage Network Policy Server](#).

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Manage Certificates Used with NPS

9/1/2018 • 6 minutes to read • [Edit Online](#)

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

If you deploy a certificate-based authentication method, such as Extensible Authentication Protocol-Transport Layer Security (EAP-TLS), Protected Extensible Authentication Protocol-Transport Layer Security (PEAP-TLS), and PEAP-Microsoft Challenge Handshake Authentication Protocol version 2 (MS-CHAP v2), you must enroll a server certificate to all of your NPSs. The server certificate must:

- Meet the minimum server certificate requirements as described in [Configure Certificate Templates for PEAP and EAP Requirements](#)
- Be issued by a certification authority (CA) that is trusted by client computers. A CA is trusted when its certificate exists in the Trusted Root Certification Authorities certificate store for the current user and local computer.

The following instructions assist in managing NPS certificates in deployments where the trusted root CA is a third-party CA, such as Verisign, or is a CA that you have deployed for your public key infrastructure (PKI) by using Active Directory Certificate Services (AD CS).

Change the Cached TLS Handle Expiry

During the initial authentication processes for EAP-TLS, PEAP-TLS, and PEAP-MS-CHAP v2, the NPS caches a portion of the connecting client's TLS connection properties. The client also caches a portion of the NPS's TLS connection properties.

Each individual collection of these TLS connection properties is called a TLS handle.

Client computers can cache the TLS handles for multiple authenticators, while NPSs can cache the TLS handles of many client computers.

The cached TLS handles on the client and server allow the reauthentication process to occur more rapidly. For example, when a wireless computer reauthenticates with an NPS, the NPS can examine the TLS handle for the wireless client and can quickly determine that the client connection is a reconnect. The NPS authorizes the connection without performing full authentication.

Correspondingly, the client examines the TLS handle for the NPS, determines that it is a reconnect, and does not need to perform server authentication.

On computers running Windows 10 and Windows Server 2016, the default TLS handle expiry is 10 hours.

In some circumstances, you might want to increase or decrease the TLS handle expiry time.

For example, you might want to decrease the TLS handle expiry time in circumstances where a user's certificate is revoked by an administrator and the certificate has expired. In this scenario, the user can still connect to the network if an NPS has a cached TLS handle that has not expired. Reducing the TLS handle expiry might help prevent such users with revoked certificates from reconnecting.

NOTE

The best solution to this scenario is to disable the user account in Active Directory, or to remove the user account from the Active Directory group that is granted permission to connect to the network in network policy. The propagation of these changes to all domain controllers might also be delayed, however, due to replication latency.

Configure the TLS Handle Expiry Time on Client Computers

You can use this procedure to change the amount of time that client computers cache the TLS handle of an NPS. After successfully authenticating an NPS, client computers cache TLS connection properties of the NPS as a TLS handle. The TLS handle has a default duration of 10 hours (36,000,000 milliseconds). You can increase or decrease the TLS handle expiry time by using the following procedure.

Membership in **Administrators**, or equivalent, is the minimum required to complete this procedure.

IMPORTANT

This procedure must be performed on an NPS, not on a client computer.

To configure the TLS handle expiry time on client computers

1. On an NPS, open Registry Editor.
2. Browse to the registry key
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL
3. On the **Edit** menu, click **New**, and then click **Key**.
4. Type **ClientCacheTime**, and then press ENTER.
5. Right-click **ClientCacheTime**, click **New**, and then click **DWORD (32-bit) Value**.
6. Type the amount of time, in milliseconds, that you want client computers to cache the TLS handle of an NPS after the first successful authentication attempt by the NPS.

Configure the TLS Handle Expiry Time on NPSs

Use this procedure to change the amount of time that NPSs cache the TLS handle of client computers. After successfully authenticating an access client, NPSs cache TLS connection properties of the client computer as a TLS handle. The TLS handle has a default duration of 10 hours (36,000,000 milliseconds). You can increase or decrease the TLS handle expiry time by using the following procedure.

Membership in **Administrators**, or equivalent, is the minimum required to complete this procedure.

IMPORTANT

This procedure must be performed on an NPS, not on a client computer.

To configure the TLS handle expiry time on NPSs

1. On an NPS, open Registry Editor.
2. Browse to the registry key
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL
3. On the **Edit** menu, click **New**, and then click **Key**.

4. Type **ServerCacheTime**, and then press ENTER.
5. Right-click **ServerCacheTime**, click **New**, and then click **DWORD (32-bit) Value**.
6. Type the amount of time, in milliseconds, that you want NPSs to cache the TLS handle of a client computer after the first successful authentication attempt by the client.

Obtain the SHA-1 Hash of a Trusted Root CA Certificate

Use this procedure to obtain the Secure Hash Algorithm (SHA-1) hash of a trusted root certification authority (CA) from a certificate that is installed on the local computer. In some circumstances, such as when deploying Group Policy, it is necessary to designate a certificate by using the SHA-1 hash of the certificate.

When using Group Policy, you can designate one or more trusted root CA certificates that clients must use in order to authenticate the NPS during the process of mutual authentication with EAP or PEAP. To designate a trusted root CA certificate that clients must use to validate the server certificate, you can enter the SHA-1 hash of the certificate.

This procedure demonstrates how to obtain the SHA-1 hash of a trusted root CA certificate by using the Certificates Microsoft Management Console (MMC) snap-in.

To complete this procedure, you must be a member of the **Users** group on the local computer.

To obtain the SHA-1 hash of a trusted root CA certificate

1. In the Run dialog box or Windows PowerShell, type **mmc**, and then press ENTER. The Microsoft Management Console (MMC) opens. In the MMC, click **File**, then click **Add/Remove Snapin**. The **Add or Remove Snap-ins** dialog box opens.
2. In **Add or Remove Snap-ins**, in **Available snap-ins**, double-click **Certificates**. The Certificates snap-in wizard opens. Click **Computer account**, and then click **Next**.
3. In **Select Computer**, ensure that **Local computer (the computer this console is running on)** is selected, click **Finish**, and then click **OK**.
4. In the left pane, double-click **Certificates (Local Computer)**, and then double-click the **Trusted Root Certification Authorities** folder.
5. The **Certificates** folder is a subfolder of the **Trusted Root Certification Authorities** folder. Click the **Certificates** folder.
6. In the details pane, browse to the certificate for your trusted root CA. Double-click the certificate. The **Certificate** dialog box opens.
7. In the **Certificate** dialog box, click the **Details** tab.
8. In the list of fields, scroll to and select **Thumbprint**.
9. In the lower pane, the hexadecimal string that is the SHA-1 hash of your certificate is displayed. Select the SHA-1 hash, and then press the Windows keyboard shortcut for the Copy command (CTRL+C) to copy the hash to the Windows clipboard.
10. Open the location to which you want to paste the SHA-1 hash, correctly locate the cursor, and then press the Windows keyboard shortcut for the Paste command (CTRL+V).

For more information about certificates and NPS, see [Configure Certificate Templates for PEAP and EAP Requirements](#).

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Configure Certificate Templates for PEAP and EAP Requirements

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

All certificates that are used for network access authentication with Extensible Authentication Protocol-Transport Layer Security (EAP-TLS), Protected Extensible Authentication Protocol-Transport Layer Security (PEAP-TLS), and PEAP-Microsoft Challenge Handshake Authentication Protocol version 2 (MS-CHAP v2) must meet the requirements for X.509 certificates and work for connections that use Secure Socket Layer/Transport Level Security (SSL/TLS). Both client and server certificates have additional requirements.

IMPORTANT

This topic provides instructions for configuring certificate templates. To use these instructions, it is required that you have deployed your own Public Key Infrastructure (PKI) with Active Directory Certificate Services (AD CS).

Minimum server certificate requirements

With PEAP-MS-CHAP v2, PEAP-TLS, or EAP-TLS as the authentication method, the NPS must use a server certificate that meets the minimum server certificate requirements.

Client computers can be configured to validate server certificates by using the **Validate server certificate** option on the client computer or in Group Policy.

The client computer accepts the authentication attempt of the server when the server certificate meets the following requirements:

- The Subject name contains a value. If you issue a certificate to your server running Network Policy Server (NPS) that has a blank Subject name, the certificate is not available to authenticate your NPS. To configure the certificate template with a Subject name:
 1. Open Certificate Templates.
 2. In the details pane, right-click the certificate template that you want to change, and then click **Properties**.
 3. Click the **Subject Name** tab, and then click **Build from this Active Directory information**.
 4. In **Subject name format**, select a value other than **None**.
- The computer certificate on the server chains to a trusted root certification authority (CA) and does not fail any of the checks that are performed by CryptoAPI and that are specified in the remote access policy or network policy.
- The computer certificate for the NPS or VPN server is configured with the Server Authentication purpose in Extended Key Usage (EKU) extensions. (The object identifier for Server Authentication is 1.3.6.1.5.5.7.3.1.)
- Configure the server certificate with the required cryptography setting:
 1. Open Certificate Templates.
 2. In the details pane, right-click the certificate template that you want to change, and then click **Properties**.

3. Click the **Cryptography** tab and make sure to configure the following:
 - o **Provider Category:** Key Storage Provider
 - o **Algorithm Name:** RSA
 - o **Providers:** Microsoft Platform Crypto Provider
 - o **Minimum key size:** 2048
 - o **Hash Algorithm:** SHA2
4. Click **Next**.
- The Subject Alternative Name (SubjectAltName) extension, if used, must contain the DNS name of the server. To configure the certificate template with the Domain Name System (DNS) name of the enrolling server:
 1. Open Certificate Templates.
 2. In the details pane, right-click the certificate template that you want to change, and then click **Properties**
 3. Click the **Subject Name** tab, and then click **Build from this Active Directory information**.
 4. In **Include this information in alternate subject name**, select **DNS name**.

When using PEAP and EAP-TLS, NPSs display a list of all installed certificates in the computer certificate store, with the following exceptions:

- Certificates that do not contain the Server Authentication purpose in EKU extensions are not displayed.
- Certificates that do not contain a Subject name are not displayed.
- Registry-based and smart card-logon certificates are not displayed.

For more information, see [Deploy Server Certificates for 802.1X Wired and Wireless Deployments](#).

Minimum client certificate requirements

With EAP-TLS or PEAP-TLS, the server accepts the client authentication attempt when the certificate meets the following requirements:

- The client certificate is issued by an enterprise CA or mapped to a user or computer account in Active Directory Domain Services (AD DS).
- The user or computer certificate on the client chains to a trusted root CA, includes the Client Authentication purpose in EKU extensions (the object identifier for Client Authentication is 1.3.6.1.5.5.7.3.2), and fails neither the checks that are performed by CryptoAPI and that are specified in the remote access policy or network policy nor the Certificate object identifier checks that are specified in NPS network policy.
- The 802.1X client does not use registry-based certificates that are either smart card-logon or password-protected certificates.
- For user certificates, the Subject Alternative Name (SubjectAltName) extension in the certificate contains the user principal name (UPN). To configure the UPN in a certificate template:
 1. Open Certificate Templates.
 2. In the details pane, right-click the certificate template that you want to change, and then click **Properties**.
 3. Click the **Subject Name** tab, and then click **Build from this Active Directory information**.
 4. In **Include this information in alternate subject name**, select **User principal name (UPN)**.
- For computer certificates, the Subject Alternative Name (SubjectAltName) extension in the certificate must contain the fully qualified domain name (FQDN) of the client, which is also called the *DNS name*. To configure this name in the certificate template:

1. Open Certificate Templates.
2. In the details pane, right-click the certificate template that you want to change, and then click **Properties**.
3. Click the **Subject Name** tab, and then click **Build from this Active Directory information**.
4. In **Include this information in alternate subject name**, select **DNS name**.

With PEAP-TLS and EAP-TLS, clients display a list of all installed certificates in the Certificates snap-in, with the following exceptions:

- Wireless clients do not display registry-based and smart card-logon certificates.
- Wireless clients and VPN clients do not display password-protected certificates.
- Certificates that do not contain the Client Authentication purpose in EKU extensions are not displayed.

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Manage NPSs

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the topics in this section to manage NPSs.

NOTE

For additional Network Policy Server documentation, you can use the following library sections.

- [Getting Started with Network Policy Server](#)
- [Deploy Network Policy Server](#)

This section contains the following topics.

- [Configure NPS on a Multihomed Computer](#)
- [Configure NPS UDP Port Information](#)
- [Disable NAS Notification Forwarding](#)
- [Export an NPS Configuration for Import on Another Server](#)
- [Increase Concurrent Authentications Processed by NPS](#)
- [Install Network Policy Server](#)
- [NPS Proxy Server Load Balancing](#)
- [Register an NPS in an Active Directory Domain](#)
- [Unregister an NPS from an Active Directory Domain](#)
- [Use Regular Expressions in NPS](#)
- [Verify Configuration After NPS Changes](#)

Configure NPS on a Multihomed Computer

9/1/2018 • 3 minutes to read • [Edit Online](#)

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

You can use this topic to configure an NPS with multiple network adapters.

When you use multiple network adapters in a server running Network Policy Server (NPS), you can configure the following:

- The network adapters that do and do not send and receive Remote Authentication Dial-In User Service (RADIUS) traffic.
- On a per-network adapter basis, whether NPS monitors RADIUS traffic on Internet Protocol version 4 (IPv4), IPv6, or both IPv4 and IPv6.
- The UDP port numbers over which RADIUS traffic is sent and received on a per-protocol (IPv4 or IPv6), per-network adapter basis.

By default, NPS listens for RADIUS traffic on ports 1812, 1813, 1645, and 1646 for both IPv6 and IPv4 for all installed network adapters. Because NPS automatically uses all network adapters for RADIUS traffic, you only need to specify the network adapters that you want NPS to use for RADIUS traffic when you want to prevent NPS from using a specific network adapter.

NOTE

If you uninstall either IPv4 or IPv6 on a network adapter, NPS does not monitor RADIUS traffic for the uninstalled protocol.

On an NPS that has multiple network adapters installed, you might want to configure NPS to send and receive RADIUS traffic only on the adapters you specify.

For example, one network adapter installed in the NPS might lead to a network segment that does not contain RADIUS clients, while a second network adapter provides NPS with a network path to its configured RADIUS clients. In this scenario, it is important to direct NPS to use the second network adapter for all RADIUS traffic.

In another example, if your NPS has three network adapters installed, but you only want NPS to use two of the adapters for RADIUS traffic, you can configure port information for the two adapters only. By excluding port configuration for the third adapter, you prevent NPS from using the adapter for RADIUS traffic.

Using a network adapter

To configure NPS to listen for and send RADIUS traffic on a network adapter, use the following syntax on the Properties dialog box of Network Policy Server in the NPS console:

- IPv4 traffic syntax: `IPAddress:UDPport`, where `IPAddress` is the IPv4 address that is configured on the network adapter over which you want to send RADIUS traffic, and `UDPport` is the RADIUS port number that you want to use for RADIUS authentication or accounting traffic.
- IPv6 traffic syntax: `[IPv6Address] : UDPport`, where the brackets around `IPv6Address` are required, `IPv6Address` is the IPv6 address that is configured on the network adapter over which you want to send RADIUS traffic, and `UDPport` is the RADIUS port number that you want to use for RADIUS authentication or accounting traffic.

The following characters can be used as delimiters for configuring IP address and UDP port information:

- Address/port delimiter: colon (:)
- Port delimiter: comma (,)
- Interface delimiter: semicolon (;)

Configuring network access servers

Make sure that your network access servers are configured with the same RADIUS UDP port numbers that you configure on your NPSs. The RADIUS standard UDP ports defined in RFCs 2865 and 2866 are 1812 for authentication and 1813 for accounting; however, some access servers are configured by default to use UDP port 1645 for authentication requests and UDP port 1646 for accounting requests.

IMPORTANT

If you do not use the RADIUS default port numbers, you must configure exceptions on the firewall for the local computer to allow RADIUS traffic on the new ports. For more information, see [Configure Firewalls for RADIUS Traffic](#).

Configure the multihomed NPS

You can use the following procedure to configure your multihomed NPS.

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

To specify the network adapter and UDP ports that NPS uses for RADIUS traffic

1. In Server manager, click **Tools**, and then click **Network Policy Server** to open the NPS console.
2. Right-click **Network Policy Server**, and then click **Properties**.
3. Click the **Ports** tab, and prepend the IP address for the network adapter you want to use for RADIUS traffic to the existing port numbers. For example, if you want to use the IP address 192.168.1.2 and RADIUS ports 1812 and 1645 for authentication requests, change the port setting from **1812,1645** to **192.168.1.2:1812,1645**. If your RADIUS authentication and RADIUS accounting UDP ports are different from the default values, change the port settings accordingly.
4. To use multiple port settings for authentication or accounting requests, separate the port numbers with commas.

For more information about NPS UDP ports, see [Configure NPS UDP Port Information](#)

For more information about NPS, see [Network Policy Server](#)

Configure NPS UDP Port Information

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the following procedure to configure the ports that Network Policy Server (NPS) uses for Remote Authentication Dial-In User Service (RADIUS) authentication and accounting traffic.

By default, NPS listens for RADIUS traffic on ports 1812, 1813, 1645, and 1646 for both Internet Protocol version 6 (IPv6) and IPv4 for all installed network adapters.

NOTE

If you uninstall either IPv4 or IPv6 on a network adapter, NPS does not monitor RADIUS traffic for the uninstalled protocol.

The port values of 1812 for authentication and 1813 for accounting are RADIUS standard ports defined by the Internet Engineering Task Force (IETF) in RFCs 2865 and 2866. However, by default, many access servers use ports 1645 for authentication requests and 1646 for accounting requests. No matter which port numbers you decide to use, make sure that NPS and your access server are configured to use the same ones.

[IMPORTANT] If you do not use the RADIUS default port numbers, you must configure exceptions on the firewall for the local computer to allow RADIUS traffic on the new ports. For more information, see [Configure Firewalls for RADIUS Traffic](#).

Membership in **Domain Admins**, or equivalent, is the minimum required to complete this procedure.

To configure NPS UDP port information

1. Open the NPS console.
2. Right-click **Network Policy Server**, and then click **Properties**.
3. Click the **Ports** tab, and then examine the settings for ports. If your RADIUS authentication and RADIUS accounting UDP ports vary from the default values provided (1812 and 1645 for authentication, and 1813 and 1646 for accounting), type your port settings in **Authentication** and **Accounting**.
4. To use multiple port settings for authentication or accounting requests, separate the port numbers with commas.

For more information about managing NPS, see [Manage Network Policy Server](#).

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Disable NAS Notification Forwarding in NPS

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this procedure to disable the forwarding of start and stop messages from network access servers (NASs) to members of a remote RADIUS server group configured in NPS.

When you have remote RADIUS server groups configured and, in NPS **Connection Request Policies**, you clear the **Forward accounting requests to this remote RADIUS server group** check box, these groups are still sent NAS start and stop notification messages.

This creates unnecessary network traffic. To eliminate this traffic, disable NAS notification forwarding for individual servers in each remote RADIUS server group.

To complete this procedure, you must be a member of the **Administrators** group.

To disable NAS notification forwarding

1. In Server Manager, click **Tools**, and then click **Network Policy Server**. The NPS console opens.
2. In the NPS console, double-click **RADIUS Clients and Servers**, click **Remote RADIUS Server Groups**, and then double-click the remote RADIUS server group that you want to configure. The remote RADIUS server group **Properties** dialog box opens.
3. Double-click the group member that you want to configure, and then click the **Authentication/Accounting** tab.
4. In **Accounting**, clear the **Forward network access server start and stop notifications to this server** check box, and then click **OK**.
5. Repeat steps 3 and 4 for all group members that you want to configure.

For more information about managing NPS, see [Manage Network Policy Server](#).

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Export an NPS Configuration for Import on Another Server

6/27/2018 • 3 minutes to read • [Edit Online](#)

Applies To: Windows Server 2016

You can export the entire NPS configuration — including RADIUS clients and servers, network policy, connection request policy, registry, and logging configuration — from one NPS for import on another NPS.

Use one of the following tools to export the NPS configuration:

- In Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012, you can use Netsh, or you can use Windows PowerShell.
- In Windows Server 2008 R2 and Windows Server 2008, use Netsh.

IMPORTANT

Do not use this procedure if the source NPS database has a higher version number than the version number of the destination NPS database. You can view the version number of the NPS database from the display of the **netsh nps show config** command.

Because NPS configurations are not encrypted in the exported XML file, sending it over a network might pose a security risk, so take precautions when moving the XML file from the source server to the destination servers. For example, add the file to an encrypted, password protected archive file before moving the file. In addition, store the file in a secure location to prevent malicious users from accessing it.

NOTE

If SQL Server logging is configured on the source NPS, SQL Server logging settings are not exported to the XML file. After you import the file on another NPS, you must manually configure SQL Server logging.

Export and Import the NPS configuration by using Windows PowerShell

For Windows Server 2012 and later operating system versions, you can export the NPS configuration using Windows PowerShell.

The command syntax for exporting the NPS configuration is as follows.

```
Export-NpsConfiguration -Path <filename>
```

The following table lists parameters for the **Export-NpsConfiguration** cmdlet in Windows PowerShell.

Parameters in bold are required.

PARAMETER	DESCRIPTION
Path	Specifies the name and location of the XML file to which you want to export the NPS configuration.

Administrative credentials

To complete this procedure, you must be a member of the Administrators group.

Export Example

In the following example, the NPS configuration is exported to an XML file located on the local drive. To run this command, run Windows PowerShell as Administrator on the source NPS, type the following command, and press Enter.

```
Export-NpsConfiguration -Path c:\config.xml
```

For more information, see [Export-NpsConfiguration](#).

After you have exported the NPS configuration, copy the XML file to the destination server.

The command syntax for importing the NPS configuration on the destination server is as follows.

```
Import-NpsConfiguration [-Path] <String> [ <CommonParameters>]
```

Import Example

The following command imports settings from the file named C:\Npsconfig.xml to NPS. To run this command, run Windows PowerShell as Administrator on the destination NPS, type the following command, and press Enter.

```
PS C:\> Import-NpsConfiguration -Path "C:\Npsconfig.xml"
```

For more information, see [Import-NpsConfiguration](#).

Export and Import the NPS configuration by using Netsh

You can use Network Shell (Netsh) to export the NPS configuration by using the **netsh nps export** command.

When the **netsh nps import** command is run, NPS is automatically refreshed with the updated configuration settings. You do not need to stop NPS on the destination computer to run the **netsh nps import** command, however if the NPS console or NPS MMC snap-in is open during the configuration import, changes to the server configuration are not visible until you refresh the view.

NOTE

When you use the **netsh nps export** command, you are required to provide the command parameter **exportPSK** with the value **YES**. This parameter and value explicitly state that you understand that you are exporting the NPS configuration, and that the exported XML file contains unencrypted shared secrets for RADIUS clients and members of remote RADIUS server groups.

Administrative credentials

To complete this procedure, you must be a member of the Administrators group.

To copy an NPS configuration to another NPS using Netsh commands

1. On the source NPS, open **Command Prompt**, type **netsh**, and then press Enter.
2. At the **netsh** prompt, type **nps**, and then press Enter.
3. At the **netsh nps** prompt, type **export filename="path\file.xml" exportPSK=YES**, where *path* is the folder location where you want to save the NPS configuration file, and *file* is the name of the XML file that you want to save. Press Enter.

This stores configuration settings (including registry settings) in an XML file. The path can be relative or absolute, or it can be a Universal Naming Convention (UNC) path. After you press Enter, a message appears indicating whether the export to file was successful.

1. Copy the file you created to the destination NPS.
2. At a command prompt on the destination NPS, type **netsh nps import filename="*path\file.xml*"**, and then press Enter. A message appears indicating whether the import from the XML file was successful.

For more information about netsh, see [Network Shell \(Netsh\)](#).

Increase Concurrent Authentications Processed by NPS

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic for instructions on configuring Network Policy Server concurrent authentications.

If you installed Network Policy Server (NPS) on a computer other than a domain controller and the NPS is receiving a large number of authentication requests per second, you can improve NPS performance by increasing the number of concurrent authentications allowed between the NPS and the domain controller.

To do this, you must edit the following registry key:

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters`

Add a new value named **MaxConcurrentApi** and assign to it a value from 2 through 5.

Caution

If you assign a value to **MaxConcurrentApi** that is too high, your NPS might place an excessive load on your domain controller.

For more information about managing NPS, see [Manage Network Policy Server](#).

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Install Network Policy Server

6/27/2018 • 2 minutes to read • [Edit Online](#)

You can use this topic to install Network Policy Server (NPS) by using either Windows PowerShell or the Add Roles and Features Wizard. NPS is a role service of the Network Policy and Access Services server role.

NOTE

By default, NPS listens for RADIUS traffic on ports 1812, 1813, 1645, and 1646 on all installed network adapters. If Windows Firewall with Advanced Security is enabled when you install NPS, firewall exceptions for these ports are automatically created during the installation process for both Internet Protocol version 6 (IPv6) and IPv4 traffic. If your network access servers are configured to send RADIUS traffic over ports other than these defaults, remove the exceptions created in Windows Firewall with Advanced Security during NPS installation, and create exceptions for the ports that you do use for RADIUS traffic.

Administrative Credentials

To complete this procedure, you must be a member of the **Domain Admins** group.

To install NPS by using Windows PowerShell

To perform this procedure by using Windows PowerShell, run Windows PowerShell as Administrator, type the following command, and then press ENTER.

```
Install-WindowsFeature NPAS -IncludeManagementTools
```

To install NPS by using Server Manager

1. On NPS1, in Server Manager, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features Wizard opens.
2. In **Before You Begin**, click **Next**.

NOTE

The **Before You Begin** page of the Add Roles and Features Wizard is not displayed if you have previously selected **Skip this page by default** when the Add Roles and Features Wizard was run.

3. In **Select Installation Type**, ensure that **Role-Based or feature-based installation** is selected, and then click **Next**.
4. In **Select destination server**, ensure that **Select a server from the server pool** is selected. In **Server Pool**, ensure that the local computer is selected. Click **Next**.
5. In **Select Server Roles**, in **Roles**, select **Network Policy and Access Services**. A dialog box opens asking if it should add features that are required for Network Policy and Access Services. Click **Add Features**, and then click **Next**.
6. In **Select features**, click **Next**, and in **Network Policy and Access Services**, review the information that is provided, and then click **Next**.
7. In **Select role services**, click **Network Policy Server**. In **Add features that are required for Network Policy Server**, click **Add Features**. Click **Next**.

8. In **Confirm installation selections**, click **Restart the destination server automatically if required**.

When you are prompted to confirm this selection, click **Yes**, and then click **Install**. The Installation progress page displays status during the installation process. When the process completes, the message "Installation succeeded on *ComputerName*" is displayed, where *ComputerName* is the name of the computer upon which you installed Network Policy Server. Click **Close**.

For more information, see [Manage NPSs](#).

NPS Proxy Server Load Balancing

6/27/2018 • 5 minutes to read • [Edit Online](#)

Applies To: Windows Server 2016

Remote Authentication Dial-In User Service (RADIUS) clients, which are network access servers such as virtual private network (VPN) servers and wireless access points, create connection requests and send them to RADIUS servers such as NPS. In some cases, an NPS might receive too many connection requests at one time, resulting in degraded performance or an overload. When an NPS is overloaded, it is a good idea to add more NPSs to your network and to configure load balancing. When you evenly distribute incoming connection requests among multiple NPSs to prevent the overloading of one or more NPSs, it is called load balancing.

Load balancing is particularly useful for:

- Organizations that use Extensible Authentication Protocol-Transport Layer Security (EAP-TLS) or Protected Extensible Authentication Protocol (PEAP)-TLS for authentication. Because these authentication methods use certificates for server authentication and for either user or client computer authentication, the load on RADIUS proxies and servers is heavier than when password-based authentication methods are used.
- Organizations that need to sustain continuous service availability.
- Internet service providers (ISPs) that outsource VPN access for other organizations. The outsourced VPN services can generate a large volume of authentication traffic.

There are two methods you can use to balance the load of connection requests sent to your NPSs:

- Configure your network access servers to send connection requests to multiple RADIUS servers. For example, if you have 20 wireless access points and two RADIUS servers, configure each access point to send connection requests to both RADIUS servers. You can load balance and provide failover at each network access server by configuring the access server to send connection requests to multiple RADIUS servers in a specified order of priority. This method of load balancing is usually best for small organizations that do not deploy a large number of RADIUS clients.
- Use NPS configured as a RADIUS proxy to load balance connection requests between multiple NPSs or other RADIUS servers. For example, if you have 100 wireless access points, one NPS proxy, and three RADIUS servers, you can configure the access points to send all traffic to the NPS proxy. On the NPS proxy, configure load balancing so that the proxy evenly distributes the connection requests between the three RADIUS servers. This method of load balancing is best for medium and large organizations that have many RADIUS clients and servers.

In many cases, the best approach to load balancing is to configure RADIUS clients to send connection requests to two NPS proxy servers, and then configure the NPS proxies to load balance among RADIUS servers. This approach provides both failover and load balancing for NPS proxies and RADIUS servers.

RADIUS server priority and weight

During the NPS proxy configuration process, you can create remote RADIUS server groups and then add RADIUS servers to each group. To configure load balancing, you must have more than one RADIUS server per remote RADIUS server group. While adding group members, or after creating a RADIUS server as a group member, you can access the Add RADIUS server dialog box to configure the following items on the Load Balancing tab:

- **Priority.** Priority specifies the order of importance of the RADIUS server to the NPS proxy server. Priority level must be assigned a value that is an integer, such as 1, 2, or 3. The lower the number, the higher priority

the NPS proxy gives to the RADIUS server. For example, if the RADIUS server is assigned the highest priority of 1, the NPS proxy sends connection requests to the RADIUS server first; if servers with priority 1 are not available, NPS then sends connection requests to RADIUS servers with priority 2, and so on. You can assign the same priority to multiple RADIUS servers, and then use the Weight setting to load balance between them.

- **Weight.** NPS uses this Weight setting to determine how many connection requests to send to each group member when the group members have the same priority level. Weight setting must be assigned a value between 1 and 100, and the value represents a percentage of 100 percent. For example, if the remote RADIUS server group contains two members that both have a priority level of 1 and a weight rating of 50, the NPS proxy forwards 50 percent of the connection requests to each RADIUS server.
- **Advanced settings.** These failover settings provide a way for NPS to determine whether the remote RADIUS server is unavailable. If NPS determines that a RADIUS server is unavailable, it can start sending connection requests to other group members. With these settings you can configure the number of seconds that the NPS proxy waits for a response from the RADIUS server before it considers the request dropped; the maximum number of dropped requests before the NPS proxy identifies the RADIUS server as unavailable; and the number of seconds that can elapse between requests before the NPS proxy identifies the RADIUS server as unavailable.

Configure NPS proxy load balancing

Before configuring load balancing, create a deployment plan that includes how many remote RADIUS server groups you require, which servers are members of each particular group, and the Priority and Weight setting for each server.

NOTE

The steps that follow assume that you have already deployed and configured RADIUS servers.

To configure NPS to act as a proxy server and forward connection requests from RADIUS clients to remote RADIUS servers, you must take the following actions:

1. Deploy your RADIUS clients (VPN servers, dial-up servers, Terminal Services Gateway servers, 802.1X authenticating switches, and 802.1X wireless access points) and configure them to send connection requests to your NPS proxy servers.
2. On the NPS proxy, configure the network access servers as RADIUS clients. For more information, see [Configure RADIUS Clients](#).
3. On the NPS proxy, create one or more remote RADIUS server groups. During this process, add RADIUS servers to the remote RADIUS server groups. For more information, see [Configure Remote RADIUS Server Groups](#).
4. On the NPS proxy, for each RADIUS server that you add to a remote RADIUS server group, click the RADIUS server **Load Balancing** tab, and then configure **Priority**, **Weight**, and **Advanced settings**.
5. On the NPS proxy, configure connection request policies to forward authentication and accounting requests to remote RADIUS server groups. You must create one connection request policy per remote RADIUS server group. For more information, see [Configure Connection Request Policies](#).

Register an NPS in an Active Directory Domain

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to register a server running Network Policy Server in Windows Server 2016 in the NPS default domain or in another domain.

Register an NPS in its Default Domain

You can use this procedure to register an NPS in the domain where the server is a domain member.

NPSs must be registered in Active Directory so that they have permission to read the dial-in properties of user accounts during the authorization process. Registering an NPS adds the server to the **RAS and IAS Servers** group in Active Directory.

Membership in **Administrators**, or equivalent, is the minimum required to perform these procedures.

To register an NPS in its default domain

1. On the NPS, in Server Manager, click **Tools**, and then click **Network Policy Server**. The Network Policy Server console opens.
2. Right-click **NPS (Local)**, and then click **Register Server in Active Directory**. The **Network Policy Server** dialog box opens.
3. In **Network Policy Server**, click **OK**, and then click **OK** again.

Register an NPS in Another Domain

To provide an NPS with permission to read the dial-in properties of user accounts in Active Directory, the NPS must be registered in the domain where the accounts reside.

You can use this procedure to register an NPS in a domain where the NPS is not a domain member.

Membership in **Administrators**, or equivalent, is the minimum required to perform these procedures.

To register an NPS in another domain

1. On the domain controller, in Server Manager, click **Tools**, and then click **Active Directory Users and Computers**. The Active Directory Users and Computers console opens.
2. In the console tree, navigate to the domain where you want the NPS to read user account information, and then click the **Users** folder.
3. In the details pane, right-click **RAS and IAS Servers**, and then click **Properties**. The **RAS and IAS Servers Properties** dialog box opens.
4. In the **RAS and IAS Servers Properties** dialog box, click the **Members** tab, add each of the NPSs that you want to register in the domain, and then click **OK**.

To register an NPS in another domain by using Netsh commands for NPS

1. Open Command Prompt or windows PowerShell.
2. Type the following at the command prompt: **netsh nps add registeredserver domain server**, and then press ENTER.

NOTE

In the preceding command, *domain* is the DNS domain name of the domain where you want to register the NPS, and *server* is the name of the NPS computer.

Unregister an NPS from an Active Directory Domain

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

In the process of managing your NPS deployment, you might find it useful to move an NPS to another domain, to replace an NPS, or to retire an NPS.

When you move or decommission an NPS, you can unregister the NPS in the Active Directory domains where the NPS has permission to read the properties of user accounts in Active Directory.

Membership in **Administrators**, or equivalent, is the minimum required to perform these procedures.

To unregister an NPS

1. On the domain controller, in Server Manager, click **Tools**, and then click **Active Directory Users and Computers**. The Active Directory Users and Computers console opens.
2. Click **Users**, and then double-click **RAS and IAS servers**.
3. Click the **Members** tab, and then select the NPS that you want to unregister.
4. Click **Remove**, click **Yes**, and then click **OK**.

Use Regular Expressions in NPS

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

This topic explains the use of regular expressions for pattern matching in NPS in Windows Server 2016. You can use this syntax to specify the conditions of network policy attributes and RADIUS realms.

Pattern-matching reference

You can use the following table as a reference source when creating regular expressions with pattern-matching syntax.

CHARACTER	DESCRIPTION	EXAMPLE
\	Marks the next character as a character to match.	/n/ matches the character "n". The sequence /\n/ matches a line feed or newline character.
^	Matches the beginning of the input or line.	
\$	Matches the end of the input or line.	
*	Matches the preceding character zero or more times.	/zo*/ matches either "z" or "zoo."
+	Matches the preceding character one or more times.	/zo+/ matches "zoo" but not "z."
?	Matches the preceding character zero or one times.	/a?ve?/ matches the "ve" in "never."
.	Matches any single character except a newline character.	
(pattern)	Matches "pattern" and remembers the match.	To match () (parentheses), use "\(" or "\)".
'x	y `	Matches either x or y.
{ n }	Matches exactly n times (n is a non-negative integer).	/o{2}/ does not match the "o" in "Bob," but matches the first two instances of the letter o in "fooooood."
{ n ,}	Matches at least n times (n is a non-negative integer).	/o{2,}/ does not match the "o" in "Bob" but matches all of the instances of the letter o in "fooooood." /o{1,}/ is equivalent to /o+/.

CHARACTER	DESCRIPTION	EXAMPLE
{ n , m }	Matches at least n and at most m times (m and n are non-negative integers).	/o{1,3}/ matches the first three instances of the letter o in "foooooood."
[xyz]	Matches any one of the enclosed characters (a character set).	/[abc]/ matches the "a" in "plain."
[^ xyz]	Matches any characters that are not enclosed (a negative character set).	/[^abc]/ matches the "p" in "plain."
\b	Matches a word boundary (for example, a space).	/ea*r\b/ matches the "er" in "never early."
\B	Matches a nonword boundary.	/ea*r\B/ matches the "ear" in "never early."
\d	Matches a digit character (equivalent to digits from 0 to 9).	
\D	Matches a nondigit character (equivalent to [^0-9]).	
\f	Matches a form feed character.	
\n	Matches a line feed character.	
\r	Matches a carriage return character.	
\s	Matches any white space character including space, tab, and form feed (equivalent to [\f\n\r\t\v]).	
\S	Matches any non-white space character (equivalent to [^ \f\n\r\t\v]).	
\t	Matches a tab character.	
\v	Matches a vertical tab character.	
\w	Matches any word character, including underscore (equivalent to [A-Za-z0-9_]).	
\W	Matches any non-word character, excluding underscore (equivalent to [^A-Za-z0-9_]).	
\ num	Refers to remembered matches (?num , where num is a positive integer). This option can be used only in the Replace text box when configuring attribute manipulation.	\1 replaces what is stored in the first remembered match.

CHARACTER	DESCRIPTION	EXAMPLE
/ n /	Allows the insertion of ASCII codes into regular expressions (<code>?n</code> , where n is an octal, hexadecimal, or decimal escape value).	

Examples for network policy attributes

The following examples describe the use of the pattern-matching syntax to specify network policy attributes:

- To specify all phone numbers within the 899 area code, the syntax is:

`899.*`

- To specify a range of IP addresses that begin with 192.168.1, the syntax is:

`192\.168\.1\..+`

Examples for manipulation of the realm name in the User Name attribute

The following examples describe the use of the pattern-matching syntax to manipulate realm names for the User Name attribute, which is located on the **Attribute** tab in the properties of a connection request policy.

To remove the realm portion of the User Name attribute

In an outsourced dial-up scenario in which an Internet service provider (ISP) routes connection requests to an organization NPS, the ISP RADIUS proxy might require a realm name to route the authentication request. However, the NPS might not recognize the realm name portion of the user name. Therefore, the realm name must be removed by the ISP RADIUS proxy before it is forwarded to the organization NPS.

- Find: `@microsoft\com`
- Replace:

To replace `user@example.microsoft.com` with `*example.microsoft.com\user*`

- Find: `(.*)@(.*)`
- Replace: `$2\$1`

To replace `domain\user` with `*specific_domain\user*`

- Find: `(.*)\\(.*)`
- Replace: `specific_domain \$2`

To replace `user` with `*user@specific_domain*`

- Find: `$`
- Replace: `@specific_domain`

Example for RADIUS message forwarding by a proxy server

You can create routing rules that forward RADIUS messages with a specified realm name to a set of RADIUS servers when NPS is used as a RADIUS proxy. Following is a recommended syntax for routing requests based on realm name.

- **NetBIOS name:** WCOAST

- **Pattern:** ^wcoast\\

In the following example, wcoast.microsoft.com is a unique user principal name (UPN) suffix for the DNS or Active Directory domain wcoast.microsoft.com. Using the supplied pattern, the NPS proxy can route messages based on domain NetBIOS name or UPN suffix.

- **NetBIOS name:** WCOAST

- **UPN suffix:** wcoast.microsoft.com

- **Pattern:** ^wcoast\\|@wcoast\.microsoft\.com\$

For more information about managing NPS, see [Manage Network Policy Server](#).

For more information about NPS, see [Network Policy Server \(NPS\)](#).

Verify Configuration After NPS Changes

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

You can use this topic to verify NPS configuration after an IP address or name change to the server.

Verify Configuration After an NPS IP Address Change

There might be circumstances where you need to change the IP address of an NPS or proxy, such as when you move the server to a different IP subnet.

If you change an NPS or proxy IP address, it is necessary to reconfigure portions of your NPS deployment.

Use the following general guidelines to assist you in verifying that an IP address change does not interrupt network access authentication, authorization, or accounting on your network for NPS RADIUS servers and RADIUS proxy servers.

You must be a member of **Administrators**, or equivalent, to perform these procedures.

To verify configuration after an NPS IP address change

1. Reconfigure all RADIUS clients, such as wireless access points and VPN servers, with the new IP address of the NPS.
2. If the NPS is a member of a remote RADIUS server group, reconfigure the NPS proxy with the new IP address of the NPS.
3. If you have configured the NPS to use SQL Server logging, verify that connectivity between the computer running SQL Server and the NPS is still functioning properly.
4. If you have deployed IPsec to secure RADIUS traffic between your NPS and an NPS proxy or other servers or devices, reconfigure the IPsec policy or the connection security rule in Windows Firewall with Advanced Security to use the new IP address of the NPS.
5. If the NPS is multihomed and you have configured the server to bind to a specific network adapter, reconfigure NPS port settings with the new IP address.

To verify configuration after an NPS proxy IP address change

1. Reconfigure all RADIUS clients, such as wireless access points and VPN servers, with the new IP address of the NPS proxy.
2. If the NPS proxy is multihomed and you have configured the proxy to bind to a specific network adapter, reconfigure NPS port settings with the new IP address.
3. Reconfigure all members of all remote RADIUS server groups with the proxy server IP address. To accomplish this task, at each NPS that has the NPS proxy configured as a RADIUS client:
 - a. Double-click **NPS (Local)**, double-click **RADIUS Clients and Servers**, click **RADIUS Clients**, and then in the details pane, double-click the RADIUS client that you want to change.
 - b. In RADIUS client **Properties**, in **Address (IP or DNS)**, type the new IP address of the NPS proxy.
4. If you have configured the NPS proxy to use SQL Server logging, verify that connectivity between the computer running SQL Server and the NPS proxy is still functioning properly.

Verify Configuration After Renaming an NPS

There might be circumstances when you need to change the name of an NPS or proxy, such as when you redesign the naming conventions for your servers.

If you change an NPS or proxy name, it is necessary to reconfigure portions of your NPS deployment.

Use the following general guidelines to assist you in verifying that a server name change does not interrupt network access authentication, authorization, or accounting.

You must be a member of **Administrators**, or equivalent, to perform this procedure.

To verify configuration after an NPS or proxy name change

1. If the NPS is a member of a remote RADIUS server group and the group is configured with computer names rather than IP addresses, reconfigure the remote RADIUS server group with the new NPS name.
2. If certificate-based authentication methods are deployed at the NPS, the name change invalidates the server certificate. You can request a new certificate from the certification authority (CA) administrator or, if the computer is a domain member computer and you autoenroll certificates to domain members, you can refresh Group Policy to obtain a new certificate through autoenrollment. To refresh Group Policy:
 - a. Open Command Prompt or Windows PowerShell.
 - b. Type **gpupdate**, and then press ENTER.
3. After you have a new server certificate, request that the CA administrator revoke the old certificate.

After the old certificate is revoked, NPS continues to use it until the old certificate expires. By default, the old certificate remains valid for a maximum time of one week and 10 hours. This time period might be different depending on whether the Certificate Revocation List (CRL) expiry and the Transport Layer Security (TLS) cache time expiry have been modified from their defaults. The default CRL expiry is one week; the default TLS cache time expiry is 10 hours.

If you want to configure NPS to use the new certificate immediately, however, you can manually reconfigure network policies with the new certificate.

4. After the old certificate expires, NPS automatically begins using the new certificate.
5. If you have configured the NPS to use SQL Server logging, verify that connectivity between the computer running SQL Server and the NPS is still functioning properly.

Network Policy Server user data collection

5/24/2018 • 2 minutes to read • [Edit Online](#)

This document explains how to find user information collected by the Network Policy Server (NPS) in the event you would like to remove it.

NOTE

If you're interested in viewing or deleting personal data, please review Microsoft's guidance in the [Windows Data Subject Requests for the GDPR site](#). If you're looking for general information about GDPR, see the [GDPR section of the Service Trust portal](#).

Information Collected by NPS

- Timestamp
- Event Timestamp
- Username
- Full Qualified Username
- Client IP Address
- Client Vendor
- Client Friendly Name
- Authentication Type
- Numerous other fields concerning the RADIUS protocol

Gather data from NPS

If accounting data is enabled and configured, then records of a user's NPS authentication attempts can be obtained from SQL Server or the log files depending on the configuration.

If accounting data is configured for SQL Server, query for all records WHERE User_Name = '`<username>`' .

If accounting data is configured for a log file, then search the log file for the `<username>` to find all log entries.

Network Policy and Access Services event log entries are considered duplicative to the accounting data and don't need to be collected.

If accounting data is not enabled, then records of a user's NPS authentication attempts can be obtained from the Network Policy and Access Services event log by searching for the `<username>` .

Manage NPS Templates

9/1/2018 • 3 minutes to read • [Edit Online](#)

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

You can use Network Policy Server (NPS) templates to create configuration elements, such as Remote Authentication Dial-In User Service (RADIUS) clients or shared secrets, that you can reuse on the local NPS and export for use on other NPSs.

Templates Management provides a node in the NPS console where you can create, modify, delete, duplicate, and view the use of NPS templates. NPS templates are designed to reduce the amount of time and cost that it takes to configure NPS on one or more servers.

The following NPS template types are available for configuration in Templates Management.

- **Shared Secrets.** This template type makes it possible for you to specify a shared secret that you can reuse (by selecting the template in the appropriate location in the NPS console) when you configure RADIUS clients and servers.
- **RADIUS Clients.** This template type makes it possible for you to configure RADIUS client settings that you can reuse by selecting the template in the appropriate location in the NPS console.
- **Remote RADIUS Servers.** This template makes it possible for you to configure remote RADIUS server settings that you can reuse by selecting the template in the appropriate location in the NPS console.
- **IP Filters.** This template makes it possible for you to create Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) filters that you can reuse (by selecting the template in the appropriate location in the NPS console) when you configure network policies.

Create an NPS Template

Configuring a template is different than configuring the NPS directly. Creating a template does not affect the NPS's functionality. It is only when you select the template in the appropriate location in the NPS console and apply the template that the template affects the NPS functionality.

For example, if you configure a RADIUS client in the NPS console under **RADIUS Clients and Servers**, you alter the NPS configuration and take one step in configuring NPS to communicate with one of your network access servers. (The next step is to configure the network access server (NAS) to communicate with NPS.)

However, if you configure a new **RADIUS Clients** template in the NPS console under **Templates Management** rather than creating a new RADIUS client under **RADIUS Clients and Servers**, you have created a template, but you have not altered the NPS functionality yet. To alter the NPS functionality, you must apply the template from the correct location in the NPS console.

The following procedure provides instructions on how to create a new template.

Membership in **Administrators**, or equivalent, is the minimum required to complete this procedure.

To create an NPS template

1. On the NPS, in Server Manager, click **Tools**, and then click **Network Policy Server**. The NPS console opens.
2. In the NPS console, expand **Templates Management**, right-click a template type, such as **RADIUS**

Clients, and then click **New**.

3. A new template properties dialog box opens that you can use to configure your template.

Apply an NPS Template

You can use a template that you have created in **Templates Management** by navigating to a location in the NPS console where you can apply the template. For example, if you want to apply a Shared Secrets template to a RADIUS client configuration, you can use the following procedure.

Membership in **Administrators**, or equivalent, is the minimum required to complete this procedure.

To apply an NPS template

1. On the NPS, in Server Manager, click **Tools**, and then click **Network Policy Server**. The NPS console opens.
2. In the NPS console, expand **RADIUS Clients and Servers**, and then expand **RADIUS Clients**.
3. In **RADIUS Clients**, in the details pane, right-click the RADIUS client to which you want to apply the NPS template, and then click **Properties**.
 1. In the properties dialog box for the RADIUS client, in **Select an existing Shared Secrets template**, select the template that you want to apply from the list of templates.

Export or Import NPS Templates

You can export templates for use on other NPSs, or you can import templates into **Templates Management** for use on the local computer.

Membership in **Administrators**, or equivalent, is the minimum required to complete this procedure.

To export or import NPS templates

1. To export NPS templates, in the NPS console, right-click **Templates Management**, and then click **Export Templates to a File**.
2. To import NPS templates, in the NPS console, right-click **Templates Management**, and then click **Import Templates from a Computer** or **Import Templates from a File**.

Network Shell (Netsh)

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

Network shell (netsh) is a command-line utility that allows you to configure and display the status of various network communications server roles and components after they are installed on computers running Windows Server 2016.

Some client technologies, such as Dynamic Host Configuration Protocol (DHCP) client and BranchCache, also provide netsh commands that allow you to configure client computers that are running Windows 10.

In most cases, netsh commands provide the same functionality that is available when you use the Microsoft Management Console (MMC) snap-in for each networking server role or networking feature. For example, you can configure Network Policy Server (NPS) by using either the NPS MMC snap-in or the netsh commands in the **netsh nps** context.

In addition, there are netsh commands for network technologies, such as for IPv6, network bridge, and Remote Procedure Call (RPC), that are not available in Windows Server as an MMC snap-in.

IMPORTANT

It is recommended that you use Windows PowerShell to manage networking technologies in [Windows Server 2016](#) and [Windows 10](#) rather than Network Shell. Network Shell is included for compatibility with your scripts, however, and its use is supported.

Network Shell (Netsh) Technical Reference

The Netsh Technical Reference provides a comprehensive netsh command reference, including syntax, parameters, and examples for netsh commands. You can use the Netsh Technical Reference to build scripts and batch files by using netsh commands for local or remote management of network technologies on computers running Windows Server 2016 and Windows 10.

Content availability

The Network Shell Technical Reference is available for download in Windows Help (*.chm) format from TechNet Gallery: [Netsh Technical Reference](#)

Netsh Command Syntax, Contexts, and Formatting

9/1/2018 • 7 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to enter netsh contexts and subcontexts, understand netsh syntax and command formatting, and how to run netsh commands on local and remote computers.

Netsh is a command-line scripting utility that allows you to display or modify the network configuration of a computer that is currently running. Netsh commands can be run by typing commands at the netsh prompt and they can be used in batch files or scripts. Remote computers and the local computer can be configured by using netsh commands.

Netsh also provides a scripting feature that allows you to run a group of commands in batch mode against a specified computer. With netsh, you can save a configuration script in a text file for archival purposes or to help you configure other computers.

Netsh contexts

Netsh interacts with other operating system components by using dynamic-link library (DLL) files.

Each netsh helper DLL provides an extensive set of features called a *context*, which is a group of commands specific to a networking server role or feature. These contexts extend the functionality of netsh by providing configuration and monitoring support for one or more services, utilities, or protocols. For example, Dhcpcmon.dll provides netsh with the context and set of commands necessary to configure and manage DHCP servers.

Obtain a list of contexts

You can obtain a list of netsh contexts by opening either command prompt or Windows PowerShell on a computer running Windows Server 2016 or Windows 10. Type the command **netsh** and press ENTER. Type **/?**, and then press ENTER.

Following is example output for these commands on a computer running Windows Server 2016 Datacenter.

```
PS C:\Windows\system32> netsh
netsh>?

The following commands are available:

Commands in this context:
..           - Goes up one context level.
?            - Displays a list of commands.
abort       - Discards changes made while in offline mode.
add         - Adds a configuration entry to a list of entries.
advfirewall - Changes to the `netsh advfirewall` context.
alias       - Adds an alias.
branchcache - Changes to the `netsh branchcache` context.
bridge      - Changes to the `netsh bridge` context.
bye         - Exits the program.
commit      - Commits changes made while in offline mode.
delete      - Deletes a configuration entry from a list of entries.
dhcpclient   - Changes to the `netsh dhcpclient` context.
dnsclient    - Changes to the `netsh dnsclient` context.
dump        - Displays a configuration script.
exec        - Runs a script file.
exit        - Exits the program.
firewall    - Changes to the `netsh firewall` context.
help        - Displays a list of commands.
http         - Changes to the `netsh http` context.
interface   - Changes to the `netsh interface` context.
ipsec       - Changes to the `netsh ipsec` context.
ipsecdosprotection - Changes to the `netsh ipsecdosprotection` context.
lan          - Changes to the `netsh lan` context.
namespace   - Changes to the `netsh namespace` context.
netio        - Changes to the `netsh netio` context.
offline     - Sets the current mode to offline.
online      - Sets the current mode to online.
popd        - Pops a context from the stack.
pushd       - Pushes current context on stack.
quit        - Exits the program.
ras          - Changes to the `netsh ras` context.
rpc          - Changes to the `netsh rpc` context.
set         - Updates configuration settings.
show        - Displays information.
trace       - Changes to the `netsh trace` context.
unalias    - Deletes an alias.
wfp          - Changes to the `netsh wfp` context.
winhttp     - Changes to the `netsh winhttp` context.
winsock     - Changes to the `netsh winsock` context.
```

The following sub-contexts are available:

```
advfirewall branchcache bridge dhcpclient dnsclient firewall http interface ipsec ipsecdosprotection lan
namespace netio ras rpc trace wfp winhttp winsock
```

To view help for a command, type the command, followed by a space, and then type **?**.

Subcontexts

Netsh contexts can contain both commands and additional contexts, called *subcontexts*. For example, within the Routing context, you can change to the IP and IPv6 subcontexts.

To display a list of commands and subcontexts that you can use within a context, at the netsh prompt, type the context name, and then type either **/?** or **help**. For example, to display a list of subcontexts and commands that you can use in the Routing context, at the netsh prompt (that is, **netsh>**), type one of the following:

routing /?

routing help

To perform tasks in another context without changing from your current context, type the context path of the command you want to use at the netsh prompt. For example, to add an interface named "Local Area Connection" in the IGMP context without first changing to the IGMP context, at the netsh prompt, type:

```
routing ip igmp add interface "Local Area Connection" startupqueryinterval=21
```

Running netsh commands

To run a netsh command, you must start netsh from the command prompt by typing **netsh** and then pressing ENTER. Next, you can change to the context that contains the command you want to use. The contexts that are available depend on the networking components that you have installed. For example, if you type **dhcp** at the netsh prompt and press ENTER, netsh changes to the DHCP server context. If you do not have DHCP installed, however, the following message appears:

The following command was not found: dhcp.

Formatting Legend

You can use the following formatting legend to interpret and use correct netsh command syntax when you run the command at the netsh prompt or in a batch file or script.

- Text in *Italic* is information that you must supply while you type the command. For example, if a command has a parameter named *-UserName*, you must type the actual user name.
- Text in **Bold** is information that you must type exactly as shown while you type the command.
- Text followed by an ellipsis (...) is a parameter that can be repeated several times in a command line.
- Text that is between brackets [] is an optional item.
- Text that is between braces { } with choices separated by a pipe provides a set of choices from which you must select only one, such as `{enable|disable}`.
- Text that is formatted with the Courier font is code or program output.

Running Netsh commands from the command prompt or Windows PowerShell

To start Network Shell and enter netsh at the command prompt or in Windows PowerShell, you can use the following command.

netsh

Netsh is a command-line scripting utility that allows you to, either locally or remotely, display or modify the network configuration of a currently running computer. Used without parameters, **netsh** opens the Netsh.exe command prompt (that is, **netsh>**).

Syntax

```
netsh[ -a AliasFile] [ -c Context] [ -r RemoteComputer] [ -u [ DomainName\] UserName] [ -p Password | *]  
[{NetshCommand | -f ScriptFile}]
```

Parameters

`-a`

Optional. Specifies that you are returned to the **netsh** prompt after running *AliasFile*.

`AliasFile`

Optional. Specifies the name of the text file that contains one or more **netsh** commands.

`-c`

Optional. Specifies that netsh enters the specified **netsh** context.

`Context`

Optional. Specifies the **netsh** context that you want to enter.

`-r`

Optional. Specifies that you want the command to run on a remote computer.

IMPORTANT

When you use some netsh commands remotely on another computer with the **netsh -r** parameter, the Remote Registry service must be running on the remote computer. If it is not running, Windows displays a "Network Path Not Found" error message.

`RemoteComputer`

Optional. Specifies the remote computer that you want to configure.

`-u`

Optional. Specifies that you want to run the netsh command under a user account.

`DomainName \\`

Optional. Specifies the domain where the user account is located. The default is the local domain if *DomainName* is not specified.

`UserName`

Optional. Specifies the user account name.

`-p`

Optional. Specifies that you want to provide a password for the user account.

`Password`

Optional. Specifies the password for the user account that you specified with **-u UserName**.

`NetshCommand`

Optional. Specifies the **netsh** command that you want to run.

`-f`

Optional. Exits **netsh** after running the script that you designate with *ScriptFile*.

`ScriptFile`

Optional. Specifies the script that you want to run.

`/?`

Optional. Displays help at the netsh prompt.

NOTE

If you specify `-r` followed by another command, **netsh** runs the command on the remote computer and then returns to the Cmd.exe command prompt. If you specify `-r` without another command, **netsh** opens in remote mode. The process is similar to using **set machine** at the Netsh command prompt. When you use `-r`, you set the target computer for the current instance of **netsh** only. After you exit and reenter **netsh**, the target computer is reset as the local computer. You can run **netsh** commands on a remote computer by specifying a computer name stored in WINS, a UNC name, an Internet name to be resolved by the DNS server, or an IP address.

Typing parameter string values for netsh commands

Throughout the Netsh command reference there are commands that contain parameters for which a string value is required.

In the case where a string value contains spaces between characters, such as string values that consist of more than one word, it is required that you enclose the string value in quotation marks. For example, for a parameter named **interface** with a string value of **Wireless Network Connection**, use quotation marks around the string value:

```
interface="Wireless Network Connection"
```

Network Shell (Netsh) Example Batch File

3/23/2018 • 2 minutes to read • [Edit Online](#)

Applies To: Windows Server 2016

You can use this topic to learn how to create a batch file that performs multiple tasks by using Netsh in Windows Server 2016. In this example batch file, the **netsh wins** context is used.

Example Batch File Overview

You can use Netsh commands for Windows Internet Name Service (WINS) in batch files and other scripts to automate tasks. The following batch file example demonstrates how to use Netsh commands for WINS to perform a variety of related tasks.

In this example batch file, WINS-A is a WINS server with the IP address 192.168.125.30 and WINS-B is a WINS server with the IP address 192.168.0.189.

The example batch file accomplishes the following tasks.

- Adds a dynamic name record with IP address 192.168.0.205, MY_RECORD [04h], to WINS-A
- Sets WINS-B as a push/pull replication partner of WINS-A
- Connects to WINS-B, and then sets WINS-A as a push/pull replication partner of WINS-B
- Initiates a push replication from WINS-A to WINS-B
- Connects to WINS-B to verify that the new record, MY_RECORD, was replicated successfully

Netsh example batch file

In the following example batch file, lines that contain comments are preceded by "rem," for remark. Netsh ignores comments.

```
rem: Begin example batch file.

rem two WINS servers:

rem (WINS-A) 192.168.125.30

rem (WINS-B) 192.168.0.189

rem 1. Connect to (WINS-A), and add the dynamic name MY\_RECORD \[04h\] to the (WINS-A) database.

netsh wins server 192.168.125.30 add name Name=MY\_RECORD EndChar=04 IP={192.168.0.205}

rem 2. Connect to (WINS-A), and set (WINS-B) as a push/pull replication partner of (WINS-A).

netsh wins server 192.168.125.30 add partner Server=192.168.0.189 Type=2

rem 3. Connect to (WINS-B), and set (WINS-A) as a push/pull replication partner of (WINS-B).

netsh wins server 192.168.0.189 add partner Server=192.168.125.30 Type=2

rem 4. Connect back to (WINS-A), and initiate a push replication to (WINS-B).

netsh wins server 192.168.125.30 init push Server=192.168.0.189 PropReq=0

rem 5. Connect to (WINS-B), and check that the record MY\_RECORD \[04h\] was replicated successfully.

netsh wins server 192.168.0.189 show name Name=MY\_RECORD EndChar=04

rem 6. End example batch file.
```

Netsh WINS commands used in the example batch file

The following section lists the **netsh wins** commands that are used in this example procedure.

- **server**. Shifts the current WINS command-line context to the server specified by either its name or IP address.
- **add name**. Registers a name on the WINS server.
- **add partner**. Adds a replication partner on the WINS server.
- **init push**. Initiates and sends a push trigger to a WINS server.
- **show name**. Displays detailed information for a particular record in the WINS server database.

For more information, see [Network Shell \(Netsh\)](#).

Netsh http commands

9/21/2018 • 7 minutes to read • [Edit Online](#)

Use **netsh http** to query and configure HTTP.sys settings and parameters.

TIP

If you are using Windows PowerShell on a computer running Windows Server 2016 or Windows 10, type **netsh** and press Enter. At the netsh prompt, type **http** and press Enter to get the netsh http prompt.

```
netsh http>
```

The available netsh http commands are:

- [add iplisten](#)
- [add sslcert](#)
- [add timeout](#)
- [add urlacl](#)
- [delete cache](#)
- [delete iplisten](#)
- [delete sslcert](#)
- [delete timeout](#)
- [delete urlacl](#)
- [flush logbuffer](#)
- [show cachestate](#)
- [show iplisten](#)
- [show servicestate](#)
- [show sslcert](#)
- [show timeout](#)
- [show urlacl](#)

add iplisten

Adds a new IP address to the IP listen list, excluding the port number.

Syntax

```
add iplisten [ ipaddress= ] IPAddress
```

Parameters

Parameter	Description	Required
ipaddress	The IPv4 or IPv6 address to be added to the IP listen list. The IP listen list is used to scope the list of addresses to which the HTTP service binds. "0.0.0.0" means any IPv4 address and "::" means any IPv6 address.	Required

Examples

Following are four examples of the **add iplistens** command.

- add iplistens ipaddress=fe80::1
- add iplistens ipaddress=1.1.1.1
- add iplistens ipaddress=0.0.0.0
- add iplistens ipaddress=::

add sslcert

Adds a new SSL server certificate binding and corresponding client certificate policies for an IP address and port.

Syntax

```
add sslcert [ ipport= ] IPAddress:port [ certhash= ] CertHash [ appid= ] GUID [ [ certstorename= ] CertStoreName [ verifyclientcertrevocation= ] enable | disable [ verifyrevocationwithcachedclientcertonly= ] enable | disable [ usagecheck= ] enable | disable [ revocationfreshnesstime= ] U-Int [ urlretrievaltimeout= ] U-Int [ sslctlidentifier= ] SSLCTIdentifier [ sslctlstorename= ] SLCTStoreName [ dsmapperusage= ] enable | disable [ clientcertnegotiation= ] enable | disable ] ]
```

Parameters

ipport	Specifies the IP address and port for the binding. A colon character (:) is used as a delimiter between the IP address and the port number.	Required
certhash	Specifies the SHA hash of the certificate. This hash is 20 bytes long and is specified as a hexadecimal string.	Required
appid	Specifies the GUID to identify the owning application.	Required
certstorename	Specifies the store name for the certificate. Defaults to MY. Certificate must be stored in the local machine context.	Optional
verifyclientcertrevocation	Specifies the Turns on/off verification of revocation of client certificates.	Optional
verifyrevocationwithcachedclientcertonly	Specifies whether the usage of only cached client certificate for revocation checking is enabled or disabled.	Optional
usagecheck	Specifies whether the usage check is enabled or disabled. Default is enabled.	Optional
revocationfreshnesstime	Specifies the time interval, in seconds, to check for an updated certificate revocation list (CRL). If this value is zero, then the new CRL is updated only if the previous one expires.	Optional

urlretrievaltimeout	Specifies the timeout interval (in milliseconds) after the attempt to retrieve the certificate revocation list for the remote URL.	Optional
sslctlidentifier	Specifies the list of the certificate issuers that can be trusted. This list can be a subset of the certificate issuers that are trusted by the computer.	Optional
sslctlstorename	Specifies the certificate store name under LOCAL_MACHINE where SslCtlIdentifier is stored.	Optional
dsmapperusage	Specifies whether DS mappers is enabled or disabled. Default is disabled.	Optional
clientcertnegotiation	Specifies whether the negotiation of certificate is enabled or disabled. Default is disabled.	Optional

Examples

Following is an example of the **add sslicert** command.

```
add sslicert ipport=1.1.1.1:443 certhash=0102030405060708090A0B0C0D0E0F1011121314 appid={00112233-4455-6677-8899- AABC C DDEEFF}
```

add timeout

Adds a global timeout to the service.

Syntax

```
add timeout [ timeouttype= ] IdleConnectionTimeout | HeaderWaitTimeout [ value= ] U-Short
```

Parameters

timeouttype	Type of timeout for setting.
value	Value of the timeout (in seconds). If the value is in hexadecimal notation, then add the prefix 0x.

Examples

Following are two examples of the **add timeout** command.

- add timeout timeouttype=idleconnectiontimeout value=120
- add timeout timeouttype=headerwaittimeout value=0x40

add urlacl

Adds a Uniform Resource Locator (URL) reservation entry. This command reserves the URL for non-administrator users and accounts. The DACL can be specified by using an NT account name with the listen and delegate parameters or by using an SDDL string.

Syntax

```
add urlacl [ url= ] URL [ [user=] User [ [ listen= ] yes | no [ delegate= ] yes | no ] | [ sddl= ] SDDL ]
```

Parameters

url	Specifies the fully qualified Uniform Resource Locator (URL).	Required
user	Specifies the user or user-group name	Required
listen	Specifies one of the following values: yes: Allow the user to register URLs. This is the default value. no: Deny the user from registering URLs.	Optional
delegate	Specifies one of the following values: yes: Allow the user to delegate URLs no: Deny the user from delegating URLs. This is the default value.	Optional
sddl	Specifies an SDDL string that describes the DACL.	Optional

Examples

Following are four examples of the **add urlacl** command.

- add urlacl url=http://+:80/MyUri user=DOMAIN\user
- add urlacl url=http://www.contoso.com:80/MyUri user=DOMAIN\user listen=yes
- add urlacl url=http://www.contoso.com:80/MyUri user=DOMAIN\user delegate=no
- add urlacl url=http://+:80/MyUri sddl=...

delete cache

Deletes all the entries, or a specified entry, from the HTTP service kernel URI cache.

Syntax

```
delete cache [ [ url= ] URL [ [recursive= ] yes | no ]
```

Parameters

url	Specifies the fully qualified Uniform Resource Locator (URL) that you want to delete.	Optional
------------	---	----------

recursive	Specifies whether all entries under the url cache get removed. yes : remove all entries no : do not remove all entries	Optional
------------------	--	----------

Examples

Following are two examples of the **delete cache** command.

- delete cache url=<http://www.contoso.com:80/myresource/> recursive=yes
- delete cache

delete iplisten

Deletes an IP address from the IP listen list. The IP listen list is used to scope the list of addresses to which the HTTP service binds.

Syntax

```
delete iplisten [ ipaddress= ] IPAddress
```

Parameters

ipaddress	The IPv4 or IPv6 address to be deleted from the IP listen list. The IP listen list is used to scope the list of addresses to which the HTTP service binds. "0.0.0.0" means any IPv4 address and "::" means any IPv6 address. This does not include the port number.	Required
------------------	---	----------

Examples

Following are four examples of the **delete iplisten** command.

- delete iplisten ipaddress=fe80::1
- delete iplisten ipaddress=1.1.1.1
- delete iplisten ipaddress=0.0.0.0
- delete iplisten ipaddress=::

delete sslcert

Deletes SSL server certificate bindings and corresponding client certificate policies for an IP address and port.

Syntax

```
delete sslcert [ ipport= ] IPAddress:port
```

Parameters

ipport	Specifies the IPv4 or IPv6 address and port for which the SSL certificate bindings get deleted. A colon character (:) is used as a delimiter between the IP address and the port number.	Required
---------------	--	----------

Examples

Following are three examples of the **delete sslcert** command.

- delete sslcert ipport=1.1.1.1:443
- delete sslcert ipport=0.0.0.0:443
- delete sslcert ipport=[::]:443

delete timeout

Deletes a global timeout and makes the service revert to default values.

Syntax

```
delete timeout [ timeouttype= ] idleconnectiontimeout | headerwaittimeout
```

Parameters

timeouttype	Specifies the type of timeout setting.	Required
--------------------	--	----------

Examples

Following are two examples of the **delete timeout** command.

- delete timeout timeouttype=idleconnectiontimeout
- delete timeout timeouttype=headerwaittimeout

delete urlacl

Deletes URL reservations.

Syntax

```
delete urlacl [ url= ] URL
```

Parameters

url	Specifies the fully qualified Uniform Resource Locator (URL) that you want to delete.	Required
------------	---	----------

Examples

Following are two examples of the **delete urlacl** command.

- delete urlacl url=http://+:80/MyUri
- delete urlacl url=<http://www.contoso.com:80/MyUri>

flush logbuffer

Flushes the internal buffers for the logfiles.

Syntax

```
flush logbuffer
```

show cachestate

Lists cached URI resources and their associated properties. This command lists all resources and their associated properties that are cached in HTTP response cache or displays a single resource and its associated properties.

Syntax

```
show cachestate [ [url= ] URL]
```

Parameters

Parameter	Description	Optional
url	Specifies the fully qualified URL that you want to display. If not specified, display all URLs. The URL could also be a prefix to registered URLs.	Optional

Examples

Following are two examples of the **show cachestate** command:

- show cachestate url=<http://www.contoso.com:80/myresource>
- show cachestate

show iplisten

Displays all IP addresses in the IP listen list. The IP listen list is used to scope the list of addresses to which the HTTP service binds. "0.0.0.0" means any IPv4 address and ":" means any IPv6 address.

Syntax

```
show iplisten
```

show servicestate

Displays a snapshot of the HTTP service.

Syntax

```
show servicestate [ [ view= ] session | requestq ] [ [ verbose= ] yes | no ]
```

Parameters

View	Specifies whether to view a snapshot of the HTTP service state based on the server session or on the request queues.	Optional
Verbose	Specifies whether to display verbose information that also shows property information.	Optional

Examples

Following are two examples of the **show servicestate** command.

- show servicestate view="session"
- show servicestate view="requestq"

show sslcert

Displays Secure Sockets Layer (SSL) server certificate bindings and corresponding client certificate policies for an IP address and port.

Syntax

```
show sslcert [ ipport= ] IPAddress:port
```

Parameters

ipport	Specifies the IPv4 or IPv6 address and port for which the SSL certificate bindings display. A colon character (:) is used as a delimiter between the IP address and the port number. If you do not specify ipport, all bindings are displayed.	Required

Examples

Following are five examples of the **show sslcert** command.

- show sslcert ipport=[fe80::1]:443
- show sslcert ipport=1.1.1.1:443
- show sslcert ipport=0.0.0.0:443
- show sslcert ipport=[::]:443
- show sslcert

show timeout

Displays, in seconds, the timeout values of the HTTP service.

Syntax

```
show timeout
```

show urlacl

Displays discretionary access control lists (DACLs) for the specified reserved URL or all reserved URLs.

Syntax

```
show urlacl [ [url= ] URL]
```

Parameters

Parameter	Description	Optional
url	Specifies the fully qualified URL that you want to display. If not specified, display all URLs.	Optional

Examples

Following are three examples of the **show urlacl** command.

- show urlacl url=http://+:80/MyUri
- show urlacl url=<http://www.contoso.com:80/MyUri>
- show urlacl

Netsh interface portproxy commands

9/21/2018 • 9 minutes to read • [Edit Online](#)

Use the **netsh interface portproxy** commands to act as proxies between IPv4 and IPv6 networks and applications. You can use these commands to establish proxy service in the following ways:

- IPv4-configured computer and application messages sent to other IPv4-configured computers and applications.
- IPv4-configured computer and application messages sent to IPv6-configured computers and applications.
- IPv6-configured computer and application messages sent to IPv4-configured computers and applications.
- IPv6-configured computer and application messages sent to other IPv6-configured computers and applications.

When writing batch files or scripts using these commands, each command must start with **netsh interface portproxy**. For example, when using the **delete v4tov6** command to specify that the portproxy server deletes an IPv4 port and address from the list of IPv4 addresses for which the server listens, the batch file or script must use the following syntax:

```
netsh interface portproxy delete v4tov6listenport= {Integer | ServiceName} [[listenaddress=] {IPv4Address|HostName}] [[protocol=]tcp]
```

The available netsh interface portproxy commands are:

- [add v4tov4](#)
- [add v4tov6](#)
- [add v6tov4](#)
- [add v6tov6](#)
- [delete v4tov4](#)
- [delete v4tov6](#)
- [delete v6tov6](#)
- [reset](#)
- [set v4tov4](#)
- [set v4tov6](#)
- [setv6tov4](#)
- [set v6tov6](#)
- [show all](#)
- [show v4tov4](#)
- [show v4tov6](#)
- [show v6tov4](#)

- [show v6tov6](#)

add v4tov4

The portproxy server listens for messages sent to a specific port and IPv4 address and maps a port and IPv4 address to send the messages received after establishing a separate TCP connection.

Syntax

```
add v4tov4 listenport= {Integer | ServiceName} [[connectaddress=] {IPv4Address | HostName}] [[connectport=]
{Integer | ServiceName}] [[listenaddress=] {IPv4Address | HostName}] [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv4 port, by port number or service name, on which to listen.
connectaddress	Specifies the IPv4 address to which to connect. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
connectport	Specifies the IPv4 port, by port number or service name, to which to connect. If connectport is not specified, the default is the value of listenport on the local computer.
listenaddress	Specifies the IPv4 address for which to listen. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
protocol	Specifies the protocol to use.

add v4tov6

The portproxy server listens for messages sent to a specific port and IPv4 address, and maps a port and IPv6 address to send the messages received after establishing a separate TCP connection.

Syntax

```
add v4tov6 listenport= {Integer | ServiceName} [[connectaddress=] {IPv6Address | HostName}] [[connectport=]
{Integer | ServiceName}] [[listenaddress=] {IPv4Address | HostName}] [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv4 port, by port number or service name, on which to listen.
connectaddress	Specifies the IPv6 address to which to connect. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.

connectport	Specifies the IPv6 port, by port number or service name, to which to connect. If connectport is not specified, the default is the value of listenport on the local computer.
listenaddress	Specifies the IPv4 address on which to listen. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
protocol	Specifies the protocol to use.

add v6tov4

The portproxy server listens for messages sent to a specific port and IPv6 address, and maps a port and IPv4 address to which to send the messages received after establishing a separate TCP connection.

Syntax

```
add v6tov4 listenport= {Integer | ServiceName} [[connectaddress=] {IPv4Address | HostName} [[connectport=] {Integer | ServiceName}] [[listenaddress=] {IPv6Address | HostName} [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv6 port, by port number or service name, on which to listen.
connectaddress	Specifies the IPv4 address to which to connect. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
connectport	Specifies the IPv4 port, by port number or service name, to which to connect. If connectport is not specified, the default is the value of listenport on the local computer.
listenaddress	Specifies the IPv6 address on which to listen. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
protocol	Specifies the protocol to use.

add v6tov6

The portproxy server listens for messages sent to a specific port and IPv6 address, and maps a port and IPv6 address to which to send the messages received after establishing a separate TCP connection.

Syntax

```
add v6tov6 listenport= {Integer | ServiceName} [[connectaddress=] {IPv6Address | HostName} [[connectport=] {Integer | ServiceName}] [[listenaddress=] {IPv6Address | HostName} [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv6 port, by port number or service name, on which to listen.
connectaddress	Specifies the IPv6 address to which to connect. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
connectport	Specifies the IPv6 port, by port number or service name, to which to connect. If connectport is not specified, the default is the value of listenport on the local computer.
listenaddress	Specifies the IPv6 address on which to listen. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
protocol	Specifies the protocol to use.

delete v4tov4

The portproxy server deletes an IPv4 address from the list of IPv4 ports and addresses for which the server listens.

Syntax

```
delete v4tov4 listenport= {Integer | ServiceName} [[listenaddress=] {IPv4Address | HostName} [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv4 port to delete.
listenaddress	Specifies the IPv4 address to delete. If an address is not specified, the default is the local computer.
protocol	Specifies the protocol to use.

delete v4tov6

The portproxy server deletes an IPv4 port and address from the list of IPv4 addresses for which the server listens.

Syntax

```
delete v4tov6 listenport= {Integer | ServiceName} [[listenaddress=] {IPv4Address | HostName} [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv4 port to delete.
listenaddress	Specifies the IPv4 address to delete. If an address is not specified, the default is the local computer.

protocol	Specifies the protocol to use.
-----------------	--------------------------------

delete v6tov4

The portproxy server deletes an IPv6 port and address from the list of IPv6 addresses for which the server listens.

Syntax

```
delete v6tov4 listenport= {Integer | ServiceName} [[listenaddress=] {IPv6Address | HostName} [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv6 port to delete.
listenaddress	Specifies the IPv6 address to delete. If an address is not specified, the default is the local computer.
protocol	Specifies the protocol to use.

delete v6tov6

The portproxy server deletes an IPv6 address from the list of IPv6 addresses for which the server listens.

Syntax

```
delete v6tov6 listenport= {Integer | ServiceName} [[listenaddress=] {IPv6Address | HostName} [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv6 port to delete.
listenaddress	Specifies the IPv6 address to delete. If an address is not specified, the default is the local computer.
protocol	Specifies the protocol to use.

reset

Resets the IPv6 configuration state.

Syntax

```
reset
```

set v4tov4

Modifies the parameter values of an existing entry on the portproxy server created with the **add v4tov4** command, or adds a new entry to the list that maps port/address pairs.

Syntax

```
set v4tov4 listenport= {Integer | ServiceName} [[connectaddress=] {IPv4Address | HostName} [[connectport=]
{Integer | ServiceName}] [[listenaddress=] {IPv4Address | HostName} [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv4 port, by port number or service name, on which to listen.
connectaddress	Specifies the IPv4 address to which to connect. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
connectport	Specifies the IPv4 port, by port number or service name, to which to connect. If connectport is not specified, the default is the value of listenport on the local computer.
listenaddress	Specifies the IPv4 address for which to listen. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
protocol	Specifies the protocol to use.

set v4tov6

Modifies the parameter values of an existing entry on the portproxy server created with the **add v4tov6** command, or adds a new entry to the list that maps port/address pairs.

Syntax

```
set v4tov6 listenport= {Integer | ServiceName} [[connectaddress=] {IPv6Address | HostName} [[connectport=]
{Integer | ServiceName}] [[listenaddress=] {IPv4Address | HostName} [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv4 port, by port number or service name, on which to listen.
connectaddress	Specifies the IPv6 address to which to connect. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
connectport	Specifies the IPv6 port, by port number or service name, to which to connect. If connectport is not specified, the default is the value of listenport on the local computer.
listenaddress	Specifies the IPv4 address on which to listen. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.

protocol	Specifies the protocol to use.
-----------------	--------------------------------

set v6tov4

Modifies the parameter values of an existing entry on the portproxy server created with the **add v6tov4** command, or adds a new entry to the list that maps port/address pairs.

Syntax

```
set v6tov4 listenport= {Integer | ServiceName} [[connectaddress=] {IPv4Address | HostName} [[connectport=]
{Integer | ServiceName}] [[listenaddress=] {IPv6Address | HostName} [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv6 port, by port number or service name, on which to listen.
connectaddress	Specifies the IPv4 address to which to connect. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
connectport	Specifies the IPv4 port, by port number or service name, to which to connect. If connectport is not specified, the default is the value of listenport on the local computer.
listenaddress	Specifies the IPv6 address on which to listen. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
protocol	Specifies the protocol to use.

set v6tov6

Modifies the parameter values of an existing entry on the portproxy server created with the **add v6tov6** command, or adds a new entry to the list that maps port/address pairs.

Syntax

```
set v6tov6 listenport= {Integer | ServiceName} [[connectaddress=] {IPv6Address | HostName} [[connectport=]
{Integer | ServiceName}] [[listenaddress=] {IPv6Address | HostName} [[protocol=]tcp]
```

Parameters

listenport	Specifies the IPv6 port, by port number or service name, on which to listen.
-------------------	--

connectaddress	Specifies the IPv6 address to which to connect. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If an address is not specified, the default is the local computer.
connectport	Specifies the IPv6 port, by port number or service name, to which to connect. If connectport is not specified, the default is the value of listenport on the local computer.
listenaddress	Specifies the IPv6 address on which to listen. Acceptable values are IP address, computer NetBIOS name, or computer DNS name. If you do not specify an address, the default is the local computer.
protocol	Specifies the protocol to use.

show all

Displays all portproxy parameters, including port/address pairs for v4tov4, v4tov6, v6tov4, and v6tov6.

Syntax

```
show all
```

show v4tov4

Displays v4tov4 portproxy parameters.

Syntax

```
show v4tov4
```

show v4tov6

Displays v4tov6 portproxy parameters.

Syntax

```
show v4tov6
```

show v6tov4

Displays v6tov4 portproxy parameters.

Syntax

```
show v6tov4
```

show v6tov6

Displays v6tov6 portproxy parameters.

Syntax

```
show v6tov6
```

Network Subsystem Performance Tuning

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic for an overview of the network subsystem and for links to other topics in this guide.

NOTE

In addition to this topic, the following sections of this guide provide performance tuning recommendations for network devices and the network stack.

- [Choosing a Network Adapter](#)
- [Configure the Order of Network Interfaces](#)
- [Performance Tuning Network Adapters](#)
- [Network-Related Performance Counters](#)
- [Performance Tools for Network Workloads](#)

Performance tuning the network subsystem, particularly for network intensive workloads, can involve each layer of the network architecture, which is also called the network stack. These layers are broadly divided into the following sections.

1. **Network interface.** This is the lowest layer in the network stack, and contains the network driver that communicates directly with the network adapter.
2. **Network Driver Interface Specification (NDIS).** NDIS exposes interfaces for the driver below it and for the layers above it, such as the Protocol Stack.
3. **Protocol Stack.** The protocol stack implements protocols such as TCP/IP and UDP/IP. These layers expose the transport layer interface for layers above them.
4. **System Drivers.** These are typically clients that use a transport data extension (TDX) or Winsock Kernel (WSK) interface to expose interfaces to user-mode applications. The WSK interface was introduced in Windows Server 2008 and Windows® Vista, and it is exposed by AFD.sys. The interface improves performance by eliminating the switching between user mode and kernel mode.
5. **User-Mode Applications.** These are typically Microsoft solutions or custom applications.

The table below provides a vertical illustration of the layers of the network stack, including examples of items that run in each layer.

5	User-Mode Applications	WMS	DNS	IIS
4	System Drivers	AFD.sys	HTTP.sys	
3	Protocol Stack	TCP/IP	UDP/IP	VPN
2	NDIS	Network Driver Interface Specification (NDIS)		
1	Network interface	Network driver		

Choosing a Network Adapter

9/1/2018 • 10 minutes to read • [Edit Online](#)

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

You can use this topic to learn some of the features of network adapters that might affect your purchasing choices.

Network-intensive applications require high-performance network adapters. This section explores some considerations for choosing network adapters, as well as how to configure different network adapter settings to achieve the best network performance.

TIP

You can configure network adapter settings by using Windows PowerShell. For more information, see [Network Adapter Cmdlets in Windows PowerShell](#).

Offload Capabilities

Offloading tasks from the central processing unit (CPU) to the network adapter can reduce CPU usage on the server, which improves the overall system performance.

The network stack in Microsoft products can offload one or more tasks to a network adapter if you select a network adapter that has the appropriate offload capabilities. The following table provides a brief overview of different offload capabilities that are available in Windows Server 2016.

OFFLOAD TYPE	DESCRIPTION
Checksum calculation for TCP	The network stack can offload the calculation and validation of Transmission Control Protocol (TCP) checksums on send and receive code paths. It can also offload the calculation and validation of IPv4 and IPv6 checksums on send and receive code paths.
Checksum calculation for UDP	The network stack can offload the calculation and validation of User Datagram Protocol (UDP) checksums on send and receive code paths.
Checksum calculation for IPv4	The network stack can offload the calculation and validation of IPv4 checksums on send and receive code paths.
Checksum calculation for IPv6	The network stack can offload the calculation and validation of IPv6 checksums on send and receive code paths.
Segmentation of large TCP packets	The TCP/IP transport layer supports Large Send Offload v2 (LSOv2). With LSOv2, the TCP/IP transport layer can offload the segmentation of large TCP packets to the network adapter.

OFFLOAD TYPE	DESCRIPTION
Receive Side Scaling (RSS)	RSS is a network driver technology that enables the efficient distribution of network receive processing across multiple CPUs in multiprocessor systems. More detail about RSS is provided later in this topic.
Receive Segment Coalescing (RSC)	RSC is the ability to group packets together to minimize the header processing that is necessary for the host to perform. A maximum of 64 KB of received payload can be coalesced into a single larger packet for processing. More detail about RSC is provided later in this topic.

Receive Side Scaling

Windows Server 2016, Windows Server 2012, Windows Server 2012 R2, Windows Server 2008 R2, and Windows Server 2008 support Receive Side Scaling (RSS).

Some servers are configured with multiple logical processors that share hardware resources (such as a physical core) and which are treated as Simultaneous Multi-Threading (SMT) peers. Intel Hyper-Threading Technology is an example. RSS directs network processing to up to one logical processor per core. For example, on a server with Intel Hyper-Threading, 4 cores, and 8 logical processors, RSS uses no more than 4 logical processors for network processing.

RSS distributes incoming network I/O packets among logical processors so that packets which belong to the same TCP connection are processed on the same logical processor, which preserves ordering.

RSS also load balances UDP unicast and multicast traffic, and it routes related flows (which are determined by hashing the source and destination addresses) to the same logical processor, preserving the order of related arrivals. This helps improve scalability and performance for receive-intensive scenarios for servers that have fewer network adapters than they do eligible logical processors.

Configuring RSS

In Windows Server 2016, you can configure RSS by using Windows PowerShell cmdlets and RSS profiles.

You can define RSS profiles by using the **-Profile** parameter of the **Set-NetAdapterRss** Windows PowerShell cmdlet.

Windows PowerShell commands for RSS configuration

The following cmdlets allow you to see and modify RSS parameters per network adapter.

NOTE

For a detailed command reference for each cmdlet, including syntax and parameters, you can click the following links. In addition, you can pass the cmdlet name to **Get-Help** at the Windows PowerShell prompt for details on each command.

- [Disable-NetAdapterRss](#). This command disables RSS on the network adapter that you specify.
- [Enable-NetAdapterRss](#). This command enables RSS on the network adapter that you specify.
- [Get-NetAdapterRss](#). This command retrieves RSS properties of the network adapter that you specify.
- [Set-NetAdapterRss](#). This command sets the RSS properties on the network adapter that you specify.

RSS profiles

You can use the **-Profile** parameter of the **Set-NetAdapterRss** cmdlet to specify which logical processors are assigned to which network adapter. Available values for this parameter are:

- **Closest**. Logical processor numbers that are near the network adapter's base RSS processor are preferred. With this profile, the operating system might rebalance logical processors dynamically based on load.
- **ClosestStatic**. Logical processor numbers near the network adapter's base RSS processor are preferred. With this profile, the operating system does not rebalance logical processors dynamically based on load.
- **NUMA**. Logical processor numbers are generally selected on different NUMA nodes to distribute the load. With this profile, the operating system might rebalance logical processors dynamically based on load.
- **NUMAStatic**. This is the **default profile**. Logical processor numbers are generally selected on different NUMA nodes to distribute the load. With this profile, the operating system will not rebalance logical processors dynamically based on load.
- **Conservative**. RSS uses as few processors as possible to sustain the load. This option helps reduce the number of interrupts.

Depending on the scenario and the workload characteristics, you can also use other parameters of the **Set-NetAdapterRss** Windows PowerShell cmdlet to specify the following:

- On a per-network adapter basis, how many logical processors can be used for RSS.
- The starting offset for the range of logical processors.
- The node from which the network adapter allocates memory.

Following are the additional **Set-NetAdapterRss** parameters that you can use to configure RSS:

NOTE

In the example syntax for each parameter below, the network adapter name **Ethernet** is used as an example value for the **-Name** parameter of the **Set-NetAdapterRss** command. When you run the cmdlet, ensure that the network adapter name that you use is appropriate for your environment.

- * **MaxProcessors**: Sets the maximum number of RSS processors to be used. This ensures that application traffic is bound to a maximum number of processors on a given interface. Example syntax:

```
Set-NetAdapterRss -Name "Ethernet" -MaxProcessors <value>
```

- * **BaseProcessorGroup**: Sets the base processor group of a NUMA node. This impacts the processor array that is used by RSS. Example syntax:

```
Set-NetAdapterRss -Name "Ethernet" -BaseProcessorGroup <value>
```

- * **MaxProcessorGroup**: Sets the Max processor group of a NUMA node. This impacts the processor array that is used by RSS. Setting this would restrict a maximum processor group so that load balancing is aligned within a k-group. Example syntax:

```
Set-NetAdapterRss -Name "Ethernet" -MaxProcessorGroup <value>
```

- * **BaseProcessorNumber**: Sets the base processor number of a NUMA node. This impacts the processor array that is used by RSS. This allows partitioning processors across network adapters. This is the first logical processor in the range of RSS processors that is assigned to each adapter. Example syntax:

```
Set-NetAdapterRss -Name "Ethernet" -BaseProcessorNumber <Byte Value>
```

- * **NumaNode**: The NUMA node that each network adapter can allocate memory from. This can be within a k-group or from different k-groups. Example syntax:

```
Set-NetAdapterRss -Name "Ethernet" -NumaNodeID <value>
```

- * **NumberofReceiveQueues**: If your logical processors seem to be underutilized for receive traffic (for

example, as viewed in Task Manager), you can try increasing the number of RSS queues from the default of 2 to the maximum that is supported by your network adapter. Your network adapter may have options to change the number of RSS queues as part of the driver. Example syntax:

```
Set-NetAdapterRss -Name "Ethernet" -NumberOfReceiveQueues <value>
```

For more information, click the following link to download [Scalable Networking: Eliminating the Receive Processing Bottleneck—Introducing RSS](#) in Word format.

Understanding RSS Performance

Tuning RSS requires understanding the configuration and the load-balancing logic. To verify that the RSS settings have taken effect, you can review the output when you run the **Get-NetAdapterRss** Windows PowerShell cmdlet. Following is example output of this cmdlet.

```
PS C:\Users\Administrator> get-netadapterrss
Name : testnic 2
InterfaceDescription : Broadcom BCM5708C NetXtreme II GigE (NDIS VBD Client) #66
Enabled : True
NumberOfReceiveQueues : 2
Profile : NUMAStatic
BaseProcessor: [Group:Number] : 0:0
MaxProcessor: [Group:Number] : 0:15
MaxProcessors : 8

IndirectionTable: [Group:Number]:
 0:0  0:4  0:0  0:4  0:0  0:4  0:0  0:4
...
(# indirection table entries are a power of 2 and based on # of processors)
...
 0:0  0:4  0:0  0:4  0:0  0:4  0:0  0:4
```

In addition to echoing parameters that were set, the key aspect of the output is the indirection table output. The indirection table displays the hash table buckets that are used to distribute incoming traffic. In this example, the n:c notation designates the Numa K-Group:CPU index pair that is used to direct incoming traffic. We see exactly 2 unique entries (0:0 and 0:4), which represent k-group 0/cpu0 and k-group 0/cpu 4, respectively.

There is only one k-group for this system (k-group 0) and a n (where n <= 128) indirection table entry. Because the number of receive queues is set to 2, only 2 processors (0:0, 0:4) are chosen - even though maximum processors is set to 8. In effect, the indirection table is hashing incoming traffic to only use 2 CPUs out of the 8 that are available.

To fully utilize the CPUs, the number of RSS Receive Queues must be equal to or greater than Max Processors. In the previous example, the Receive Queue should be set to 8 or greater.

NIC Teaming and RSS

RSS can be enabled on a network adapter that is teamed with another network interface card using NIC Teaming. In this scenario, only the underlying physical network adapter can be configured to use RSS. A user cannot set RSS cmdlets on the teamed network adapter.

Receive Segment Coalescing (RSC)

Receive Segment Coalescing (RSC) helps performance by reducing the number of IP headers that are processed for a given amount of received data. It should be used to help scale the performance of received data by grouping (or coalescing) the smaller packets into larger units.

This approach can affect latency with benefits mostly seen in throughput gains. RSC is recommended to increase throughput for received heavy workloads. Consider deploying network adapters that support RSC.

On these network adapters, ensure that RSC is on (this is the default setting), unless you have specific workloads

(for example, low latency, low throughput networking) that show benefit from RSC being off.

Understanding RSC Diagnostics

You can diagnose RSC by using the Windows PowerShell cmdlets **Get-NetAdapterRsc** and **Get-NetAdapterStatistics**.

Following is example output when you run the Get-NetAdapterRsc cmdlet.

```
PS C:\Users\Administrator> Get-NetAdapterRsc

Name          IPv4Enabled  IPv6Enabled  IPv4Operational  IPv6Operational
IPv4FailureReason      IPv6Failure
                           Reason
-----
-----
Ethernet        True       False       True           False
NicProperties

NoFailure
```

The **Get** cmdlet shows whether RSC is enabled in the interface and whether TCP enables RSC to be in an operational state. The failure reason provides details about the failure to enable RSC on that interface.

In the previous scenario, IPv4 RSC is supported and operational in the interface. To understand diagnostic failures, one can see the coalesced bytes or exceptions caused. This provides an indication of the coalescing issues.

Following is example output when you run the Get-NetAdapterStatistics cmdlet.

```
PS C:\Users\Administrator> $x = Get-NetAdapterStatistics "myAdapter"
PS C:\Users\Administrator> $x.rscstatistics

CoalescedBytes      : 0
CoalescedPackets    : 0
CoalescingEvents    : 0
CoalescingExceptions : 0
```

RSC and Virtualization

RSC is only supported in the physical host when the host network adapter is not bound to the Hyper-V Virtual Switch. RSC is disabled by the operating system when the host is bound to the Hyper-V Virtual Switch. In addition, virtual machines do not get the benefit of RSC because virtual network adapters do not support RSC.

RSC can be enabled for a virtual machine when Single Root Input/Output Virtualization (SR-IOV) is enabled. In this case, virtual functions support RSC capability; hence, virtual machines also receive the benefit of RSC.

Network Adapter Resources

A few network adapters actively manage their resources to achieve optimum performance. Several network adapters allow you to manually configure resources by using the **Advanced Networking** tab for the adapter. For such adapters, you can set the values of a number of parameters, including the number of receive buffers and send buffers.

Configuring network adapter resources is simplified by the use of the following Windows PowerShell cmdlets.

- [Get-NetAdapterAdvancedProperty](#)
- [Set-NetAdapterAdvancedProperty](#)
- [Enable-NetAdapter](#)
- [Enable-NetAdapterBinding](#)

- [Enable-NetAdapterChecksumOffload](#)
- [Enable-NetAdapterIPSecOffload](#)
- [Enable-NetAdapterLso](#)
- [Enable-NetAdapterPowerManagement](#)
- [Enable-NetAdapterQos](#)
- [Enable-NetAdapterRDMA](#)
- [Enable-NetAdapterSriov](#)

For more information, see [Network Adapter Cmdlets in Windows PowerShell](#).

For links to all topics in this guide, see [Network Subsystem Performance Tuning](#).

Configure the Order of Network Interfaces

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

In Windows Server 2016 and Windows 10, you can use the interface metric to configure the order of network interfaces.

This is different than in previous versions of Windows and Windows Server, which allowed you to configure the binding order of network adapters by using either the user interface or the commands

INetCfgComponentBindings::MoveBefore and **INetCfgComponentBindings::MoveAfter**. These two methods for ordering network interfaces are not available in Windows Server 2016 and Windows 10.

Instead, you can use the new method for setting the enumerated order of network adapters by configuring the interface metric of each adapter. You can configure the interface metric by using the [Set-NetIPInterface](#) Windows PowerShell command.

When network traffic routes are chosen and you have configured the **InterfaceMetric** parameter of the [Set-NetIPInterface](#) command, the overall metric that is used to determine the interface preference is the sum of the route metric and the interface metric. Typically, the interface metric gives preference to a particular interface, such as using wired if both wired and wireless are available.

The following Windows PowerShell command example shows use of this parameter.

```
Set-NetIPInterface -InterfaceIndex 12 -InterfaceMetric 15
```

The order in which adapters appear in a list is determined by the IPv4 or IPv6 interface metric. For more information, see [GetAdaptersAddresses](#) function.

For links to all topics in this guide, see [Network Subsystem Performance Tuning](#).

Performance Tuning Network Adapters

9/1/2018 • 7 minutes to read • [Edit Online](#)

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

You can use this topic to performance tune network adapters that are installed in computers that are running Windows Server 2016.

Determining the correct tuning settings for your network adapter depend on the following variables:

- The network adapter and its feature set
- The type of workload performed by the server
- The server hardware and software resources
- Your performance goals for the server

If your network adapter provides tuning options, you can optimize network throughput and resource usage to achieve optimum throughput based on the parameters described above.

The following sections describe some of your performance tuning options.

Enabling Offload Features

Turning on network adapter offload features is usually beneficial. Sometimes, however, the network adapter is not powerful enough to handle the offload capabilities with high throughput.

IMPORTANT

Do not use the offload features **IPsec Task Offload** or **TCP Chimney Offload**. These technologies are deprecated in Windows Server 2016, and might adversely affect server and networking performance. In addition, these technologies might not be supported by Microsoft in the future.

For example, enabling segmentation offload can reduce the maximum sustainable throughput on some network adapters because of limited hardware resources. However, if the reduced throughput is not expected to be a limitation, you should enable offload capabilities, even for this type of network adapter.

NOTE

Some network adapters require offload features to be independently enabled for send and receive paths.

Enabling Receive Side Scaling (RSS) for Web Servers

RSS can improve web scalability and performance when there are fewer network adapters than logical processors on the server. When all the web traffic is going through the RSS-capable network adapters, incoming web requests from different connections can be simultaneously processed across different CPUs.

It is important to note that due to the logic in RSS and Hypertext Transfer Protocol (HTTP) for load distribution, performance might be severely degraded if a non-RSS-capable network adapter accepts web traffic on a server that has one or more RSS-capable network adapters. In this circumstance, you should use RSS-capable network

adapters or disable RSS on the network adapter properties **Advanced Properties** tab. To determine whether a network adapter is RSS-capable, you can view the RSS information on the network adapter properties **Advanced Properties** tab.

RSS Profiles and RSS Queues

The default RSS predefined profile is NUMA Static, which changes the default behavior from previous versions of the operating system. To get started with RSS Profiles, you can review the available profiles to understand when they are beneficial and how they apply to your network environment and hardware.

For example, if you open Task Manager and review the logical processors on your server, and they seem to be underutilized for receive traffic, you can try increasing the number of RSS queues from the default of 2 to the maximum that is supported by your network adapter. Your network adapter might have options to change the number of RSS queues as part of the driver.

Increasing Network Adapter Resources

For network adapters that allow manual configuration of resources, such as receive and send buffers, you should increase the allocated resources.

Some network adapters set their receive buffers low to conserve allocated memory from the host. The low value results in dropped packets and decreased performance. Therefore, for receive-intensive scenarios, we recommend that you increase the receive buffer value to the maximum.

NOTE

If a network adapter does not expose manual resource configuration, it either dynamically configures the resources, or the resources are set to a fixed value that cannot be changed.

Enabling Interrupt Moderation

To control interrupt moderation, some network adapters expose different interrupt moderation levels, buffer coalescing parameters (sometimes separately for send and receive buffers), or both.

You should consider interrupt moderation for CPU-bound workloads, and consider the trade-off between the host CPU savings and latency versus the increased host CPU savings because of more interrupts and less latency. If the network adapter does not perform interrupt moderation, but it does expose buffer coalescing, increasing the number of coalesced buffers allows more buffers per send or receive, which improves performance.

Performance Tuning for Low Latency Packet Processing

Many network adapters provide options to optimize operating system-induced latency. Latency is the elapsed time between the network driver processing an incoming packet and the network driver sending the packet back. This time is usually measured in microseconds. For comparison, the transmission time for packet transmissions over long distances is usually measured in milliseconds (an order of magnitude larger). This tuning will not reduce the time a packet spends in transit.

Following are some performance tuning suggestions for microsecond-sensitive networks.

- Set the computer BIOS to **High Performance**, with C-states disabled. However, note that this is system and BIOS dependent, and some systems will provide higher performance if the operating system controls power management. You can check and adjust your power management settings from **Settings** or by using the **powercfg** command. For more information, see [Powercfg Command-Line Options](#)
- Set the operating system power management profile to **High Performance System**. Note that this will not work properly if the system BIOS has been set to disable operating system control of power management.
- Enable Static Offloads, for example, UDP Checksums, TCP Checksums, and Send Large Offload (LSO).

- Enable RSS if the traffic is multi-streamed, such as high-volume multicast receive.
- Disable the **Interrupt Moderation** setting for network card drivers that require the lowest possible latency. Remember, this can use more CPU time and it represents a tradeoff.
- Handle network adapter interrupts and DPCs on a core processor that shares CPU cache with the core that is being used by the program (user thread) that is handling the packet. CPU affinity tuning can be used to direct a process to certain logical processors in conjunction with RSS configuration to accomplish this. Using the same core for the interrupt, DPC, and user mode thread exhibits worse performance as load increases because the ISR, DPC, and thread contend for the use of the core.

System Management Interrupts

Many hardware systems use System Management Interrupts (SMI) for a variety of maintenance functions, including reporting of error correction code (ECC) memory errors, legacy USB compatibility, fan control, and BIOS controlled power management.

The SMI is the highest priority interrupt on the system and places the CPU in a management mode, which preempts all other activity while it runs an interrupt service routine, typically contained in BIOS.

Unfortunately, this can result in latency spikes of 100 microseconds or more.

If you need to achieve the lowest latency, you should request a BIOS version from your hardware provider that reduces SMIs to the lowest degree possible. These are frequently referred to as "low latency BIOS" or "SMI free BIOS." In some cases, it is not possible for a hardware platform to eliminate SMI activity altogether because it is used to control essential functions (for example, cooling fans).

NOTE

The operating system can exert no control over SMIs because the logical processor is running in a special maintenance mode, which prevents operating system intervention.

Performance Tuning TCP

You can performance tune TCP using the following items.

TCP Receive Window Auto-Tuning

Prior to Windows Server 2008, the network stack used a fixed-size receive-side window (65,535 bytes) that limited the overall potential throughput for connections. One of the most significant changes to the TCP stack is TCP receive window auto-tuning.

You can calculate the total throughput of a single connection when you use a fixed size TCP receive window as:

Total achievable throughput in bytes = TCP receive window size in bytes * (1 / connection latency in seconds)

For example, the total achievable throughput is only 51 Mbps on a connection with 10 ms latency (a reasonable value for a large corporate network infrastructure).

With auto-tuning, however, the receive-side window is adjustable, and it can grow to meet the demands of the sender. It is possible for a connection to achieve a full line rate of a 1 Gbps connection. Network usage scenarios that might have been limited in the past by the total achievable throughput of TCP connections can now fully use the network.

Deprecated TCP parameters

The following registry settings from Windows Server 2003 are no longer supported, and are ignored in later

versions.

All of these settings had the following registry location:

```
...
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters
...
```

- TcpWindowSize
- NumTcbTablePartitions
- MaxHashTableSize

Windows Filtering Platform

The Windows Filtering Platform (WFP) that was introduced in Windows Vista and Windows Server 2008 provides APIs to non-Microsoft independent software vendors (ISVs) to create packet processing filters. Examples include firewall and antivirus software.

NOTE

A poorly written WFP filter can significantly decrease a server's networking performance. For more information, see [Porting Packet-Processing Drivers and Apps to WFP](#) in the Windows Dev Center.

For links to all topics in this guide, see [Network Subsystem Performance Tuning](#).

Network-Related Performance Counters

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

This topic lists the counters that are relevant to managing network performance, and contains the following sections.

- [Resource Utilization](#)
- [Potential Network Problems](#)
- [Receive Side Coalescing \(RSC\) performance](#)

Resource Utilization

The following performance counters are relevant to network resource utilization.

- IPv4, IPv6
 - Datagrams Received/sec
 - Datagrams Sent/sec
- TCPv4, TCPv6
 - Segments Received/sec
 - Segments Sent/sec
 - Segments Retransmitted/sec
- Network Interface(*), Network Adapter(*)
 - Bytes Received/sec
 - Bytes Sent/sec
 - Packets Received/sec
 - Packets Sent/sec
 - Output Queue Length

This counter is the length of the output packet queue (in packets). If this is longer than 2, delays occur.

You should find the bottleneck and eliminate it if you can. Because NDIS queues the requests, this length should always be 0.

- Processor Information
 - % Processor Time
 - Interrupts/sec
 - DPCs Queued/sec

This counter is an average rate at which DPCs were added to the logical processor's DPC queue. Each logical processor has its own DPC queue. This counter measures the rate at which DPCs are

added to the queue, not the number of DPCs in the queue. It displays the difference between the values that were observed in the last two samples, divided by the duration of the sample interval.

Potential Network Problems

The following performance counters are relevant to potential network problems.

- Network Interface(*), Network Adapter(*)
 - Packets Received Discarded
 - Packets Received Errors
 - Packets Outbound Discarded
 - Packets Outbound Errors
- WFPv4, WFPv6
 - Packets Discarded/sec
- UDPv4, UDPv6
 - Datagrams Received Errors
- TCPv4, TCPv6
 - Connection Failures
 - Connections Reset
- Network QoS Policy
 - Packets dropped
 - Packets dropped/sec
- Per Processor Network Interface Card Activity
 - Low Resource Receive Indications/sec
 - Low Resource Received Packets/sec
- Microsoft Winsock BSP
 - Dropped Datagrams
 - Dropped Datagrams/sec
 - Rejected Connections
 - Rejected Connections/sec

Receive Side Coalescing (RSC) performance

The following performance counters are relevant to RSC performance.

- Network Adapter(*)
 - TCP Active RSC Connections
 - TCP RSC Average Packet Size
 - TCP RSC Coalesced Packets/sec
 - TCP RSC Exceptions/sec

For links to all topics in this guide, see [Network Subsystem Performance Tuning](#).

Performance Tools for Network Workloads

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to learn about performance tools.

This topic contains sections about the Client to Server Traffic tool, TCP/IP Window Size, and Microsoft Server Performance Advisor.

Client to Server Traffic tool

The Client to Server Traffic (ctsTraffic) tool provides you with the ability to create and verify network traffic.

For more information, and to download the tool, see [ctsTraffic \(Client-To-Server Traffic\)](#).

TCP/IP Window Size

For 1 GB adapters, the settings shown in the previous table should provide good throughput because NTttcp sets the default TCP window size to 64 K through a specific logical processor option (SO_RECVBUF) for the connection. This provides good performance on a low-latency network.

In contrast, for high-latency networks or for 10 GB adapters, the default TCP window size value for NTttcp yields less than optimal performance. In both cases, you must adjust the TCP window size to allow for the larger bandwidth delay product.

You can statically set the TCP window size to a large value by using the **-rb** option. This option disables TCP Window Auto-Tuning, and we recommend using it only if the user fully understands the resultant change in TCP/IP behavior. By default, the TCP window size is set at a sufficient value and adjusts only under heavy load or over high-latency links.

Microsoft Server Performance Advisor

Microsoft Server Performance Advisor (SPA) helps IT administrators collect metrics to identify, compare, and diagnose potential performance issues in a Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2, or Windows Server 2008 deployment.

SPA generates comprehensive diagnostic reports and charts, and it provides recommendations to help you quickly analyze issues and develop corrective actions.

For more information and to download the advisor, see [Microsoft Server Performance Advisor](#) in the Windows Hardware Dev Center.

For links to all topics in this guide, see [Network Subsystem Performance Tuning](#).

NIC Teaming

9/21/2018 • 10 minutes to read • [Edit Online](#)

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

In this topic, we give you an overview of Network Interface Card (NIC) Teaming in Windows Server 2016. NIC Teaming allows you to group between one and 32 physical Ethernet network adapters into one or more software-based virtual network adapters. These virtual network adapters provide fast performance and fault tolerance in the event of a network adapter failure.

IMPORTANT

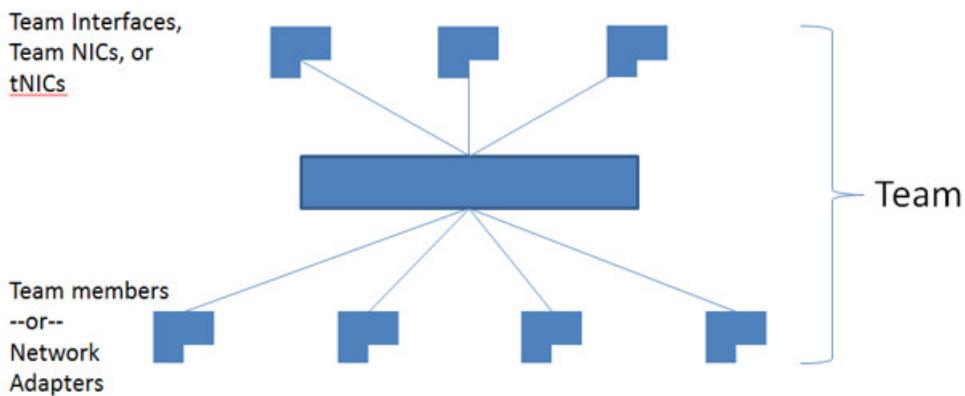
You must install NIC Team member network adapters in the same physical host computer.

TIP

A NIC team that contains only one network adapter cannot provide load balancing and failover. However, with one network adapter, you can use NIC Teaming for separation of network traffic when you are also using virtual Local Area Networks (VLANs).

When you configure network adapters into a NIC team, they connect into the NIC teaming solution common core, which then presents one or more virtual adapters (also called team NICs [tNICs] or team interfaces) to the operating system.

Since Windows Server 2016 supports up to 32 team interfaces per team, there are a variety of algorithms that distribute outbound traffic (load) between the NICs. The following illustration depicts a NIC Team with multiple tNICs.



Also, you can connect your teamed NICs to the same switch or different switches. If you connect NICs to different switches, both switches must be on the same subnet.

Availability

NIC Teaming is available in all versions of Windows Server 2016. You can use a variety of tools to manage NIC Teaming from computers running a client operating system, such as:

- Windows PowerShell cmdlets
- Remote Desktop
- Remote Server Administration Tools

Supported and Unsupported NICs

You can use any Ethernet NIC that has passed the Windows Hardware Qualification and Logo test (WHQL tests) in a NIC Team in Windows Server 2016.

You can not place the following NICs in a NIC team:

- Hyper-V virtual network adapters that are Hyper-V Virtual Switch ports exposed as NICs in the host partition.

IMPORTANT

Do not place Hyper-V virtual NICs exposed in the host partition (vNICs) in a team. Teaming of vNICs inside of the host partition is not supported in any configuration. Attempts to team vNICs might cause a complete loss of communication if network failures occur.

- Kernel debug network adapter (KDNIC).
- NICs used for network boot.
- NICs that use technologies other than Ethernet, such as WWAN, WLAN/Wi-Fi, Bluetooth, and Infiniband, including Internet Protocol over Infiniband (IPoIB) NICs.

Compatibility

NIC teaming is compatible with all networking technologies in Windows Server 2016 with the following exceptions.

- **Single-root I/O virtualization (SR-IOV).** For SR-IOV, data is delivered directly to the NIC without passing it through the networking stack (in the host operating system, in the case of virtualization). Therefore, it is not possible for the NIC team to inspect or redirect the data to another path in the team.
- **Native host Quality of Service (QoS).** When you set QoS policies on a native or host system, and those policies invoke minimum bandwidth limitations, the overall throughput for a NIC team is less than it would be without the bandwidth policies in place.
- **TCP Chimney.** TCP Chimney is not supported with NIC teaming because TCP Chimney offloads the entire networking stack directly to the NIC.
- **802.1X Authentication.** You should not use 802.1X Authentication with NIC Teaming because some switches do not permit the configuration of both 802.1X Authentication and NIC Teaming on the same port.

To learn about using NIC Teaming within virtual machines (VMs) that run on a Hyper-V host, see

Virtual Machine Queues (VMQs)

VMQs is a NIC feature that allocates a queue for each VM. Anytime you have Hyper-V enabled; you must also enable VMQ. In Windows Server 2016, VMQs use NIC Switch vPorts with a single queue assigned to the vPort to provide the same functionality.

Depending on the switch configuration mode and the load distribution algorithm, NIC teaming presents either the smallest number of available and supported queues by any adapter in the team (Min-Queues mode) or the total number of queues available across all team members (Sum-of-Queues mode).

If the team is in Switch-Independent teaming mode and you set the load distribution to Hyper-V Port mode or Dynamic mode, the number of queues reported is the sum of all the queues available from the team members (Sum-of-Queues mode). Otherwise, the number of queues reported is the smallest number of queues supported

by any member of the team (Min-Queues mode).

Here's why:

- When the switch-independent team is in Hyper-V Port mode or Dynamic mode the inbound traffic for a Hyper-V switch port (VM) always arrives on the same team member. The host can predict/control which member receives the traffic for a particular VM so NIC Teaming can be more thoughtful about which VMQ Queues to allocate on a particular team member. NIC Teaming, working with the Hyper-V switch, sets the VMQ for a VM on precisely one team member and know that inbound traffic hits that queue.
- When the team is in any switch dependent mode (static teaming or LACP teaming), the switch that the team is connected to controls the inbound traffic distribution. The host's NIC Teaming software can't predict which team member gets the inbound traffic for a VM and it may be that the switch distributes the traffic for a VM across all team members. As a result of the NIC Teaming software, working with the Hyper-V switch, programs a queue for the VM on every team member, not just one team member.
- When the team is in switch-independent mode and uses address hash load balancing, the inbound traffic always comes in on one NIC (the primary team member) - all of it on just one team member. Since other team members aren't dealing with inbound traffic, they get programmed with the same queues as the primary member so that if the primary member fails, any other team member can be used to pick up the inbound traffic, and the queues are already in place.
- Most NICs have queues used for either Receive Side Scaling (RSS) or VMQ, but not at the same time. Some VMQ settings appear to be settings for RSS queues but are settings on the generic queues that both RSS and VMQ use depending on which feature is presently in use. Each NIC has, in its advanced properties, values for *RssBaseProcNumber and *MaxRssProcessors. Following are a few VMQ settings that provide better system performance.
- Ideally, each NIC should have the *RssBaseProcNumber set to an even number greater than or equal to two (2). The first physical processor, Core 0 (logical processors 0 and 1), typically does most of the system processing so the network processing should steer away from this physical processor. Some machine architectures don't have two logical processors per physical processor, so for such machines, the base processor should be greater than or equal to 1. If in doubt assume your host is using a 2 logical processor per physical processor architecture.
- If the team is in Sum-of-Queues mode the team members' processors should be non-overlapping. For example, in a 4-core host (8 logical processors) with a team of 2 10Gbps NICs, you could set the first one to use the base processor of 2 and to use 4 cores; the second would be set to use base processor 6 and use 2 cores.
- If the team is in Min-Queues mode the processor sets used by the team members must be identical.

Hyper-V Network Virtualization (HNV)

NIC Teaming is fully compatible with Hyper-V Network Virtualization (HNV). The HNV management system provides information to the NIC Teaming driver that allows NIC Teaming to distribute the load in a way that optimizes HNV traffic.

Live Migration

NIC Teaming in VMs does not affect Live Migration. The same rules exist for Live Migration whether or not configuring NIC Teaming in the VM.

Virtual Local Area Networks (VLANs)

When you use NIC Teaming, creating multiple team interfaces allows a host to connect to different VLANs at the

same time. Configure your environment using the following guidelines:

- Before you enable NIC Teaming, configure the physical switch ports connected to the teaming host to use trunk (promiscuous) mode. The physical switch should pass all traffic to the host for filtering without modifying the traffic.
- Do not configure VLAN filters on the NICs by using the NIC advanced properties settings. Let the NIC Teaming software or the Hyper-V Virtual Switch (if present) perform VLAN filtering.

Use VLANs with NIC Teaming in a VM

When a team connects to a Hyper-V Virtual Switch, all VLAN segregation must be done in the Hyper-V Virtual Switch rather than in NIC Teaming.

Plan to use VLANs in a VM configured with a NIC Team using the following guidelines:

- The preferred method of supporting multiple VLANs in a VM is to configure the VM with multiple ports on the Hyper-V Virtual Switch and associate each port with a VLAN. Never team these ports in the VM because doing so causes network communication problems.
- If the VM has multiple SR-IOV Virtual Functions (VFs), ensure that they are on the same VLAN before teaming them in the VM. It's easily possible to configure the different VFs to be on different VLANs and doing so causes network communication problems.

Manage network interfaces and VLANs

If you must have more than one VLAN exposed into a guest operating system, consider renaming the Ethernet interfaces to clarify VLAN assigned to the interface. For example, if you associate **Ethernet** interface with VLAN 12 and the **Ethernet 2** interface with VLAN 48, rename the interface Ethernet to **EthernetVLAN12** and the other to **EthernetVLAN48**.

Rename interfaces by using the Windows PowerShell command **Rename-NetAdapter** or by performing the following procedure:

1. In Server Manager, in **Properties** for the network adapter you want to rename, click the link to the right of the network adapter name.
2. Right-click the network adapter that you want to rename, and select **Rename**.
3. Type the new name for the network adapter and press ENTER.

Virtual Machines (VMs)

If you want to use NIC Teaming in a VM, you must connect the virtual network adapters in the VM to external Hyper-V Virtual Switches only. Doing this allows the VM to sustain network connectivity even in the circumstance when one of the physical network adapters connected to one virtual switch fails or gets disconnected. Virtual network adapters connected to internal or private Hyper-V Virtual Switches are not able to connect to the switch when they are in a team, and networking fails for the VM.

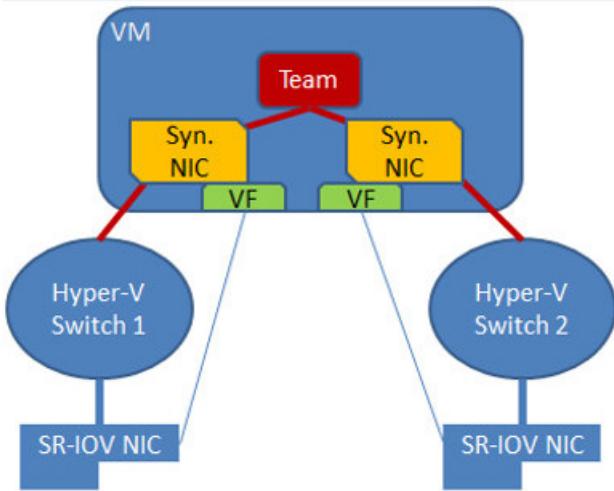
NIC Teaming in Windows Server 2016 supports teams with two members in VMs. You can create larger teams, but there is no support for larger teams. Every team member must connect to a different external Hyper-V Virtual Switch, and the VM's networking interfaces must be configured to allow teaming.

If you are configuring a NIC Team in a VM, you must select a **Teaming mode of Switch Independent** and a **Load balancing mode of Address Hash**.

SR-IOV-Capable Network Adapters

A NIC team in or under the Hyper-V host cannot protect SR-IOV traffic because it doesn't go through the Hyper-V Switch. With the VM NIC Teaming option, you can configure two external Hyper-V Virtual Switches, each

connected to its own SR-IOV-capable NIC.



Each VM can have a virtual function (VF) from one or both SR-IOV NICs and, in the event of a NIC disconnect, failover from the primary VF to the backup adapter (VF). Alternately, the VM may have a VF from one NIC and a non-VF vmNIC connected to another virtual switch. If the NIC associated with the VF gets disconnected, the traffic can failover to the other switch without loss of connectivity.

Because failover between NICs in a VM might result in traffic sent with the MAC address of the other vmNIC, each Hyper-V Virtual Switch port associated with a VM using NIC Teaming must be set to allow teaming.

Related topics

- [NIC Teaming MAC address use and management](#): When you configure a NIC Team with switch independent mode and either address hash or dynamic load distribution, the team uses the media access control (MAC) address of the primary NIC Team member on outbound traffic. The primary NIC Team member is a network adapter selected by the operating system from the initial set of team members.
- [NIC Teaming settings](#): In this topic, we give you an overview of the NIC Team properties such as teaming and load balancing modes. We also give you details about the Standby adapter setting and the Primary team interface property. If you have at least two network adapters in a NIC Team, you do not need to designate a Standby adapter for fault tolerance.
- [Create a new NIC Team on a host computer or VM](#): In this topic, you create a new NIC Team on a host computer or in a Hyper-V virtual machine (VM) running Windows Server 2016.
- [Troubleshooting NIC Teaming](#): In this topic, we discuss ways to troubleshoot NIC Teaming, such as hardware, physical switch securities, and disabling or enabling network adapters using Windows PowerShell.

NIC Teaming MAC address use and management

9/21/2018 • 4 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016

When you configure a NIC Team with switch independent mode and either address hash or dynamic load distribution, the team uses the media access control (MAC) address of the primary NIC Team member on outbound traffic. The primary NIC Team member is a network adapter selected by the operating system from the initial set of team members. It is the first team member to bind to the team after you create it or after the host computer is restarted. Because the primary team member might change in a non-deterministic manner at each boot, NIC disable/enable action, or other reconfiguration activities, the primary team member might change, and the MAC address of the team might vary.

In most situations this doesn't cause problems, but there are a few cases where issues might arise.

If the primary team member is removed from the team and then placed into operation there may be a MAC address conflict. To resolve this conflict, disable and then enable the team interface. The process of disabling and then enabling the team interface causes the interface to select a new MAC address from the remaining team members, thereby eliminating the MAC address conflict.

You can set the MAC address of the NIC team to a specific MAC address by setting it in the primary team interface, just as you can do when configuring the MAC address of any physical NIC.

MAC address use on transmitted packets

When you configure a NIC Team in switch independent mode and either address hash or dynamic load distribution, the packets from a single source (such as a single VM) is simultaneously distributed across multiple team members. To prevent the switches from getting confused and to prevent MAC flapping alarms, the source MAC address is replaced with a different MAC address on the frames transmitted on team members other than the primary team member. Because of this, each team member uses a different MAC address, and MAC address conflicts are prevented unless and until failure occurs.

When a failure is detected on the primary NIC, the NIC Teaming software starts using the primary team member's MAC address on the team member that is chosen to serve as the temporary primary team member (i.e., the one that will now appear to the switch as the primary team member). This change only applies to traffic that was going to be sent on the primary team member with the primary team member's MAC address as its source MAC address. Other traffic continues to be sent with whatever source MAC address it would have used prior to the failure.

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

1. In Switch Independent mode with Address Hash distribution

- All ARP and NS packets are sent on the primary team member
- All traffic sent on NICs other than the primary team member are sent with the source MAC address modified to match the NIC on which they are sent
- All traffic sent on the primary team member is sent with the original source MAC address (which may be the team's source MAC address)

2. In Switch Independent mode with Hyper-V Port distribution

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

3. In Switch Independent mode with Dynamic distribution

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

4. In Switch Dependent mode (all distributions)

- No source MAC address replacement is performed

Related topics

- [NIC Teaming](#): In this topic, we give you an overview of Network Interface Card (NIC) Teaming in Windows Server 2016. NIC Teaming allows you to group between one and 32 physical Ethernet network adapters into one or more software-based virtual network adapters. These virtual network adapters provide fast performance and fault tolerance in the event of a network adapter failure.
- [NIC Teaming settings](#): In this topic, we give you an overview of the NIC Team properties such as teaming and load balancing modes. We also give you details about the Standby adapter setting and the Primary team interface property. If you have at least two network adapters in a NIC Team, you do not need to designate a Standby adapter for fault tolerance.
- [Create a new NIC Team on a host computer or VM](#): In this topic, you create a new NIC Team on a host computer or in a Hyper-V virtual machine (VM) running Windows Server 2016.
- [Troubleshooting NIC Teaming](#): In this topic, we discuss ways to troubleshoot NIC Teaming, such as hardware, physical switch securities, and disabling or enabling network adapters using Windows PowerShell.

Create a new NIC Team on a host computer or VM

9/21/2018 • 8 minutes to read • [Edit Online](#)

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

In this topic, you create a new NIC Team on a host computer or in a Hyper-V virtual machine (VM) running Windows Server 2016.

Network configuration requirements

Before you can create a new NIC Team, you must deploy a Hyper-V host with two network adapters that connect to different physical switches. You must also configure the network adapters with IP addresses that are from the same IP address range.

The physical switch, Hyper-V Virtual Switch, local area network (LAN), and NIC Teaming requirements for creating a NIC Team in a VM are:

- The computer running Hyper-V must have two or more network adapters.
- If connecting the network adapters to multiple physical switches, the physical switches must be on the same Layer 2 subnet.
- You must use Hyper-V Manager or Windows PowerShell to create two external Hyper-V Virtual Switches, each connected to a different physical network adapter.
- The VM must connect to both external virtual switches you create.
- NIC Teaming, in Windows Server 2016, supports teams with two members in VMs. You can create larger teams, but there is no support.
- If you are configuring a NIC Team in a VM, you must select a **Teaming mode** of *Switch Independent* and a **Load balancing mode** of *Address Hash*.

Step 1. Configure the physical and virtual network

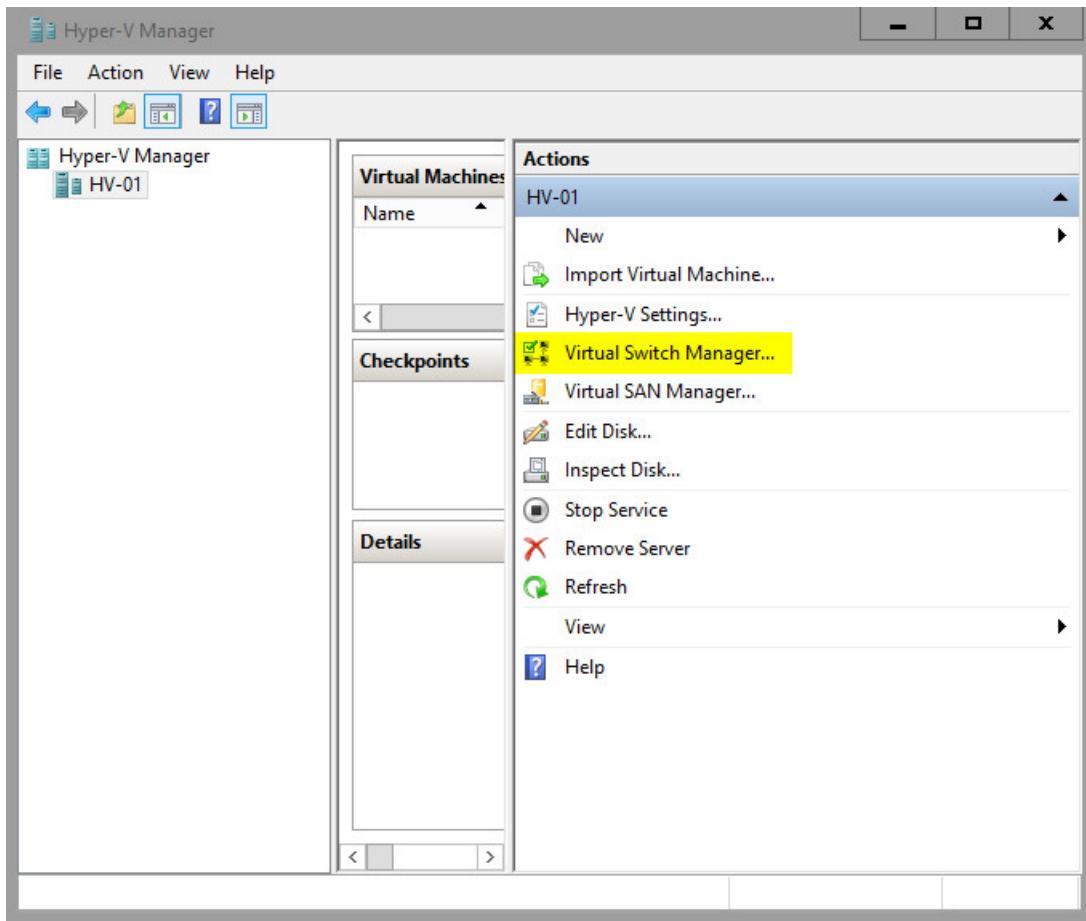
In this procedure, you create two external Hyper-V Virtual Switches, connect a VM to the switches, and then configure the VM connections to the switches.

Prerequisites

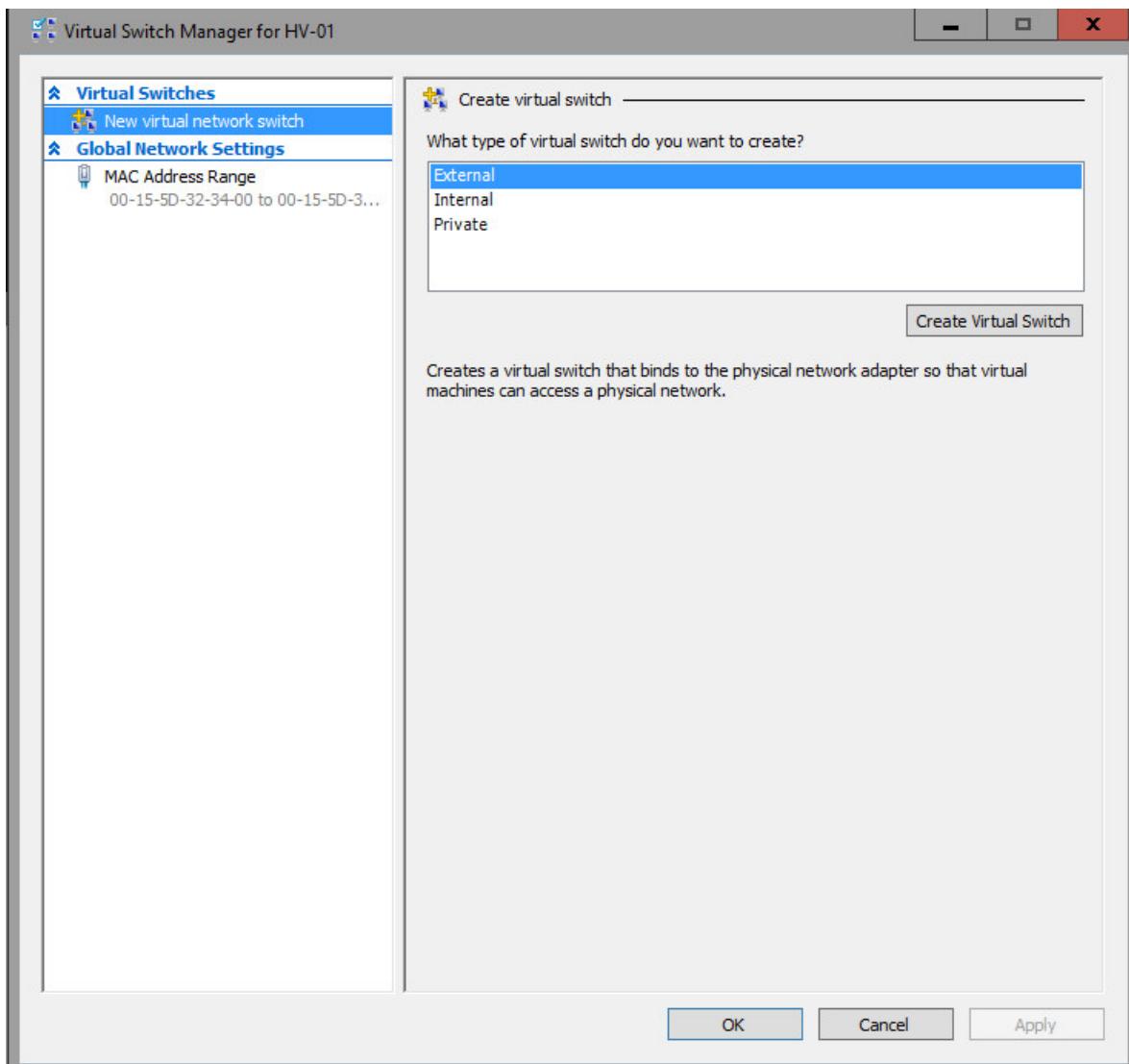
You must have membership in **Administrators**, or equivalent.

Procedure

1. On the Hyper-V host, open Hyper-V Manager, and under Actions, click **Virtual Switch Manager**.



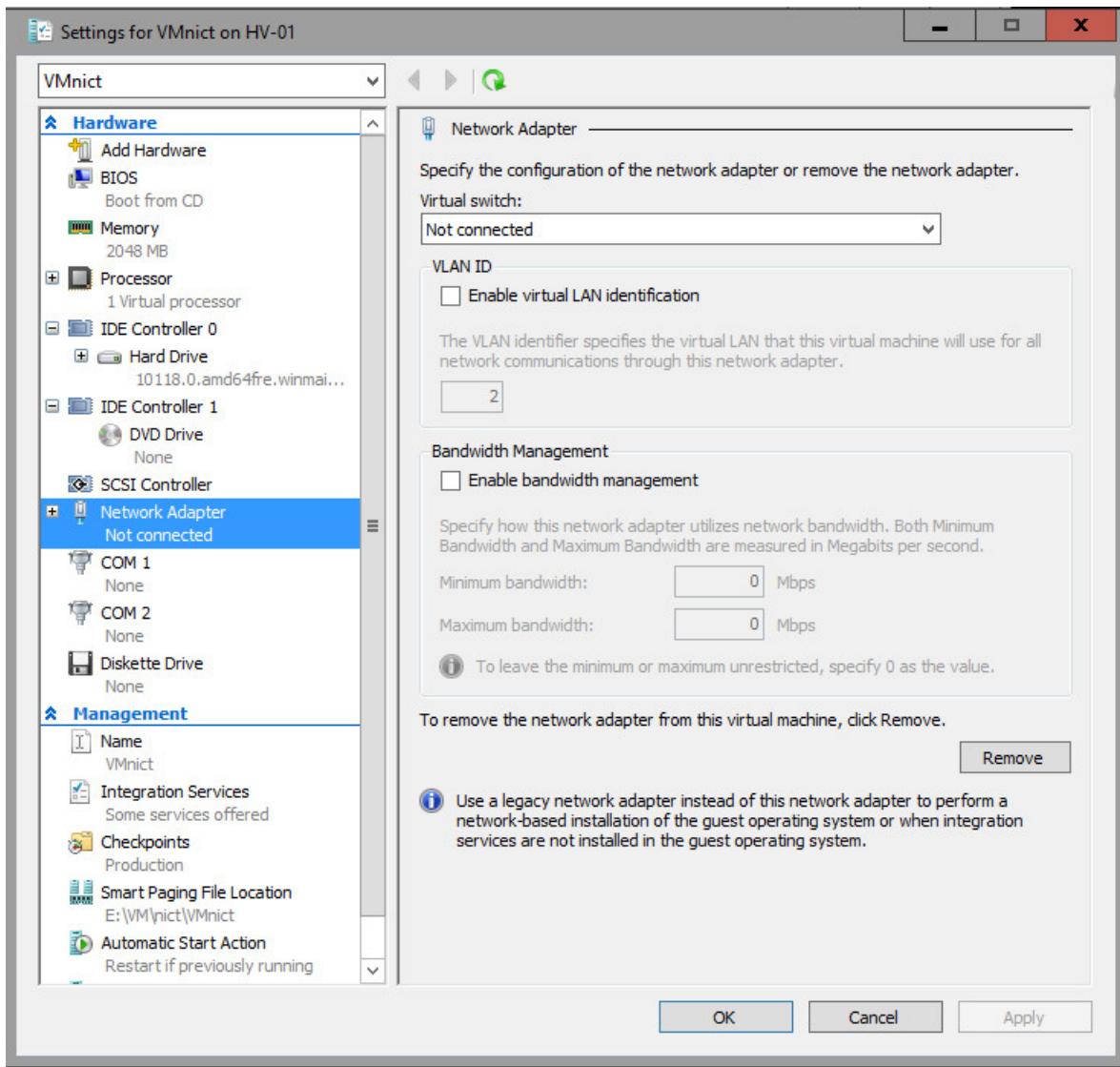
2. In Virtual Switch Manager, make sure **External** is selected, and then click **Create Virtual Switch**.



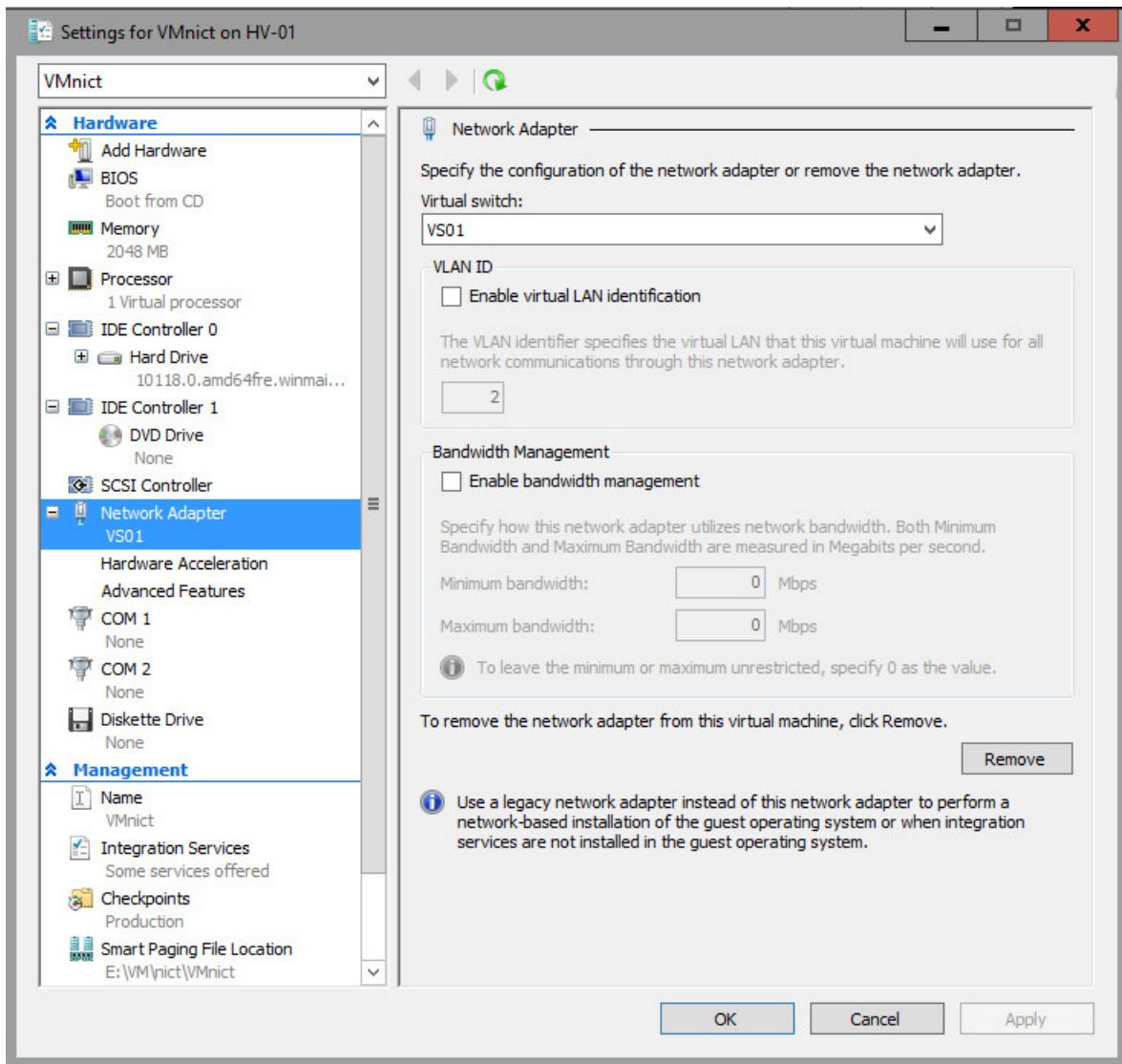
3. In Virtual Switch Properties, type a **Name** for the virtual switch, and add **Notes** as needed.
4. In **Connection type**, in **External network**, select the physical network adapter to which you want to attach the virtual switch.
5. Configure additional switch properties for your deployment, and then click **OK**.
6. Create a second external virtual switch by repeating the previous steps. Connect the second external switch to a different network adapter.
7. In Hyper-V Manager, under **Virtual Machines**, right-click the VM that you want to configure, and then click **Settings**.

The VM **Settings** dialog box opens.

8. Ensure that the VM is not started. If it is started, perform a shutdown before configuring the VM.
9. In **Hardware**, click **Network Adapter**.



10. In **Network Adapter** properties, select one of the virtual switches that you created in previous steps, and then click **Apply**.



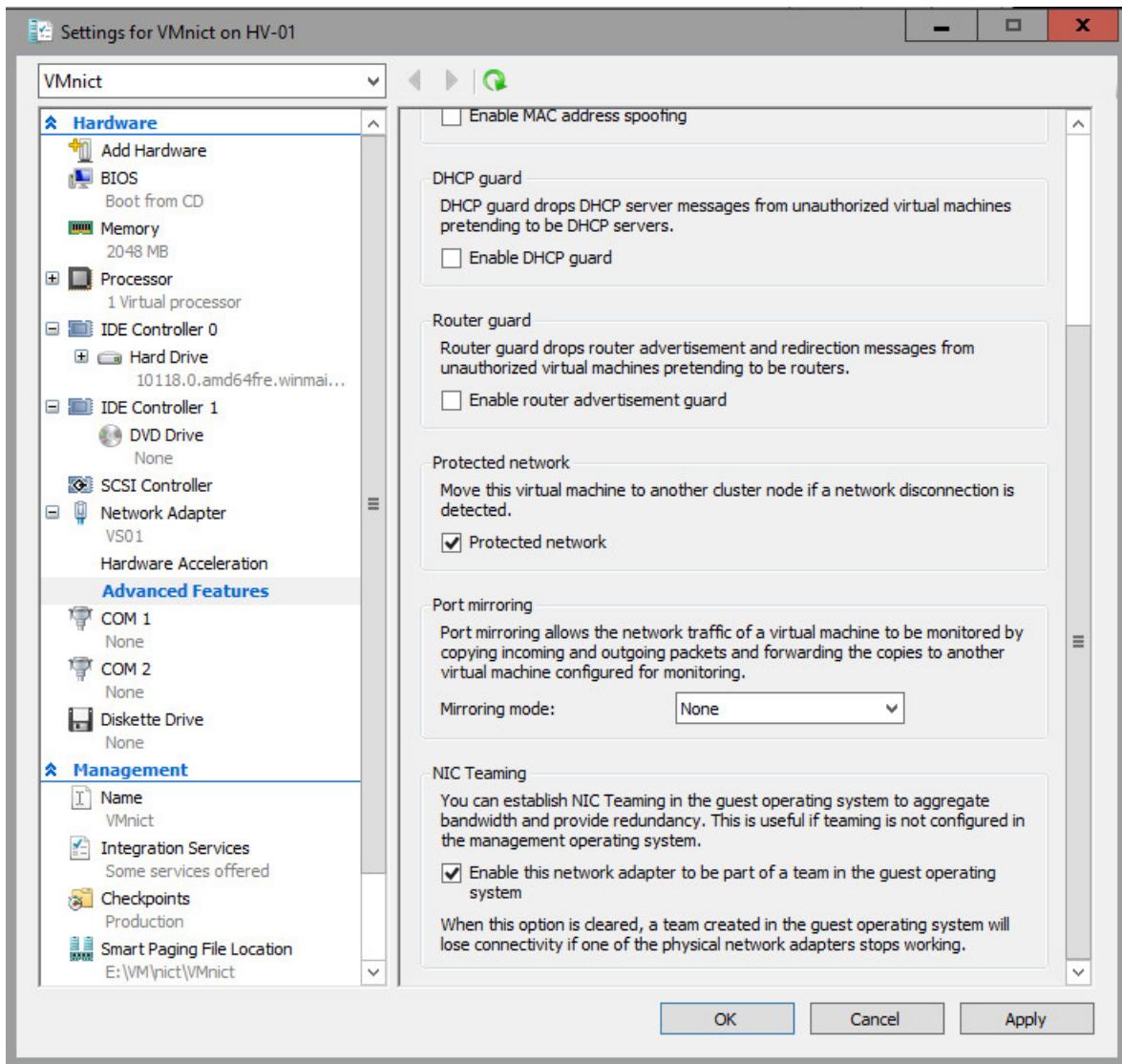
11. In **Hardware**, click to expand the plus sign (+) next to **Network Adapter**.
12. Click **Advanced Features** to enable NIC Teaming by using the graphical user interface.

TIP

You can also enable NIC Teaming with a Windows PowerShell command:

```
Set-VMNetworkAdapter -VMName <VMname> -AllowTeaming On
```

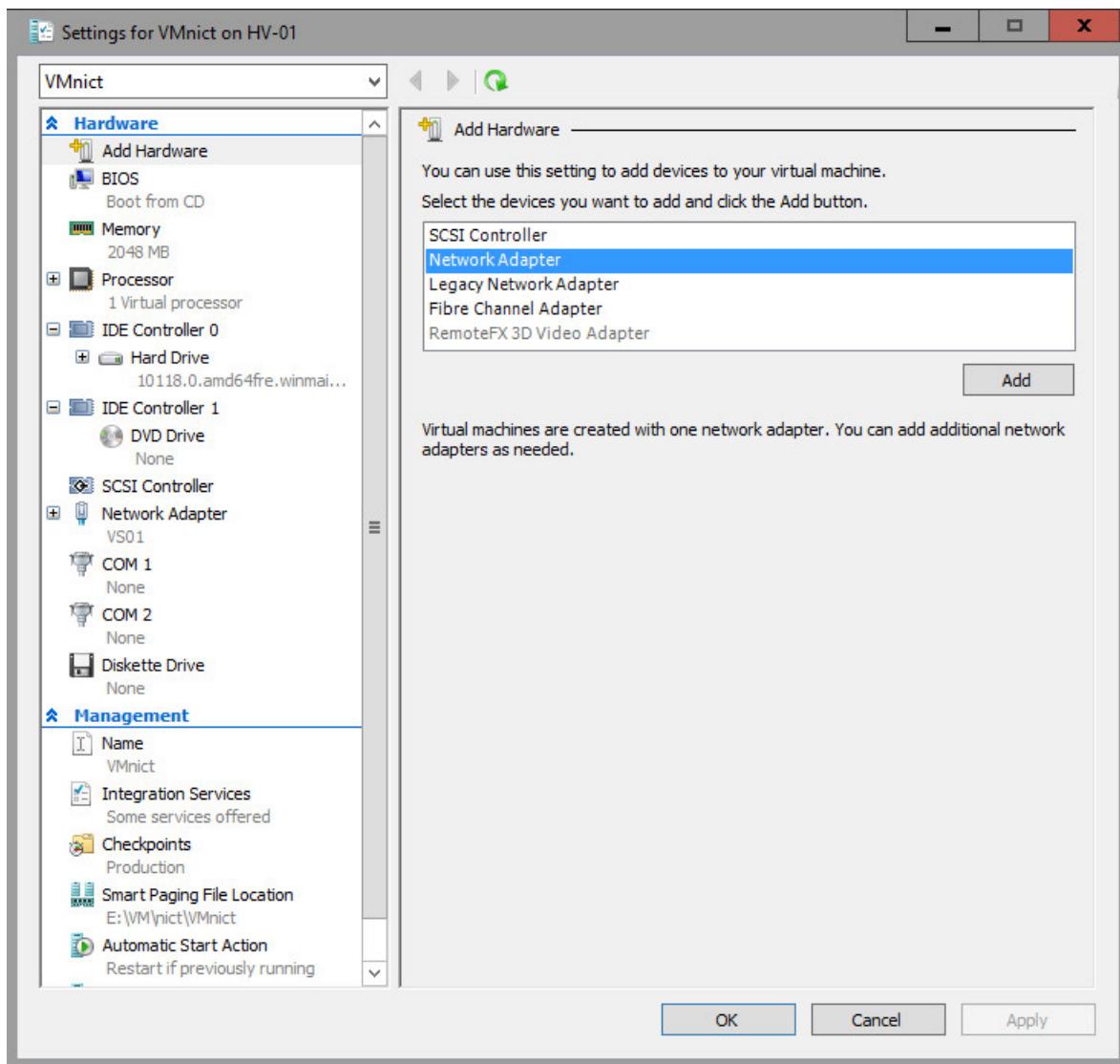
- a. Select **Dynamic** for MAC address.
- b. Click to select **Protected network**.
- c. Click to select **Enable this network adapter to be part of a team in the guest operating system**.
- d. Click **OK**.



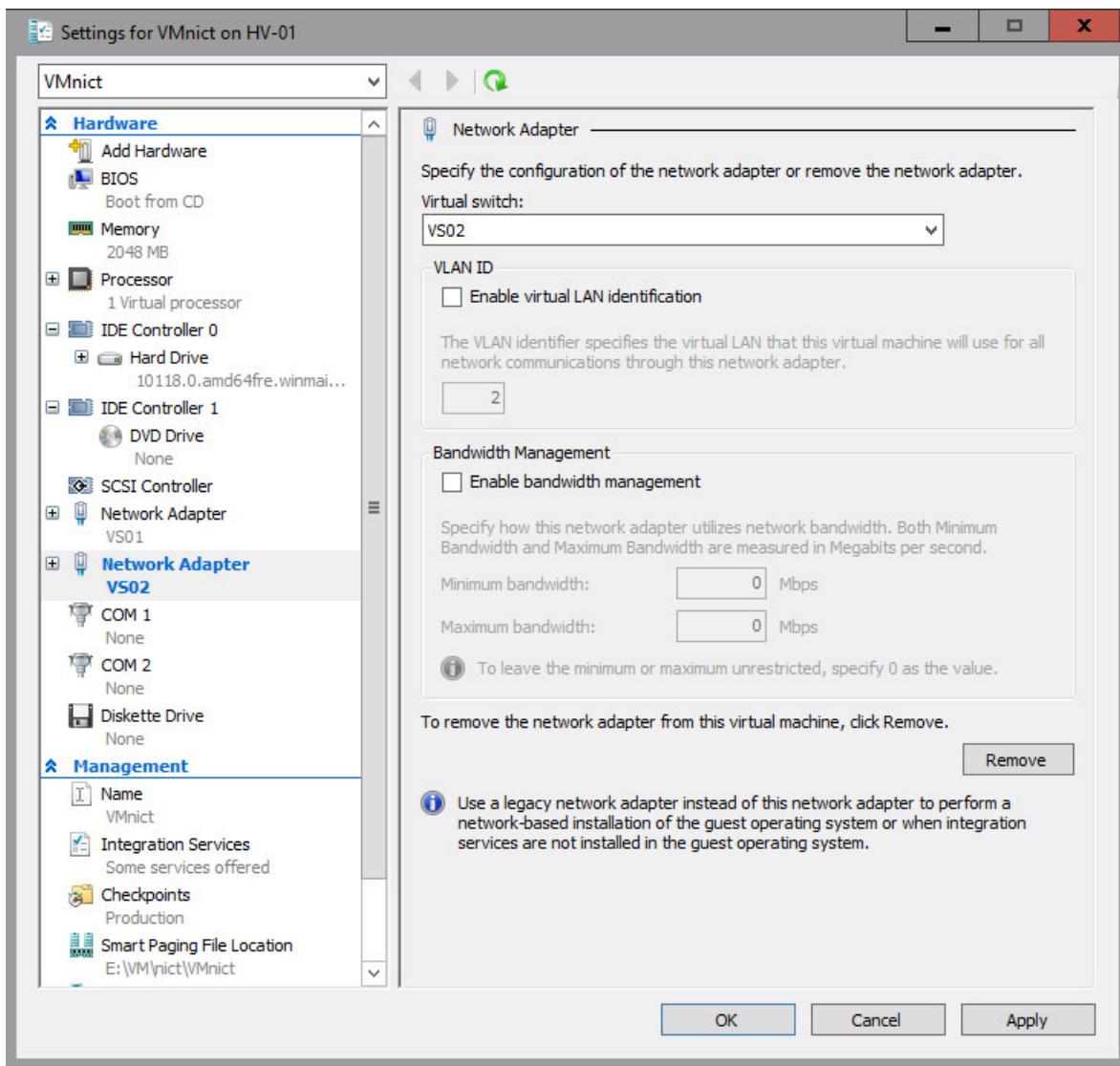
13. To add a second network adapter, in Hyper-V Manager, in **Virtual Machines**, right-click the same VM, and then click **Settings**.

The VM **Settings** dialog box opens.

14. In **Add Hardware**, click **Network Adapter**, and then click **Add**.



15. In **Network Adapter** properties, select the second virtual switch that you created in previous steps, and then click **Apply**.



16. In **Hardware**, click to expand the plus sign (+) next to **Network Adapter**.
17. Click **Advanced Features**, scroll down to **NIC Teaming**, and click to select **Enable this network adapter to be part of a team in the guest operating system**.
18. Click **OK**.

Congratulations! You have configured the physical and virtual network. Now you can proceed to creating a new NIC Team.

Step 2. Create a new NIC Team

When you create a new NIC Team, you must configure the NIC Team properties:

- Team name
- Member adapters
- Teaming mode
- Load balancing mode
- Standby adapter

You can also optionally configure the primary team interface and configure a virtual LAN (VLAN) number.

For more details on these settings, see [NIC Teaming settings](#).

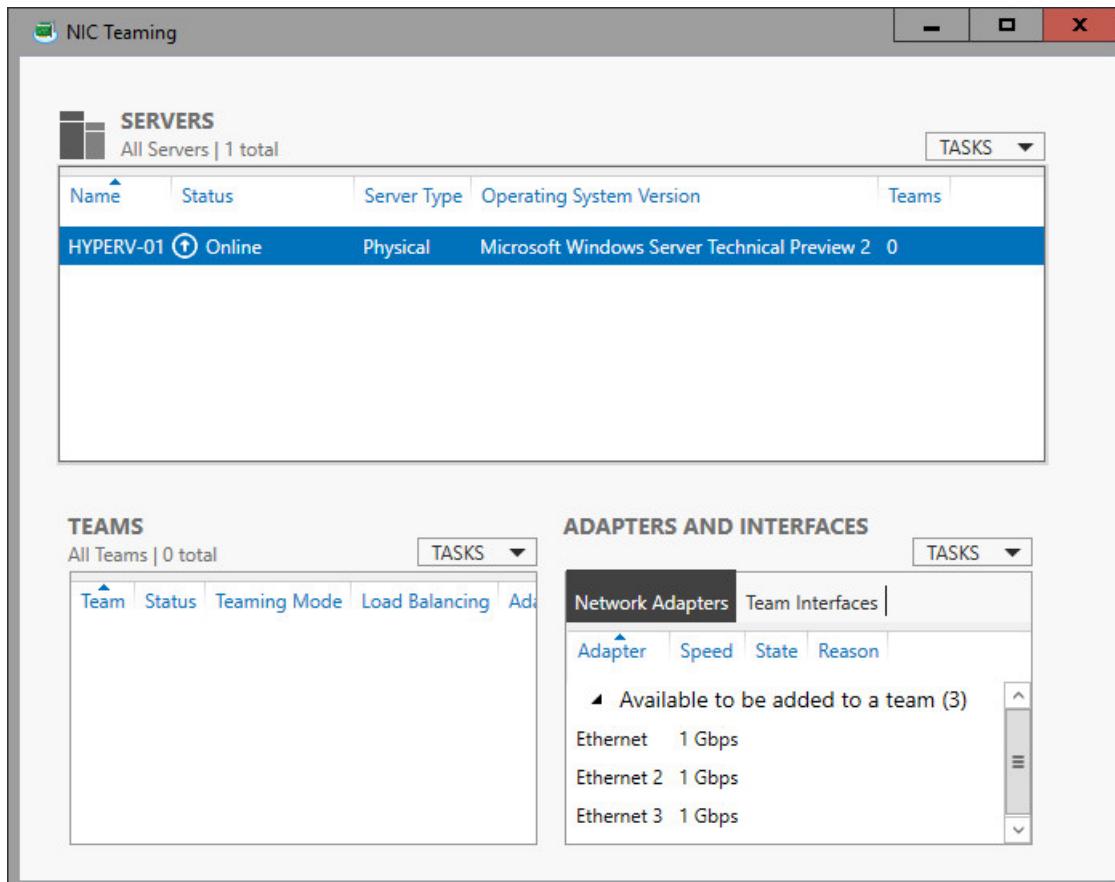
Prerequisites

You must have membership in **Administrators**, or equivalent.

Procedure

1. In Server Manager, click **Local Server**.
2. In the **Properties** pane, in the first column, locate **NIC Teaming**, and then click the **Disabled** link.

The **NIC Teaming** dialog box opens.



3. In **Adapters and Interfaces**, select the one or more network adapters that you want to add to a NIC Team.
4. Click **TASKS**, and click **Add to New Team**.

The **New team** dialog box opens and displays network adapters and team members.

5. In **Team name**, type a name for the new NIC Team, and then click **Additional properties**.
6. In **Additional properties**, select values for:

- **Teaming mode.** The options for Teaming mode are **Switch Independent** and **Switch Dependent**.
The Switch Dependent mode includes **Static Teaming** and **Link Aggregation Control Protocol (LACP)**.
 - **Switch Independent.** With Switch Independent mode, the switch or switches to which the NIC Team members are connected are unaware of the presence of the NIC team and do not determine how to distribute network traffic to NIC Team members - instead, the NIC Team distributes inbound network traffic across the NIC Team members.
 - **Switch Dependent.** With Switch Dependent modes, the switch to which the NIC Team members are connected determines how to distribute the inbound network traffic among the NIC Team members. The switch has complete independence to determine how to distribute the network traffic across the NIC Team members.

Static Teaming	Requires you to manually configure both the switch and the host to identify which links form the team. Because this is a statically configured solution, there is no additional protocol to assist the switch and the host to identify incorrectly plugged cables or other errors that could cause the team to fail to perform. This mode is typically supported by server-class switches.
Link Aggregation Control Protocol (LACP)	Unlike Static Teaming, LACP Teaming mode dynamically identifies links that are connected between the host and the switch. This dynamic connection enables the automatic creation of a team and, in theory but rarely in practice, the expansion and reduction of a team simply by the transmission or receipt of LACP packets from the peer entity. All server-class switches support LACP, and all require the network operator to administratively enable LACP on the switch port. When you configure a Teaming mode of LACP, NIC Teaming always operates in LACP's Active mode with a short timer. No option is presently available to modify the timer or change the LACP mode.

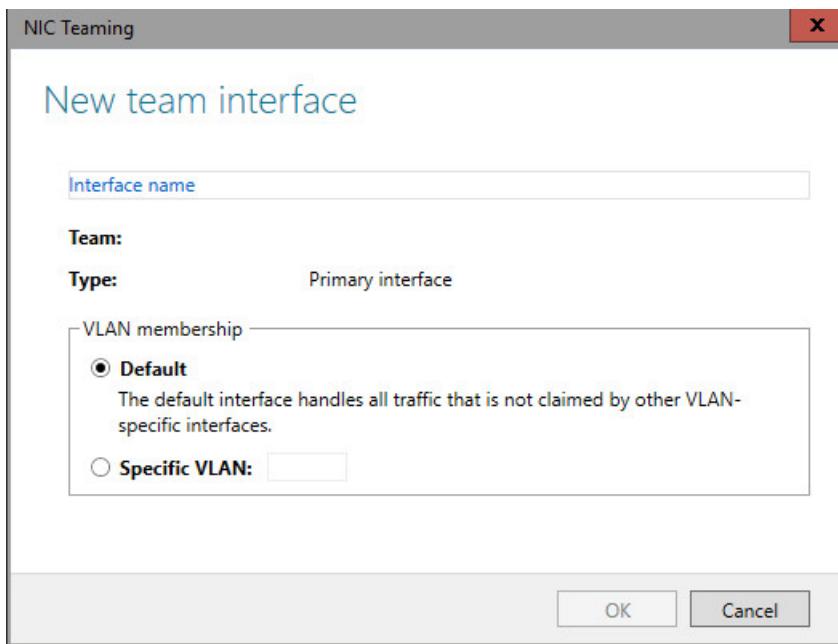
- **Load balancing mode.** The options for Load Balancing distribution mode are **Address Hash**, **Hyper-V Port**, and **Dynamic**.
 - **Address Hash.** With Address Hash, this mode creates a hash based on address components of the packet, which then get assigned to one of the available adapters. Usually, this mechanism alone is sufficient to create a reasonable balance across the available adapters.
 - **Hyper-V Port.** With Hyper-V Port, NIC Teams configured on Hyper-V hosts give VMs independent MAC addresses. The VMs MAC address or the VM ported connected to the Hyper-V switch, can be used to divide network traffic between NIC Team members. You cannot configure NIC Teams that you create within VMs with the Hyper-V Port load balancing mode. Instead, use the Address Hash mode.
 - **Dynamic.** With Dynamic, outbound loads are distributed based on a hash of the TCP ports and IP addresses. Dynamic mode also rebalances loads in real time so that a given outbound flow may move back and forth between team members. Inbound loads, on the other hand, get distributed the same way as Hyper-V Port. In a nutshell, Dynamic mode utilizes the best aspects of both Address Hash and Hyper-V Port and is the highest performing load balancing mode.
- **Standby adapter.** The options for Standby Adapter are **None (all adapters Active)** or your selection of a specific network adapter in the NIC Team that acts as a Standby adapter.

TIP

If you are configuring a NIC Team in a virtual machine (VM), you must select a **Teaming mode** of *Switch Independent* and a **Load balancing mode** of *Address Hash*.

7. If you want to configure the primary team interface name or assign a VLAN number to the NIC Team, click the link to the right of **Primary team interface**.

The **New team interface** dialog box opens.



8. Depending on your requirements, do one of the following:

- Provide a tNIC interface name.
- Configure VLAN membership: click **Specific VLAN** and type the VLAN information. For example, if you want to add this NIC Team to the accounting VLAN number 44, Type Accounting 44 - VLAN.

9. Click **OK**.

Congratulations! You've created a new NIC Team on a host computer or VM.

Related topics

- [NIC Teaming](#): In this topic, we give you an overview of Network Interface Card (NIC) Teaming in Windows Server 2016. NIC Teaming allows you to group between one and 32 physical Ethernet network adapters into one or more software-based virtual network adapters. These virtual network adapters provide fast performance and fault tolerance in the event of a network adapter failure.
- [NIC Teaming MAC address use and management](#): When you configure a NIC Team with switch independent mode and either address hash or dynamic load distribution, the team uses the media access control (MAC) address of the primary NIC Team member on outbound traffic. The primary NIC Team member is a network adapter selected by the operating system from the initial set of team members.
- [NIC Teaming settings](#): In this topic, we give you an overview of the NIC Team properties such as teaming and load balancing modes. We also give you details about the Standby adapter setting and the Primary team interface property. If you have at least two network adapters in a NIC Team, you do not need to designate a Standby adapter for fault tolerance.
- [Troubleshooting NIC Teaming](#): In this topic, we discuss ways to troubleshoot NIC Teaming, such as hardware, physical switch securities, and disabling or enabling network adapters using Windows PowerShell.

Troubleshooting NIC Teaming

9/21/2018 • 3 minutes to read • [Edit Online](#)

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

In this topic, we discuss ways to troubleshoot NIC Teaming, such as hardware and physical switch securities. When hardware implementations of standard protocols don't conform to specifications, NIC Teaming performance might be affected. Also, depending on the configuration, NIC Teaming may send packets from the same IP address with multiple MAC addresses tripping up security features on the physical switch.

Hardware that doesn't conform to specification

During normal operation, NIC Teaming may send packets from the same IP address, yet with multiple MAC addresses. According to protocol standards, the receivers of these packets must resolve the IP address of the host or VM to a specific MAC address rather than responding to the MAC address from the receiving packet. Clients that correctly implement the address resolution protocols (ARP and NDP) send packets with the correct destination MAC addresses—that is, the MAC address of the VM or host that owns that IP address.

Some embedded hardware does not correctly implement the address resolution protocols, and also might not explicitly resolve an IP address to a MAC address using ARP or NDP. For example, a storage area network (SAN) controller might perform in this manner. Non-conforming devices copy the source MAC address from a received packet and then use it as the destination MAC address in the corresponding outgoing packets, resulting in packets being sent to the wrong destination MAC address. Because of this, the packets are dropped by the Hyper-V Virtual Switch because they don't match any known destination.

If you are having trouble connecting to SAN controllers or other embedded hardware, you should take packet captures to determine if your hardware is correctly implementing ARP or NDP, and contact your hardware vendor for support.

Physical switch security features

Depending on the configuration, NIC Teaming may send packets from the same IP address with multiple source MAC addresses tripping up security features on the physical switch. For example, Dynamic ARP inspection or IP source guard, especially if the physical switch is not aware that the ports are part of a team, which occurs when you configure NIC Teaming in Switch Independent mode. Inspect the switch logs to determine if switch security features are causing connectivity problems.

Disabling and enabling network adapters by using Windows PowerShell

A common reason for a NIC Team to fail is that the team interface is disabled, and in many cases, by accident when running a sequence of commands. This particular sequence of commands does not enable all of the NetAdapters disabled because disabling all of the underlying physical members of NICs removes the NIC team interface.

In this case, the NIC team interface no longer shows in Get-NetAdapter, and because of this, **Enable-NetAdapter** * does not enable the NIC Team. The **Enable-NetAdapter** * command does, however, enable the member NICs, which then (after a short time) recreates the team interface. The team interface remains in the "disabled" state until re-enabled, allowing network traffic to begin flowing.

The following Windows PowerShell sequence of commands may disable the team interface by accident:

```
Disable-NetAdapter *  
Enable-NetAdapter *
```

Related topics

- [NIC Teaming](#): In this topic, we give you an overview of Network Interface Card (NIC) Teaming in Windows Server 2016. NIC Teaming allows you to group between one and 32 physical Ethernet network adapters into one or more software-based virtual network adapters. These virtual network adapters provide fast performance and fault tolerance in the event of a network adapter failure.
- [NIC Teaming MAC address use and management](#): When you configure a NIC Team with switch independent mode and either address hash or dynamic load distribution, the team uses the media access control (MAC) address of the primary NIC Team member on outbound traffic. The primary NIC Team member is a network adapter selected by the operating system from the initial set of team members.
- [NIC Teaming settings](#): In this topic, we give you an overview of the NIC Team properties such as teaming and load balancing modes. We also give you details about the Standby adapter setting and the Primary team interface property. If you have at least two network adapters in a NIC Team, you do not need to designate a Standby adapter for fault tolerance.

Quality of Service (QoS) Policy

9/1/2018 • 6 minutes to read • [Edit Online](#)

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

You can use QoS Policy as a central point of network bandwidth management across your entire Active Directory infrastructure by creating QoS profiles, whose settings are distributed with Group Policy.

NOTE

In addition to this topic, the following QoS Policy documentation is available.

- [Getting Started with QoS Policy](#)
- [Manage QoS Policy](#)
- [QoS Policy Frequently Asked Questions](#)

QoS policies are applied to a user login session or a computer as part of a Group Policy object (GPO) that you have linked to an Active Directory container, such as a domain, site, or organizational unit (OU).

QoS traffic management occurs below the application layer, which means that your existing applications do not need to be modified to benefit from the advantages that are provided by QoS policies.

Operating Systems that Support QoS Policy

You can use QoS policy to manage bandwidth for computers or users with the following Microsoft operating systems.

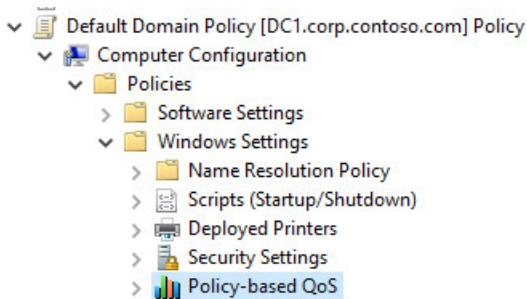
- Windows Server 2016
- Windows 10
- Windows Server 2012 R2
- Windows 8.1
- Windows Server 2012
- Windows 8
- Windows Server 2008 R2
- Windows 7
- Windows Server 2008
- Windows Vista

Location of QoS Policy in Group Policy

In Windows Server 2016 Group Policy Management Editor, the path to QoS Policy for Computer Configuration is the following.

Default Domain Policy | Computer Configuration | Policies | Windows Settings | Policy-based QoS

This path is illustrated in the following image.



In Windows Server 2016 Group Policy Management Editor, the path to QoS Policy for User Configuration is the following.

Default Domain Policy | User Configuration | Policies | Windows Settings | Policy-based QoS

By default no QoS policies are configured.

Why Use QoS Policy?

As traffic increases on your network, it is increasingly important for you to balance network performance with the cost of service - but network traffic is not normally easy to prioritize and manage.

On your network, mission-critical and latency-sensitive applications must compete for network bandwidth against lower priority traffic. At the same time, some users and computers with specific network performance requirements might require differentiated service levels.

The challenges of providing cost-effective, predictable network performance levels often first appear over wide area network (WAN) connections or with latency-sensitive applications, like voice over IP (VoIP) and video streaming. However, the end-goal of providing predictable network service levels applies to any network environment (for example, an Enterprises' local area network), and to more than VoIP applications, such as your company's custom line-of-business applications.

Policy-based QoS is the network bandwidth management tool that provides you with network control - based on applications, users, and computers.

When you use QoS Policy, your applications do not need to be written for specific application programming interfaces (APIs). This gives you the ability to use QoS with existing applications. Additionally, Policy-based QoS takes advantage of your existing management infrastructure, because Policy-based QoS is built into Group Policy.

Define QoS Priority Through a Differentiated Services Code Point (DSCP)

You can create QoS policies that define network traffic priority with a Differentiated Services Code Point (DSCP) value that you assign to different types of network traffic.

The DSCP allows you to apply a value (0–63) within the Type of Service (TOS) field in an IPv4 packet's header, and within the Traffic Class field in IPv6.

The DSCP value provides network traffic classification at the Internet Protocol (IP) level, which routers use to decide traffic queuing behavior.

For example, you can configure routers to place packets with specific DSCP values into one of three queues: high priority, best effort, or lower than best effort.

Mission-critical network traffic, which is in the high priority queue, has preference over other traffic.

Limit Network Bandwidth Use Per Application with Throttle Rate

You can also limit an application's outbound network traffic by specifying a throttle rate in QoS Policy.

A QoS policy that defines throttling limits determines the rate of outbound network traffic. For example, to manage WAN costs, an IT department might implement a service level agreement that specifies that a file server can never provide downloads beyond a specific rate.

Use QoS Policy to Apply DSCP Values and Throttle Rates

You can also use QoS Policy to apply DSCP values and throttle rates for outbound network traffic to the following:

- Sending application and directory path
- Source and destination IPv4 or IPv6 addresses or address prefixes
- Protocol - Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)
- Source and destination ports and port ranges (TCP or UDP)
- Specific groups of users or computers through deployment in Group Policy

By using these controls, you can specify a QoS policy with a DSCP value of 46 for a VoIP application, enabling routers to place VoIP packets in a low-latency queue, or you can use a QoS policy to throttle a set of servers' outbound traffic to 512 kilobytes per second (KBps) when sending from TCP port 443.

You can also apply QoS policy to a particular application that has special bandwidth requirements. For more information, see [QoS Policy Scenarios](#).

Advantages of QoS Policy

With QoS Policy, you can configure and enforce QoS policies that cannot be configured on routers and switches. QoS Policy provides the following advantages.

1. **Level of detail:** It is difficult to create user-level QoS policies on routers or switches, especially if the user's computer is either configured by using dynamic IP address assignment or if the computer is not connected to fixed switch or router ports, as is frequently the case with portable computers. In contrast, QoS Policy makes it easier to configure a user-level QoS policy on a domain controller and propagate the policy to the user's computer.
2. **Flexibility.** Regardless of where or how a computer connects to the network, QoS policy is applied - the computer can connect using WiFi or Ethernet from any location. For user-level QoS policies, the QoS policy is applied on any compatible device at any location where the user logs on.
3. **Security:** If your IT department encrypts users' traffic from end to end by using Internet Protocol security (IPsec), you cannot classify the traffic on routers based on any information above the IP layer in the packet (for example, a TCP port). However, by using QoS Policy, you can classify packets at the end device to indicate the priority of the packets in the IP header before the IP payloads are encrypted and the packets are sent.
4. **Performance:** Some QoS functions, such as throttling, are better performed when they are closer to the source. QoS Policy moves such QoS functions closest to the source.
5. **Manageability:** QoS Policy enhances network manageability in two ways:
 - a. Because it is based on Group Policy, you can use QoS Policy to configure and manage a set of user/computer QoS policies whenever necessary, and on one central domain-controller computer.
 - b. QoS Policy facilitates user/computer configuration by providing a mechanism to specify policies by Uniform Resource Locator (URL) instead of specifying policies based on the IP addresses of each of the servers where QoS policies need to be applied. For example, assume your network has a cluster of servers that share a common URL. By using QoS Policy, you can create one policy based on the common URL, instead of creating one policy for each server in the cluster, with each policy based on the IP address of each server.

For the next topic in this guide, see [Getting Started with QoS Policy](#).

Getting Started with QoS Policy

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the following topics to get started with Quality of Service (QoS) Policy.

- [How QoS Policy Works](#)
- [QoS Policy Architecture](#)
- [QoS Policy Scenarios](#)

For the first topic in this guide, see [Quality of Service \(QoS\) Policy](#).

How QoS Policy Works

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

When starting up or obtaining updated user or computer configuration Group Policy settings for QoS, the following process occurs.

1. The Group Policy engine retrieves the user or computer configuration Group Policy settings from Active Directory.
2. The Group Policy engine informs the QoS Client-Side Extension that there were changes in QoS policies.
3. The QoS Client-Side Extension sends a QoS policy event notification to the QoS Inspection Module.
4. The QoS Inspection Module retrieves the user or computer QoS policies and stores them.

When a new Transport Layer endpoint (TCP connection or UDP traffic) is created, the following process occurs.

1. The Transport Layer component of the TCP/IP stack informs the QoS Inspection Module.
2. The QoS Inspection Module compares the parameters of the Transport Layer endpoint to the stored QoS policies.
3. If a match is found, the QoS Inspection Module contacts Pacer.sys to create a flow, a data structure containing the DSCP value and the traffic throttling settings of the matching QoS policy. If there are multiple QoS policies that match the parameters of the Transport Layer endpoint, the most specific QoS policy is used.
4. Pacer.sys stores the flow and returns a flow number corresponding to the flow to the QoS Inspection Module.
5. The QoS Inspection Module returns the flow number to the Transport Layer.
6. The Transport Layer stores the flow number with the Transport Layer endpoint.

When a packet corresponding to a Transport Layer endpoint marked with a flow number is sent, the following process occurs.

1. The Transport Layer internally marks the packet with the flow number.
2. The Network Layer queries Pacer.sys for the DSCP value corresponding to the flow number of the packet.
3. Pacer.sys returns the DSCP value to the Network Layer.
4. The Network Layer changes the IPv4 TOS field or IPv6 Traffic Class field to the DSCP value specified by Pacer.sys and, for IPv4 packets, calculates the final IPv4 header checksum.
5. The Network Layer hands the packet to the Framing Layer.
6. Because the packet has been marked with a flow number, the Framing Layer hands the packet to Pacer.sys through NDIS 6.x.
7. Pacer.sys uses the flow number of the packet to determine if the packet needs to be throttled, and if so, schedules the packet for sending.
8. Pacer.sys hands the packet either immediately (if there is no traffic throttling) or as scheduled (if there is

traffic throttling) to NDIS 6.x for transmission over the appropriate network adapter.

These processes of Policy-based QoS provide the following advantages.

- The inspection of traffic to determine whether a QoS policy applies is done per-Transport Layer endpoint, rather than per-packet.
- There is no performance impact for traffic that does not match a QoS policy.
- Applications do not need to be modified to take advantage of DSCP-based differentiated service or traffic throttling.
- QoS policies can apply to traffic protected with IPsec.

For the next topic in this guide, see [QoS Policy Architecture](#).

For the first topic in this guide, see [Quality of Service \(QoS\) Policy](#).

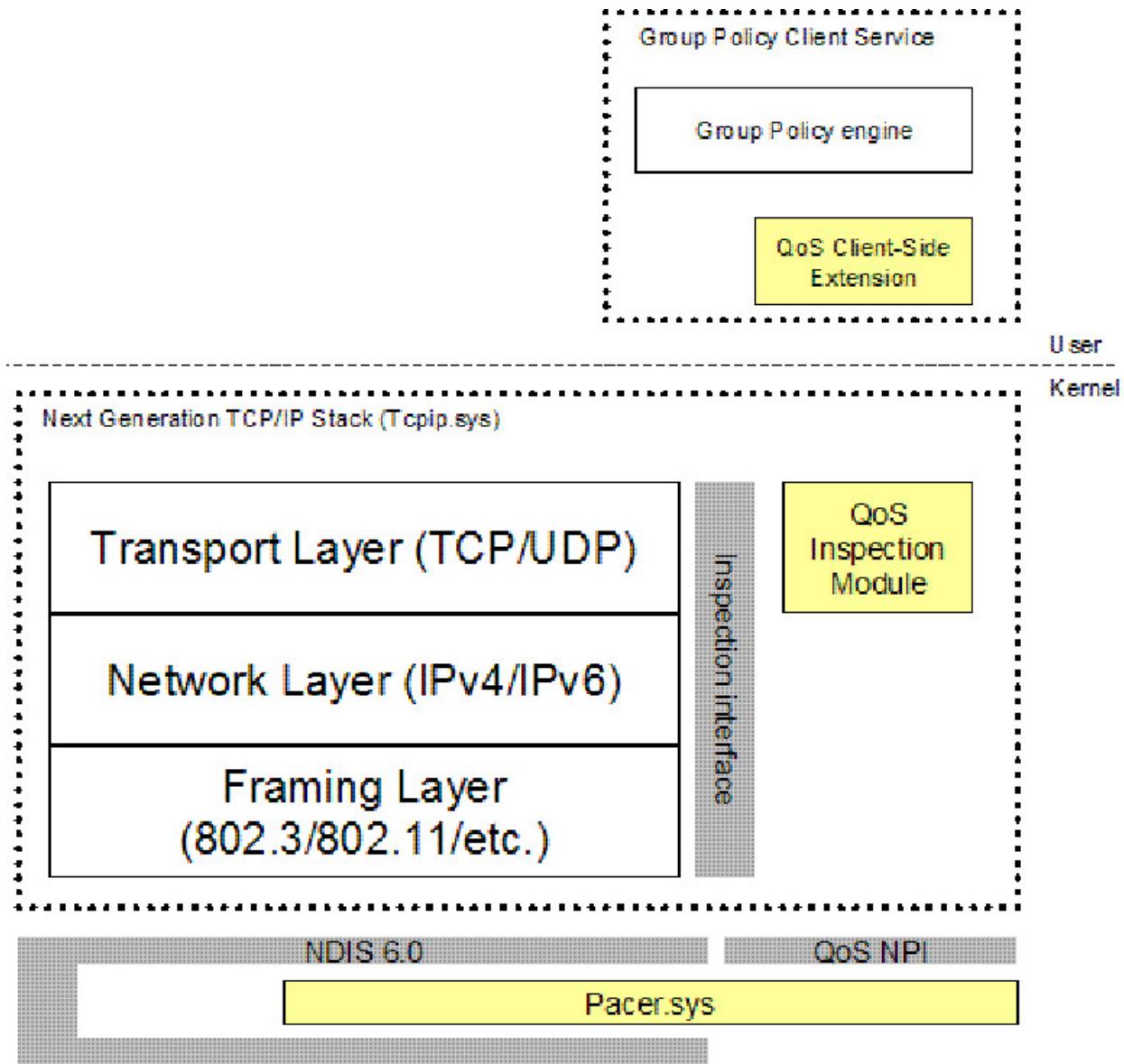
QoS Policy Architecture

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use this topic to learn about the architecture of QoS Policy.

The following figure shows the architecture of Policy-based QoS.



The architecture of Policy-based QoS consists of the following components:

- **Group Policy Client Service.** A Windows service that manages user and computer configuration Group Policy settings.
- **Group Policy engine.** A component of the Group Policy Client Service that retrieves user and computer configuration Group Policy settings from Active Directory upon startup and periodically checks for changes (by default, every 90 minutes). If changes are detected, the Group Policy engine retrieves the new Group Policy settings. The Group Policy engine processes the incoming GPOs and informs the QoS Client Side Extension when the QoS policies are updated.

- **QoS Client Side Extension.** A component of the Group Policy Client Service that waits for an indication from the Group Policy engine that the QoS policies have changed and informs the QoS Inspection Module.
- **TCP/IP Stack.** The TCP/IP stack that includes integrated support for IPv4 and IPv6 and supports Windows Filtering Platform.
- **QoS Inspection.** Module A component within the TCP/IP stack that waits for indications of QoS policy changes from the QoS Client Side Extension, retrieves the QoS policy settings, and interacts with the Transport Layer and Pacer.sys to internally mark traffic that matches the QoS policies.
- **NDIS 6.x.** A standard interface between kernel-mode network drivers and the operating system in Windows Server and Client operating systems. NDIS 6.x supports lightweight filters, which is a simplified driver model for NDIS intermediate drivers and miniport drivers that provides better performance.
- **QoS Network Provider Interface (NPI).** An interface for kernel-mode drivers to interact with Pacer.sys.
- **Pacer.sys.** An NDIS 6.x lightweight filter driver that controls packet scheduling for Policy-based QoS and for the traffic of applications that use the Generic QoS (GQoS) and Traffic Control (TC) APIs. Pacer.sys replaced Psched.sys in Windows Server 2003 and Windows XP. Pacer.sys is installed with the QoS Packet Scheduler component from the properties of a network connection or adapter.

For the next topic in this guide, see [QoS Policy Scenarios](#).

For the first topic in this guide, see [Quality of Service \(QoS\) Policy](#).

QoS Policy Scenarios

9/1/2018 • 8 minutes to read • [Edit Online](#)

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

You can use this topic to review hypothetical scenarios that demonstrate how, when, and why to use QoS Policy.

The two scenarios in this topic are:

1. Prioritize Network Traffic for a Line-of-Business Application
2. Prioritize Network Traffic for an HTTP Server Application

NOTE

Some sections of this topic contain general steps you can take to perform the described actions. For more detailed instructions on managing QoS Policy, see [Manage QoS Policy](#).

Scenario 1: Prioritize Network Traffic for a Line-of-Business Application

In this scenario, an IT department has several goals that they can accomplish by using QoS Policy:

- Provide better network performance for mission-critical applications.
- Provide better network performance for a key set of users while they are using a specific application.
- Ensure that the company-wide data Backup application doesn't impede network performance by using too much bandwidth at one time.

The IT department decides to configure QoS Policy to prioritize specific applications by using Differentiation Service Code Point (DSCP) values to classify network traffic, and to configure its routers to provide preferential treatment for higher priority traffic.

NOTE

For more information on DSCP, see the section **Define QoS Priority Through a Differentiated Services Code Point** in the topic [Quality of Service \(QoS\) Policy](#).

In addition to DSCP values, QoS policies can specify a throttle rate. Throttling has the effect of limiting all outbound traffic that matches the QoS Policy to a specific send rate.

QoS Policy Configuration

With three separate goals to accomplish, the IT administrator decides to create three different QoS policies.

QoS Policy for LOB App Servers

The first mission-critical application for which the IT department creates a QoS Policy is a company-wide Enterprise resource planning (ERP) application. The ERP application is hosted on several computers that are all running Windows Server 2016. In Active Directory Domain Services, these computers are members of an organization unit (OU) that was created for line-of-business (LOB) application servers. The client-side component for the ERP application is installed on computers that are running Windows 10 and Windows 8.1.

In Group Policy, an IT administrator selects the Group Policy Object (GPO) upon which the QoS policy will be applied. By using the QoS policy wizard, the IT administrator creates a QoS policy called "Server LOB policy" that

specifies a high-priority DSCP value of 44 for all applications, any IP address, TCP and UDP, and port number.

The QoS policy is applied only to the LOB servers by linking the GPO to the OU that contains only these servers, via the Group Policy Management Console (GPMC) tool. This initial server LOB policy applies the high-priority DSCP value whenever the computer sends network traffic. This QoS policy can later be edited (in the Group Policy Object Editor tool) to include the ERP application's port numbers, which limits the policy to apply only when the specified port number is used.

QoS Policy for the Finance Group

While many groups within the company access the ERP application, the finance group depends on this application when dealing with customers, and the group requires consistently high performance from the app.

To ensure that the finance group can support their customers, the QoS policy must classify these users' traffic as high priority. However, the policy should not apply when members of the finance group use applications other than the ERP application.

Because of this, the IT department defines a second QoS policy called "Client LOB policy" in the Group Policy Object Editor tool that applies a DSCP value of 60 when the finance user group runs the ERP application.

QoS Policy for a Backup App

A separate backup application is running on all computers. To ensure the backup application's traffic does not use all available network resources, the IT department creates a backup data policy. This backup policy specifies a DSCP value of 1 based on the executable name for the backup app, which is **backup.exe**.

A third GPO is created and deployed for all client computers in the domain. Whenever the backup application sends data, the low-priority DSCP value is applied, even if it originates from computers in the finance department.

NOTE

Network traffic without a QoS Policy sends with a DSCP value of 0.

Scenario Policies

The following table summarizes the QoS policies for this scenario.

POLICY NAME	DSCP VALUE	THROTTLE RATE	APPLIED TO ORGANIZATION UNITS	DESCRIPTION
[No policy]	0	None	[No deployment]	Best effort (default) treatment for unclassified traffic.
Backup data	1	None	All clients	Applies a low-priority DSCP value for this bulk data.
Server LOB	44	None	Computer OU for ERP servers	Applies high-priority DSCP for ERP server traffic
Client LOB	60	None	Finance user group	Applies high-priority DSCP for ERP client traffic

NOTE

DSCP values are represented in decimal form.

With QoS policies defined and applied by using Group Policy, outbound network traffic receives the policy-specified DSCP value. Routers then provide differential treatment based on these DSCP values by using queuing. For this IT department, the routers are configured with four queues: high-priority, middle-priority, best-effort, and low-priority.

When traffic arrives at the router with DSCP values from "Server LOB policy" and "Client LOB policy," the data is placed into high-priority queues. Traffic with a DSCP value of 0 receives a best-effort level of service. Packets with a DSCP value of 1 (from the backup application) receive low-priority treatment.

Prerequisites for prioritizing a line-of-business application

To complete this task, ensure that you meet the following requirements:

- The computers involved are running QoS-compatible operating systems.
- The computers involved are members of an Active Directory Domain Services (AD DS) domain so that they can be configured by using Group Policy.
- TCP/IP networks are set up with routers configured for DSCP (RFC 2474). For more information, see [RFC 2474](#).
- Administrative credentials requirements are met.

Administrative credentials

To complete this task, you must be able to create and deploy Group Policy Objects.

Setting up the test environment for prioritizing a line-of-business application

To set up the test environment, complete the following tasks.

- Create an AD DS domain with clients and users grouped into organization units. For instructions on deploying AD DS, see the [Core Network Guide](#).
- Configure the routers to differentially queue based on DSCP values. For example, DSCP value 44 enters a "Platinum" queue and all others are weighted-fair-queued.

NOTE

You can view DSCP values by using network captures with tools like Network Monitor. After you perform a network capture, you can observe the TOS field in captured data.

Steps for prioritizing a line-of-business application

To prioritize a line-of-business application, complete the following tasks:

1. Create and link a Group Policy Object (GPO) with a QoS policy.
2. Configure the routers to differentially treat a line-of-business application (by using queuing) based on the selected DSCP values. The procedures of this task will vary depending upon the type of routers you have.

Scenario 2: Prioritize Network Traffic for an HTTP Server Application

In Windows Server 2016, Policy-based QoS includes the feature URL-based Policies. URL Policies enable you to manage bandwidth for HTTP servers.

Many Enterprise applications are developed for and hosted on Internet Information Services (IIS) web servers, and the Web apps are accessed from browsers on client computers.

In this scenario, assume that you manage a set of IIS servers that host training videos for all your organization's employees. Your objective is to ensure that the traffic from these video servers won't overwhelm your network, and ensure that video traffic is differentiated from voice and data traffic on the network.

The task is similar to the task in Scenario 1. You will design and configure the traffic management settings, such as the DSCP value for the video traffic, and the throttling rate the same as you would for the line-of-business applications. But when specifying the traffic, instead of providing the application name, you only enter the URL to which your HTTP server application will respond: for example, <https://hrweb/training>.

NOTE

You cannot use URL-based QoS policies to prioritize network traffic for computers running Windows operating systems that were released prior to Windows 7 and Windows Server 2008 R2.

Precedence rules for URL-based policies

All the following URLs are valid and can be specified in QoS Policy and applied simultaneously to a computer or a user:

- <http://video>
- <https://training.hr.mycompany.com>
- <http://10.1.10.249:8080/tech>
- https://*/ebooks

But which one will receive precedence? The rules are simple. URL-based policies are prioritized in a left-to-right reading order. So, from the highest priority to the lowest priority, the URL fields are:

1. URL scheme

2. URL host

3. URL port

4. URL path

Details are as follows:

1. URL scheme

`https://` has a higher priority than `http://`.

2. URL host

From the highest priority to the lowest, they are:

1. Hostname
2. IPv6 address
3. IPv4 address
4. Wildcard

In the case of hostname, a hostname with more dotted elements (more depth) has a higher priority than a hostname with fewer dotted elements. For example, among the following hostnames:

- `video.internal.training.hr.mycompany.com` (depth = 6)
- `selfguide.training.mycompany.com` (depth = 4)
- `training` (depth = 1)
- `library` (depth = 1)

video.internal.training.hr.mycompany.com has the highest priority, and

selfguide.training.mycompany.com has the next highest priority. **Training** and **library** share the same lowest priority.

3. URL port

A specific or an implicit port number has a higher priority than a wildcard port.

4. URL path

Like a hostname, a URL path may consist of multiple elements. The one with more elements always has a higher priority than the one with less. For example, the following paths are listed by priority:

1. /ebooks/tech/windows/networking/qos
2. /ebooks/tech/windows/
3. /ebooks
4. /

If a user chooses to include all subdirectories and files following a URL path, this URL path will have a lower priority than it would have if the choice were not made.

A user may also choose to specify a destination IP address in a URL-based policy. The destination IP address has a lower priority than any of the four URL fields described previously.

Quintuple policy

A Quintuple policy is specified by protocol ID, source IP address, source port, destination IP address, and destination port. A Quintuple policy always has a higher precedence than any URL-based policy.

If a Quintuple policy is already applied for a user, a new URL-based policy will not cause conflicts on any of that user's client computers.

For the next topic in this guide, see [Manage QoS Policy](#).

For the first topic in this guide, see [Quality of Service \(QoS\) Policy](#).

Manage QoS Policy

9/1/2018 • 17 minutes to read • [Edit Online](#)

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

You can use this topic to learn about using the QoS Policy wizard to create, edit, or delete a QoS Policy.

NOTE

In addition to this topic, the following QoS Policy management documentation is available.

- [QoS Policy Events and Errors](#)

In Windows operating systems, QoS Policy combines the functionality of standards-based QoS with the manageability of Group Policy. Configuration of this combination makes for easy application of QoS policies to Group Policy Objects. Windows includes a QoS Policy Wizard to help you do the following tasks.

- [Create a QoS policy](#)
- [View, edit, or delete a QoS policy](#)

Create a QoS policy

Before you create a QoS policy, it is important that you understand the two key QoS controls that are used to manage network traffic:

- DSCP value
- Throttle rate

Prioritizing traffic with DSCP

As noted in the previous line-of-business application example, you can define the priority of outbound network traffic by using **Specify DSCP Value** to configure a QoS policy with a specific DSCP value.

As described in RFC 2474, DSCP allows values from 0 to 63 to be specified within the TOS field of an IPv4 packet and within the Traffic Class field in IPv6. Network routers use the DSCP value to classify network packets and to queue them appropriately.

NOTE

By default, Windows traffic has a DSCP value of 0.

The number of queues and their prioritization behavior needs to be designed as part of your organization's QoS strategy. For example, your organization may choose to have five queues: latency-sensitive traffic, control traffic, business-critical traffic, best-effort traffic, and bulk-data-transfer traffic.

Throttling traffic

Along with DSCP values, throttling is another key control for managing network bandwidth. As mentioned earlier, you can use the **Specify Throttle Rate** setting to configure a QoS policy with a specific throttle rate for outbound traffic. By using throttling, a QoS policy limits the outgoing network traffic to a specified throttle rate. Both DSCP marking and throttling can be used together to manage traffic effectively.

NOTE

By default, the **Specify Throttle Rate** check box is not selected.

To create a QoS policy, edit the settings of a Group Policy Object (GPO) from within the Group Policy Management Console (GPMC) tool. GPMC then opens the Group Policy Object Editor.

QoS policy names must be unique. How policies are applied to servers and end users depends on where the QoS policy is stored in the Group Policy Object Editor:

- A QoS policy in Computer Configuration\Windows Settings\QoS Policy applies to computers, regardless of the user that is currently logged on. You typically use computer-based QoS policies for server computers.
- A QoS policy in User Configuration\Windows Settings\QoS Policy applies to users after they have logged on, regardless of which computer they have logged on to.

To create a new QoS policy with the QoS Policy wizard

- In Group Policy Object Editor, right-click either of the **QoS Policy** nodes, and then click **Create a new policy**.

Wizard page 1 - Policy Profile

On the first page of the QoS Policy wizard, you can specify a policy name and configure how QoS controls outgoing network traffic.

To configure the Policy Profile page of the QoS-based Policy wizard

1. In **Policy name**, type a name for the QoS policy. The name must uniquely identify the policy.
2. Optionally, use **Specify DSCP Value** to enable DSCP marking, and then configure a DSCP value between 0 and 63.
3. Optionally, use **Specify Throttle Rate** to enable traffic throttling and configure the throttle rate. The throttle rate value must be greater than 1 and you can specify units of kilobytes per second (Kbps) or megabytes per second (MBps).
4. Click **Next**.

Wizard page 2 - Application Name

In the second page of the QoS Policy wizard you can apply the policy to all applications, to a specific application as identified by its executable name, to a path and application name, or to the HTTP server applications that handle requests for a specific URL.

- **All applications** specifies that the traffic management settings on the first page of the QoS Policy wizard apply to all applications.
- **Only applications with this executable name** specifies that the traffic management settings on the first page of the QoS Policy wizard are for a specific application. The executable file name must end with the .exe file name extension.
- **Only HTTP server applications responding to requests for this URL** specifies that the traffic management settings on the first page of the QoS Policy wizard apply to certain HTTP server applications only.

Optionally, you can enter the application path. To specify an application path, include the path with the application name. The path can include environment variables. For example, %ProgramFiles%\My Application Path\MyApp.exe, or c:\program files\my application path\myapp.exe.

NOTE

The application path cannot include a path that resolves to a symbolic link.

The URL must conform to [RFC 1738](#), in the form of `http[s]://<hostname>:<port>/<url-path>`. You can use a wildcard, `*`, for `<hostname>` and/or `<port>`, e.g. `http://training.*`, `https://*.*`, but the wildcard cannot denote a substring of `<hostname>` or `<port>`.

In other words, neither `http://my*site/` nor `https://*training*/` is valid.

Optionally, you can check **Include subdirectories and files** to perform matching on all subdirectories and files following a URL. For example, if this option is checked and the URL is `http://training`, QoS Policy will consider requests for `http://training/video` a good match.

To configure the Application Name page of the QoS Policy wizard

1. In **This QoS policy applies to**, select either **All applications** or **Only applications with this executable name**.
2. If you select **Only applications with this executable name**, specify an executable name ending with the .exe file name extension.
3. Click **Next**.

Wizard page 3 - IP Addresses

In the third page of the QoS Policy wizard you can specify IP address conditions for the QoS policy, including the following:

- All source IPv4 or IPv6 addresses or specific source IPv4 or IPv6 addresses
- All destination IPv4 or IPv6 addresses or specific destination IPv4 or IPv6 addresses

If you select **Only for the following source IP address** or **Only for the following destination IP address**, you must type one of the following:

- An IPv4 address, such as `192.168.1.1`
- An IPv4 address prefix using network prefix length notation, such as `192.168.1.0/24`
- An IPv6 address, such as `3ffe:ffff::1`
- An IPv6 address prefix, such as `3ffe:ffff::/48`

If you select both **Only for the following source IP address** and **Only for the following destination IP address**, both addresses or address prefixes must be either IPv4- or IPv6-based.

If you specified the URL for HTTP server applications in the previous wizard page, you'll notice that the source IP address for the QoS policy on this wizard page is grayed out.

This is true because the source IP address is the HTTP server address and it is not configurable here. On the other hand, you can still customize the policy by specifying the destination IP address. This makes it possible for you to create different policies for different clients by using the same HTTP server applications.

To configure the IP Addresses page of the QoS Policy wizard

1. In **This QoS policy applies to** (source), select **Any source IP address** or **Only for the following IP source address**.
2. If you selected **Only the following IP source address**, specify an IPv4 or IPv6 address or prefix.
3. In **This QoS policy applies to** (destination), select **Any destination address** or **Only for the following**

IP destination address.

4. If you selected **Only for the following IP destination address**, specify an IPv4 or IPv6 address or prefix that corresponds to the type of address or prefix specified for the source address.
5. Click **Next**.

Wizard page 4 - Protocols and Ports

On the fourth page of the QoS Policy wizard, you can specify the types of traffic and the ports that are controlled by the settings on the first page of the wizard. You can specify:

- TCP traffic, UDP traffic, or both
- All source ports, a range of source ports, or a specific source port
- All destination ports, a range of destination ports, or a specific destination port

To configure the Protocols and Ports page of the QoS Policy wizard

1. In **Select the protocol this QoS policy applies to**, select **TCP**, **UDP**, or **TCP and UDP**.
2. In **Specify the source port number**, select **From any source port** or **From this source port number**.
3. If you selected **From this source port number**, type a port number between 1 and 65535.

Optionally, you can specify a port range, in the format of "*Low:High*," where *Low* and *High* represent the lower bounds and upper bounds of the port range, inclusively. *Low* and *High* each must be a number between 1 and 65535. No space is allowed between the colon (:) character and the numbers.

4. In **Specify the destination port number**, select **To any destination port** or **To this destination port number**.
5. If you selected **To this destination port number** in the previous step, type a port number between 1 and 65535.

To complete the creation of the new QoS policy, click **Finish** on the **Protocols and Ports** page of the QoS Policy wizard. When completed, the new QoS policy is listed in the details pane of the Group Policy Object Editor.

To apply the QoS policy settings to users or computers, link the GPO in which the QoS policies are located to an Active Directory Domain Services container, such as a domain, a site, or an organizational unit (OU).

View, Edit, or Delete a QoS Policy

The pages of the QoS Policy wizard described previously correspond to the properties pages that are displayed when you view or edit the properties of a policy.

To view the properties of a QoS policy

- Right-click the policy name in the details pane of the Group Policy Object Editor, and then click **Properties**.

The Group Policy Object Editor displays the properties page with the following tabs:

- Policy Profile
- Application Name
- IP Addresses
- Protocols and Ports

To edit a QoS policy

- Right-click the policy name in the details pane of the Group Policy Object Editor, and then click **Edit existing policy**.

The Group Policy Object Editor displays the **Edit an existing QoS policy** dialog box.

To delete a QoS policy

- Right-click the policy name in the details pane of the Group Policy Object Editor, and then click **Delete policy**.

QoS Policy GPMC Reporting

After you have applied a number of QoS policies across your organization, it may be useful or necessary to periodically review how the policies are applied. A summary of the QoS policies for a specific user or computer can be viewed by using GPMC reporting.

To run the Group Policy Results Wizard for a report of QoS policies

- In GPMC, right-click the **Group Policy Results** node, and then select the menu option for **Group Policy Results Wizard**.

After Group Policy results are generated, click the **Settings** tab. On the **Settings** tab, the QoS policies can be found under the "Computer Configuration\Windows Settings\QoS Policy" and "User Configuration\Windows Settings\QoS Policy" nodes.

On the **Settings** tab, the QoS policies are listed by their QoS policy names with their DSCP value, throttle rate, policy conditions, and winning GPO listed in the same row..

The Group Policy results view uniquely identifies the winning GPO. When multiple GPOs have QoS policies with the same QoS policy name, the GPO with the highest GPO precedence is applied. This is the winning GPO. Conflicting QoS policies (identified by policy name) that are attached to a lower priority GPO are not applied. Note the GPO priorities define which QoS policies are deployed in the site, domain, or OU, as appropriate. After deployment, at a user or computer level, the **QoS Policy Precedence Rules** determine which traffic is allowed and blocked.

The QoS policy's DSCP value, throttle rate, and policy conditions are also visible in Group Policy Object Editor (GPOE)

Advanced settings for roaming and remote users

With QoS Policy, the goal is to manage traffic on an enterprise's network. In mobile scenarios, users might be sending traffic on or off the enterprise network. Because QoS policies are not relevant while away from the enterprise's network, QoS policies are enabled only on network interfaces that are connected to the enterprise for Windows 8, Windows 7, or Windows Vista.

For example, a user might connect her portable computer to her enterprise's network via virtual private network (VPN) from a coffee shop. For VPN, the physical network interface (such as wireless) will not have QoS policies applied. However, the VPN interface will have QoS policies applied because it connects to the enterprise. If the user later enters another enterprise's network that does not have an AD DS trust relationship, QoS policies will not be enabled.

Note that these mobile scenarios do not apply to server workloads. For example, a server with multiple network adapters might sit on the edge of an enterprise's network. The IT department might choose to have QoS policies throttle traffic that egresses the enterprise; however, this network adapter that sends this egress traffic does not necessarily connect back to the enterprise network. For this reason, QoS policies are always enabled on all network interfaces of a computer running Windows Server 2012.

NOTE

Selective enablement only applies to QoS policies and not to the Advanced QoS settings discussed next in this document.

Advanced QoS settings

Advanced QoS settings provide additional controls for IT administrators to manage computer network

consumption and DSCP markings. Advanced QoS settings apply only at the computer level, whereas QoS policies can be applied at both the computer and user levels.

To configure advanced QoS settings

1. Click **Computer Configuration**, and then click **Windows Settings in Group Policy**.
2. Right-click **QoS Policy**, and then click **Advanced QoS Settings**.

The following figure shows the two advanced QoS settings tabs: **Inbound TCP Traffic** and **DSCP Marking Override**.

NOTE

Advanced QoS Settings are computer-level Group Policy settings.

Advanced QoS settings: inbound TCP traffic

Inbound TCP Traffic controls the TCP bandwidth consumption on the receiver's side, whereas QoS policies affect the outbound TCP and UDP traffic.

By setting a lower throughput level on the **Inbound TCP Traffic** tab, TCP will limit the size of its advertised TCP receive window. The effect of this setting will be increased throughput rates and link utilization for TCP connections with higher bandwidths or latencies (bandwidth delay product). By default, computers running Windows Server 2012, Windows 8, Windows Server 2008 R2, Windows Server 2008, and Windows Vista are set to the maximum throughput level.

The TCP receive window has changed in Windows Server 2012, Windows 8, Windows Server 2008 R2, Windows Server 2008, and Windows Vista from previous versions of Windows. Previous versions of Windows limited the TCP receive-side window to a maximum of 64 kilobytes (KB), whereas Windows Server 2012, Windows 8, Windows Server 2008 R2, Windows Server 2008, and Windows Vista dynamically size the receive-side window up to 16 megabytes (MB). In the Inbound TCP Traffic control, you can control the inbound throughput level by setting the maximum value to which the TCP receive-window can grow. The levels correspond to the following maximum values.

INBOUND THROUGHPUT LEVEL	MAXIMUM
0	64 KB
1	256 KB
2	1 MB
3	16 MB

The actual window size may be a value equal to or smaller than the maximum, depending on network conditions.

To set the TCP receive-side window

1. In Group Policy Object Editor, click **Local Computer Policy**, click **Windows Settings**, right click **QoS Policy**, and then click **Advanced QoS Settings**.
2. In **TCP Receiving Throughput**, select **Configure TCP Receiving Throughput**, and then select the level of throughput that you want.
3. Link the GPO to the OU.

Advanced QoS settings: DSCP Marking Override

DSCP Marking Override restricts the ability of applications to specify—or "mark"—DSCP values other than those specified in QoS policies. By specifying that applications are allowed to set DSCP values, applications can set non-

zero DSCP values.

By specifying **Ignore**, applications that use QoS APIs will have their DSCP values set to zero, and only QoS policies can set DSCP values.

By default, computers running Windows Server 2016, Windows 10, Windows Server 2012 R2, Windows 8.1, Windows Server 2012, Windows 8, Windows Server 2008 R2, Windows Server 2008, and Windows Vista allow applications to specify DSCP values; applications and devices that do not use the QoS APIs are not overridden.

Wireless Multimedia and DSCP values

The [Wi-Fi Alliance](#) has established a certification for Wireless Multimedia (WMM) that defines four access categories (WMM_AC) for prioritizing network traffic transmitted on a Wi-Fi wireless network. The access categories include (in order of highest-to-lowest priority): voice, video, best effort, and background; respectively abbreviated as VO, VI, BE, and BK. The WMM specification defines which DSCP values correspond with each of the four access categories:

DSCP VALUE	WMM ACCESS CATEGORY
48-63	Voice (VO)
32-47	Video (VI)
24-31, 0-7	Best effort (BE)
8-23	Background (BK)

You can create QoS policies that use these DSCP values to ensure that portable computers with Wi-Fi Certified™ for WMM wireless adapters receive prioritized handling when associated with Wi-Fi Certified for WMM access points.

QoS Policy Precedence Rules

Similar to GPO's priorities, QoS policies have precedence rules to resolve conflicts when multiple QoS policies apply to a specific set of traffic. For outbound TCP or UDP traffic, only one QoS policy can be applied at a time, which means that QoS policies do not have a cumulative effect, such as where throttle rates would be summed.

In general, the QoS policy with the most matching conditions wins. When multiple QoS policies apply, the rules fall into three categories: user-level versus computer-level; application versus the network quintuple; and among the network quintuple.

By *network quintuple*, we mean the source IP address, destination IP address, source port, destination port, and protocol (TCP/UDP).

User-level QoS policy takes precedence over computer-level QoS policy

This rule greatly facilitates network administrators' management of QoS GPOs, particularly for user group-based policies. For example, if the network admin wants to define a QoS policy for a user group, they can just create and distribute a GPO to that group. They don't have to worry about which computers those users are logged on to and whether those computers will have conflicting QoS policies defined, because, if a conflict exists, the user-level policy always takes precedence.

NOTE

A user-level QoS policy is only applicable to traffic that is generated by that user. Other users of a specific computer, and the computer itself, will not be subject to any QoS policies that are defined for that user.

Application specificity and taking precedence over network quintuple

When multiple QoS policies match the specific traffic, the more specific policy is applied. Among policies that identify applications, a policy that includes the sending application's file path is considered more specific than another policy that only identifies the application name (no path). If multiple policies with applications still apply, the precedence rules use the network quintuple to find the best match.

Alternatively, multiple QoS policies might apply to the same traffic by specifying non-overlapping conditions. Between the conditions of applications and the network quintuple, the policy that specifies the application is considered more specific and is applied.

For example, policy_A only specifies an application name (app.exe), and policy_B specifies the destination IP address 192.168.1.0/24. When these QoS policies conflict (app.exe sends traffic to an IP address within the range of 192.168.4.0/24), policy_A gets applied.

More specificity takes precedence within the network quintuple

For policy conflicts within the network quintuple, the policy with the most matching conditions takes precedence. For example, assume policy_C specifies source IP address "any", destination IP address 10.0.0.1, source port "any", destination port "any", and protocol "TCP".

Next, assume policy_D specifies source IP address "any", destination IP address 10.0.0.1, source port "any", destination port 80, and protocol "TCP". Then policy_C and policy_D both match connections to destination 10.0.0.1:80. Because QoS Policy applies the policy with the most specific matching conditions, policy_D takes precedence in this example.

However, QoS policies might have an equal number of conditions. For example, several policies may each specify only one (but not the same) piece of the network quintuple. Among the network quintuple, the following order is from higher to lower precedence:

- Source IP address
- Destination IP address
- Source port
- Destination port
- Protocol (TCP or UDP)

Within a specific condition, such as IP address, a more specific IP address is treated with higher precedence; for example, an IP address 192.168.4.1 is more specific than 192.168.4.0/24.

Design your QoS policies as specifically as possible to simplify your organization's ability to understand which policies are in effect.

For the next topic in this guide, see [QoS Policy Events and Errors](#).

For the first topic in this guide, see [Quality of Service \(QoS\) Policy](#).

QoS Policy Error and Event Messages

9/1/2018 • 5 minutes to read • [Edit Online](#)

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

Following are the error and event messages that are associated with QoS Policy.

Informational Messages

Following is a list of QoS Policy informational messages.

MessageId	16500
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_MACHINE_POLICY_REFRESH_NO_CHANGE
Language	English
Message	Computer QoS policies successfully refreshed. No changes detected.
MessageId	16501
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_MACHINE_POLICY_REFRESH_WITH_CHANGE
Language	English
Message	Computer QoS policies successfully refreshed. Policy changes detected.
MessageId	16502
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_USER_POLICY_REFRESH_NO_CHANGE
Language	English
Message	User QoS policies successfully refreshed. No changes detected.

MessageId	16503
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_USER_POLICY_REFRESH_WITH_CHANGE
Language	English
Message	User QoS policies successfully refreshed. Policy changes detected.

MessageId	16504
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_TCP_AUTOTUNING_NOT_CONFIGURED
Language	English
Message	The Advanced QoS Setting for inbound TCP throughput level successfully refreshed. Setting value is not specified by any QoS policy. Local computer default will be applied.

MessageId	16505
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_TCP_AUTOTUNING_OFF
Language	English
Message	The Advanced QoS Setting for inbound TCP throughput level successfully refreshed. Setting value is Level 0 (minimum throughput).

MessageId	16506
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_TCP_AUTOTUNING_HIGHLY_RESTRICTED
Language	English
Message	The Advanced QoS Setting for inbound TCP throughput level successfully refreshed. Setting value is Level 1.

MessageId	16507
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_TCP_AUTOTUNING_RESTRICTED
Language	English

Message

The Advanced QoS Setting for inbound TCP throughput level successfully refreshed. Setting value is Level 2.

MessageId	16508
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_TCP_AUTOTUNING_NORMAL
Language	English

Message

The Advanced QoS Setting for inbound TCP throughput level successfully refreshed. Setting value is Level 3 (maximum throughput).

MessageId	16509
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_APP_MARKING_NOT_CONFIGURED
Language	English

Message

The Advanced QoS Setting for DSCP marking overrides successfully refreshed. Setting value is not specified. Applications can set DSCP values independently of QoS policies.

MessageId	16510
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_APP_MARKING_IGNORED
Language	English

Message

The Advanced QoS Setting for DSCP marking overrides successfully refreshed. Application DSCP marking requests will be ignored. Only QoS policies can set DSCP values.

MessageId	16511
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_APP_MARKING_ALLOWED
Language	English
Message	The Advanced QoS Setting for DSCP marking overrides successfully refreshed. Applications can set DSCP values independently of QoS policies.

MessageId	16512
Severity	Informational
SymbolicName	EVENT_EQOS_INFO_LOCAL_SETTING_DONT_USE_NLA
Language	English
Message	Selective application of QoS policies based on domain network category has been disabled. QoS policies will be applied to all network interfaces.

Warning Messages

Following is a list of QoS Policy warning messages.

MessageId	16600
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_TEST_1
Language	English
Message	EQOS: ***Testing***[with one string] "%2".

MessageId	16601
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_TEST_2
Language	English

Message	EQOS: ***Testing***[, with two strings, string1 is] "%2"[, string2 is] "%3".
MessageId	16602
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_MACHINE_POLICY_VERSION
Language	English
Message	The computer QoS policy "%2" has an invalid version number. This policy will not be applied.
MessageId	16603
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_USER_POLICY_VERSION
Language	English
Message	The user QoS policy "%2" has an invalid version number. This policy will not be applied.
MessageId	16604
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_MACHINE_POLICY_PROFILE_NOT_SPECIFIED
Language	English
Message	The computer QoS policy "%2" does not specify a DSCH value or throttle rate. This policy will not be applied.
MessageId	16605
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_USER_POLICY_PROFILE_NOT_SPECIFIED
Language	English

Message	The user QoS policy "%2" does not specify a DSCP value or throttle rate. This policy will not be applied.
MessageId	16606
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_MACHINE_POLICY_QUOTA_EXCEEDED
Language	English
Message	Exceeded the maximum number of computer QoS policies. The QoS policy "%2" and subsequent computer QoS policies will not be applied.
MessageId	16607
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_USER_POLICY_QUOTA_EXCEEDED
Language	English
Message	Exceeded the maximum number of user QoS policies. The QoS policy "%2" and subsequent user QoS policies will not be applied.
MessageId	16608
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_MACHINE_POLICY_CONFLICT
Language	English
Message	The computer QoS policy "%2" potentially conflicts with other QoS policies. See documentation for rules about which policy will be applied.
MessageId	16609
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_USER_POLICY_CONFLICT

Language	English
Message	The user QoS policy "%2" potentially conflicts with other QoS policies. See documentation for rules about which policy will be applied.
MessageId	16610
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_MACHINE_POLICY_NO_FULLPATH_APPNAME
Language	English
Message	The computer QoS policy "%2" was ignored because the application path cannot be processed. The application path may be invalid, contain an invalid drive letter, or contain a network mapped drive.
MessageId	16611
Severity	Warning
SymbolicName	EVENT_EQOS_WARNING_USER_POLICY_NO_FULLPATH_APPNAME
Language	English
Message	The user QoS policy "%2" was ignored because the application path cannot be processed. The application path may be invalid, contain an invalid drive letter, or contain a network mapped drive.

Error Messages

Following is a list of QoS Policy error messages.

MessageId	16700
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_MACHINE_POLICY_REFRESH
Language	English
Message	Computer QoS policies failed to refresh. Error code: "%2".

MessageId	16701
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_USER_POLICY_REFRESH
Language	English
Message	User QoS policies failed to refresh. Error code: "%2".

MessageId	16702
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_OPENING_MACHINE_POLICY_ROOT_KEY
Language	English
Message	QoS failed to open the machine-level root key for QoS policies. Error code: "%2".

MessageId	16703
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_OPENING_USER_POLICY_ROOT_KEY
Language	English
Message	QoS failed to open the user-level root key for QoS policies. Error code: "%2".

MessageId	16704
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_MACHINE_POLICY_KEYNAME_TOO_LONG
Language	English
Message	A computer QoS policy exceeds the maximum allowed name length. The offending policy is listed under the machine-level QoS policy root key, with index "%2".

MessageId	16705
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_USER_POLICY_KEYNAME_TOO_LONG
Language	English
Message	A user QoS policy exceeds the maximum allowed name length. The offending policy is listed under the user-level QoS policy root key, with index "%2".

MessageId	16706
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_MACHINE_POLICY_KEYNAME_SIZE_ZERO
Language	English
Message	A computer QoS policy has a zero length name. The offending policy is listed under the machine-level QoS policy root key, with index "%2".

MessageId	16707
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_USER_POLICY_KEYNAME_SIZE_ZERO
Language	English
Message	A user QoS policy has a zero length name. The offending policy is listed under the user-level QoS policy root key, with index "%2".

MessageId	16708
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_OPENING_MACHINE_POLICY_SUBKEY
Language	English
Message	QoS failed to open the registry subkey for a computer QoS policy. The policy is listed under the machine-level QoS policy root key, with index "%2".

MessageId	16709
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_OPENING_USER_POLICY_SUBKEY
Language	English
Message	QoS failed to open the registry subkey for a user QoS policy. The policy is listed under the user-level QoS policy root key, with index "%2".

MessageId	16710
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_PROCESSING_MACHINE_POLICY_FIELD
Language	English
Message	QoS failed to read or validate the "%2" field for the computer QoS policy "%3".

MessageId	16711
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_PROCESSING_USER_POLICY_FIELD
Language	English
Message	QoS failed to read or validate the "%2" field for the user QoS policy "%3".

MessageId	16712
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_SETTING_TCP_AUTOTUNING
Language	English

Message	QoS failed to read or set inbound TCP throughput level, error code: "%2".
MessageId	16713
Severity	Error
SymbolicName	EVENT_EQOS_ERROR_SETTING_APP_MARKING
Language	English
Message	QoS failed to read or set the DSCP marking override setting, error code: "%2".

For the next topic in this guide, see [QoS Policy Frequently Asked Questions](#).

For the first topic in this guide, see [Quality of Service \(QoS\) Policy](#).

QoS Policy Frequently Asked Questions

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

Following are frequently asked questions – and answers to those questions – for QoS Policy.

1. What operating system does my domain controller need to be running to use QoS Policy?

Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2, or Windows Server 2008

2. What operating systems support the application of QoS Policy to the user or computer?

You can apply QoS policies to users or computers running Windows Server 2016, Windows 10, Windows Server 2012 R2, Windows 8.1, Windows Server 2012, Windows 8, Windows Server 2008 R2, Windows Server 2008, and Windows Vista.

3. Do QoS policies apply to the sender or receiver of traffic?

QoS policies must be applied on the sending computer to affect its outbound traffic. In order to affect the bidirectional traffic of two computers, QoS policies need to be applied to both computers.

4. What happens if conflicting QoS policies are deployed to the same computer?

If multiple policies apply, the more specific QoS policy takes precedence. For example, a policy that states a host address (192.168.4.12) gets applied instead of a less specific network address (192.168.0.0/16). If a computer-level and user-level policy have the same specificity, the user-level QoS policy is applied instead of the computer-level QoS policy.

5. Is QoS Policy enabled by default?

No, QoS Policy is not enabled by default. You must create QoS policies manually to enable QoS. For more information, see [Manage QoS Policy](#).

For the first topic in this guide, see [Quality of Service \(QoS\) Policy](#).

SDN in Windows Server overview

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

Software Defined Networking (SDN) provides a method to centrally configure and manage physical and virtual network devices such as routers, switches, and gateways in your datacenter. You can use your existing SDN-compatible devices to achieve deeper integration between the virtual network and the physical network. Virtual network elements such as Hyper-V Virtual Switch, Hyper-V Network Virtualization, and RAS Gateway are designed to be integral elements of your SDN infrastructure.

NOTE

Hyper-V hosts and virtual machines (VMs) running SDN infrastructure servers, such as Network Controller and Software Load Balancing nodes, must have Windows Server 2016 Datacenter edition installed.

Hyper-V hosts containing only tenant workload VMs connected to SDN-controlled networks can use Windows Server 2016 Standard edition.

SDN is possible because network planes are no longer bound to the network devices themselves. However, other entities, such as datacenter management software like System Center 2016 use network planes. SDN allows you to manage your datacenter network dynamically, providing an automated, centralized way to meet the requirements of your applications and workloads.

You can use SDN to:

- Dynamically create, secure, and connect your network to meet the evolving needs of your apps
- Speed up the deployment of your workloads in a non-disruptive manner
- Contain security vulnerabilities from spreading across your network
- Define and control policies that govern both physical and virtual networks
- Implement network policies consistently at scale

SDN allows you to accomplish all of this while also reducing your overall infrastructure costs.

Contact the Datacenter and Cloud Networking product team

If you're interested in discussing SDN technologies with Microsoft or other SDN customers, there are a variety of methods for making contact.

For more information, see [Contact the Datacenter and Cloud Networking Team](#).

SDN Technologies

9/21/2018 • 4 minutes to read • [Edit Online](#)

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

The topics in this section provide overview and technical information about the Software Defined Networking technologies that are included in Windows Server 2016.

Network Controller

The network controller provides a centralized, programmable point of automation to manage, configure, monitor, and troubleshoot both virtual and physical network infrastructure in your datacenter. With the network controller, you can automate the configuration of network infrastructure instead of performing manual configuration of network devices and services.

The network controller is a highly available and scalable server and provides two application programming interfaces (APIs):

1. **Southbound API** – allows the network controller to communicate with the network.
2. **Northbound API** – allows you to communicate with the network controller.

You can use either Windows PowerShell, the Representational State Transfer (REST) API or a management application to manage the following physical and virtual network infrastructure.

- Hyper-V VMs and virtual switches
- Physical network switches
- Physical network routers
- Firewall software
- VPN Gateways, including Remote Access Service (RAS) Multitenant Gateways
- Load Balancers

Hyper-V Network Virtualization

Hyper-V Network Virtualization (HNV) helps you abstract your applications and workloads from the physical network by using virtual networks. Virtual networks provide the necessary multitenant isolation while running on a shared physical network fabric, thereby driving up resource utilization. To ensure that you can carry forward your existing investments, you can set up virtual networks on existing networking gear. Also, virtual networks are compatible with virtual Local Area Networks (VLANs).

Hyper-V Virtual Switch

The Hyper-V Virtual Switch is a software-based layer-2 Ethernet network switch that is available in Hyper-V Manager after you have installed the Hyper-V server role. The switch includes programmatically managed and extensible capabilities to connect virtual machines to both virtual networks and the physical network. Also, Hyper-V Virtual Switch provides policy enforcement for security, isolation, and service levels.

In Hyper-V Virtual Switch in Windows Server 2016, you can also deploy Switch Embedded Teaming (SET) and Remote Direct Memory Access (RDMA). For more information, see the section [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#) in this topic.

Internal DNS Service (iDNS) for SDN

Hosted virtual machines (VMs) and applications require DNS to communicate within their networks and with external resources on the Internet. With iDNS, you can provide tenants with DNS name resolution services for isolated, local namespace and Internet resources.

Network Function Virtualization

Hardware appliances, such as load balancers, firewalls, routers, and switches are increasingly becoming virtual appliances. Microsoft has virtualized networks, switches, gateways, NATs, load balancers, and firewalls. This "network function virtualization" is a natural progression of server virtualization and network virtualization. Virtual appliances are quickly emerging and creating a brand new market. They continue to generate interest and gain momentum in both virtualization platforms and cloud services.

The following Network Function Virtualization technologies are available.

- **Software Load Balancer (SLB) and Network Address Translation (NAT).** Enhance throughput by supporting Direct Server Return in which the return network traffic can bypass the Load Balancing multiplexer. For more details, see [Software Load Balancing /\(SLB/\) for SDN](#).
- **Datacenter Firewall.** Provide granular access control lists (ACLs), enabling you to apply firewall policies at the VM interface level or the subnet level. For more details, see [Datacenter Firewall Overview](#).
- **RAS Gateway for SDN.** Route network traffic between the physical network and VM network resources, regardless of the location. You can route the network traffic at the same physical location or many different locations. For more details, see [RAS Gateway for SDN](#).

Remote Direct Memory Access (RDMA) and Switch Embedded Teaming (SET)

In Windows Server 2016, you can enable RDMA on network adapters that are bound to a Hyper-V Virtual Switch with or without Switch Embedded Teaming (SET). This allows you to use fewer network adapters when you want to use RDMA and SET at the same time.

SET is an alternative NIC Teaming solution that you can use in environments that include Hyper-V and the Software Defined Networking (SDN) stack in Windows Server 2016. SET integrates some of the NIC Teaming functionality into the Hyper-V Virtual Switch.

SET allows you to group between one and eight physical Ethernet network adapters into one or more software-based virtual network adapters. These virtual network adapters provide fast performance and fault tolerance in the event of a network adapter failure.

SET member network adapters must all be installed in the same physical Hyper-V host to be placed in a team.

In addition, you can use Windows PowerShell commands to enable Data Center Bridging (DCB), create a Hyper-V Virtual Switch with an RDMA virtual NIC (vNIC), and create a Hyper-V Virtual Switch with SET and RDMA vNICs.

Border Gateway Protocol (BGP)

Border Gateway Protocol (BGP) is a dynamic routing protocol that automatically learns routes between sites that use site-to-site VPN connections. Therefore, BGP reduces manual configuration of routers. When you configure RAS Gateway, BGP lets you manage the routing of network traffic between your tenants' VM networks and remote sites.

Software Load Balancing (SLB) for SDN

Cloud Service Providers (CSPs) and Enterprises that deploy SDN can use Software Load Balancing (SLB) to evenly

distribute tenant and tenant customer network traffic among virtual network resources. The Windows Server SLB enables multiple servers to host the same workload, providing high availability and scalability.

Windows Server Containers

Windows Server Containers are a lightweight operating system virtualization method separating applications or services from other services running on the same container host. Each container has its own operating system, processes, file system, registry, and IP addresses, which you can connect to virtual networks.

System Center

Deploy and manage the SDN infrastructure with [Virtual Machine Management \(VMM\)](#) and [Operations Manager](#). With VMM, you provision and manage the resources needed to create and deploy virtual machines and services to private clouds. With Operations Manager, you monitor services, devices, and operations across your enterprise to identify problems for immediate action.

Hyper-V Network Virtualization

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

Introduced in Windows Server 2012, Hyper-V Network Virtualization (HNV) enables virtualization of customer networks on top of a shared physical network infrastructure. With minimal changes necessary on the physical network fabric, HNV gives service providers the agility to deploy and migrate tenant workloads anywhere across the three clouds: the service provider cloud, the private cloud, or the Microsoft Azure public cloud.

For more information, see the following topics:

- [Hyper-V Network Virtualization Overview in Windows Server 2016](#)
- [What's New in Hyper-V Network Virtualization in Windows Server 2016](#)



Did you know that Microsoft Azure provides similar functionality in the cloud? Learn more about [Microsoft Azure virtualization solutions](#).

Create a hybrid virtualization solution in Microsoft Azure:

- [Connect an On-premises Network to Azure via Site to Site VPN and Extend Active Directory onto an IaaS VM DC in Azure](#)

Hyper-V Network Virtualization Overview in Windows Server 2016

9/1/2018 • 11 minutes to read • [Edit Online](#)

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

In Windows Server 2016 and Virtual Machine Manager, Microsoft provides an end-to-end network virtualization solution. There are five major components that comprise Microsoft's network virtualization solution:

- **Windows Azure Pack for Windows Server** provides a tenant facing portal to create virtual networks, and an administrative portal to manage virtual networks.
- **Virtual Machine Manager** (VMM) provides centralized management of the network fabric.
- **Microsoft Network Controller** provides a centralized, programmable point of automation to manage, configure, monitor, and troubleshoot virtual and physical network infrastructure in your datacenter.
- **Hyper-V Network Virtualization** provides the infrastructure needed to virtualize network traffic.
- **Hyper-V Network Virtualization gateways** provide connections between virtual and physical networks.

This topic introduces concepts and explains the key benefits and capabilities of Hyper-V Network Virtualization (one part of the overall network virtualization solution) in Windows Server 2016. It explains how network virtualization benefits both private clouds looking for enterprise workload consolidation and public cloud service providers of Infrastructure as a Service (IaaS).

For more technical details about networking virtualization in Windows Server 2016, see [Hyper-V Network Virtualization Technical Details in Windows Server 2016](#).

Did you mean

- [Hyper-V Network Virtualization Overview \(Windows Server 2012 R2 \)](#)
- [Hyper-V Overview](#)
- [Hyper-V Virtual Switch Overview](#)

Feature description

Hyper-V Network Virtualization provides "virtual networks" (called a VM network) to virtual machines similar to how server virtualization (hypervisor) provides "virtual machines" to the operating system. Network virtualization decouples virtual networks from the physical network infrastructure and removes the constraints of VLAN and hierarchical IP address assignment from virtual machine provisioning. This flexibility makes it easy for customers to move to IaaS clouds and efficient for hosters and datacenter administrators to manage their infrastructure, while maintaining the necessary multi-tenant isolation, security requirements, and supporting overlapping Virtual Machine IP addresses.

Customers want to seamlessly extend their datacenters to the cloud. Today there are technical challenges in making such seamless hybrid cloud architectures. One of the biggest hurdles customers face is reusing their existing network topologies (subnets, IP addresses, network services, and so on.) in the cloud and bridging between their on-premises resources and their cloud resources. Hyper-V Network Virtualization provides the concept of a VM Network that is independent of the underlying physical network. With this concept of a VM Network, composed of one or more Virtual Subnets, the exact location in the physical network of virtual machines

attached to a virtual network is decoupled from the virtual network topology. As a result, customers can easily move their virtual subnets to the cloud while preserving their existing IP addresses and topology in the cloud so that existing services continue to work unaware of the physical location of the subnets. That is, Hyper-V Network Virtualization enables a seamless hybrid cloud.

In addition to hybrid cloud, many organizations are consolidating their datacenters and creating private clouds to internally get the efficiency and scalability benefit of cloud architectures. Hyper-V Network Virtualization allows better flexibility and efficiency for private clouds by decoupling a business unit's network topology (by making it virtual) from the actual physical network topology. In this way, the business units can easily share an internal private cloud while being isolated from each other and continue to keep existing network topologies. The datacenter operations team has flexibility to deploy and dynamically move workloads anywhere in the datacenter without server interruptions providing better operational efficiencies and an overall more effective datacenter.

For workload owners, the key benefit is that they can now move their workload "topologies" to the cloud without changing their IP addresses or re-writing their applications. For example, the typical three-tier LOB application is composed of a front end tier, a business logic tier, and a database tier. Through policy, Hyper-V Network Virtualization allows customer onboarding all or parts of the three tiers to the cloud, while keeping the routing topology and the IP addresses of the services (i.e. virtual machine IP addresses), without requiring the applications to be changed.

For infrastructure owners, the additional flexibility in virtual machine placement makes it possible to move workloads anywhere in the datacenters without changing the virtual machines or reconfiguring the networks. For example Hyper-V Network Virtualization enables cross subnet live migration so that a virtual machine can live migrate anywhere in the datacenter without a service disruption. Previously live migration was limited to the same subnet restricting where virtual machines could be located. Cross subnet live migration allows administrators to consolidate workloads based on dynamic resource requirements, energy efficiency, and can also accommodate infrastructure maintenance without disrupting customer workload up time.

Practical applications

With the success of virtualized datacenters, IT organizations and hosting providers (providers who offer colocation or physical server rentals) have begun offering more flexible virtualized infrastructures that make it easier to offer on-demand server instances to their customers. This new class of service is referred to as Infrastructure as a Service (IaaS). Windows Server 2016 provides all the required platform capabilities to enable enterprise customers to build private clouds and transition to an IT as a service operational model. Windows Server 2016 also enables hosters to build public clouds and offer IaaS solutions to their customers. When combined with Virtual Machine Manager and Windows Azure Pack to manage Hyper-V Network Virtualization policy, Microsoft provides a powerful cloud solution.

Windows Server 2016 Hyper-V Network Virtualization provides policy-based, software-controlled network virtualization that reduces the management overhead faced by enterprises when they expand dedicated IaaS clouds, and it provides cloud hosters better flexibility and scalability for managing virtual machines to achieve higher resource utilization.

An IaaS scenario that has virtual machines from different organizational divisions (dedicated cloud) or different customers (hosted cloud) requires secure isolation. Today's solution, virtual local area networks (VLANs), can present significant disadvantages in this scenario.

VLANs

Currently, VLANs are the mechanism that most organizations use to support address space reuse and tenant isolation. A VLAN uses explicit tagging (VLAN ID) in the Ethernet frame headers, and it relies on Ethernet switches to enforce isolation and restrict traffic to network nodes with the same VLAN ID. The main disadvantages with VLANs are as follows:

- Increased risk of an inadvertent outage due to cumbersome reconfiguration of production switches

whenever virtual machines or isolation boundaries move in the dynamic datacenter.

- Limited in scalability because there is a maximum of 4094 VLANs and typical switches support no more than 1000 VLAN IDs.
- Constrained within a single IP subnet, which limits the number of nodes within a single VLAN and restricts the placement of virtual machines based on physical locations. Even though VLANs can be expanded across sites, the entire VLAN must be on the same subnet.

IP address assignment

In addition to the disadvantages that are presented by VLANs, virtual machine IP address assignment presents issues, which include:

- Physical locations in datacenter network infrastructure determine virtual machine IP addresses. As a result, moving to the cloud typically requires changing IP addresses of the service workloads.
- Policies are tied to IP addresses, such as firewall rules, resource discovery and directory services, and so on. Changing IP addresses requires updating all the associated policies.
- Virtual machine deployment and traffic isolation are dependent on the topology.

When datacenter network administrators plan the physical layout of the datacenter, they must make decisions about where subnets will be physically placed and routed. These decisions are based on IP and Ethernet technology that influence the potential IP addresses that are allowed for virtual machines running on a given server or a blade that is connected to a particular rack in the datacenter. When a virtual machine is provisioned and placed in the datacenter, it must adhere to these choices and restrictions regarding the IP address. Therefore, the typical result is that the datacenter administrators assign new IP addresses to the virtual machines.

The problem with this requirement is that in addition to being an address, there is semantic information associated with an IP address. For instance, one subnet may contain given services or be in a distinct physical location. Firewall rules, access control policies, and IPsec security associations are commonly associated with IP addresses. Changing IP addresses forces the virtual machine owners to adjust all their policies that were based on the original IP address. This renumbering overhead is so high that many enterprises choose to deploy only new services to the cloud, leaving legacy applications alone.

Hyper-V Network Virtualization decouples virtual networks for customer virtual machines from the physical network infrastructure. As a result, it enables customer virtual machines to maintain their original IP addresses, while allowing datacenter administrators to provision customer virtual machines anywhere in the datacenter without reconfiguring physical IP addresses or VLAN IDs. The next section summarizes the key functionality.

Important functionality

The following is a list of the key functionality, benefits, and capabilities of Hyper-V Network Virtualization in Windows Server 2016:

- **Enables flexible workload placement - Network isolation and IP address re-use without VLANs**

Hyper-V Network Virtualization decouples the customer's virtual networks from the physical network infrastructure of the hosters, providing freedom for workload placements inside the datacenters. Virtual machine workload placement is no longer limited by the IP address assignment or VLAN isolation requirements of the physical network because it is enforced within Hyper-V hosts based on software-defined, multitenant virtualization policies.

Virtual machines from different customers with overlapping IP addresses can now be deployed on the same host server without requiring cumbersome VLAN configuration or violating the IP address hierarchy. This can streamline the migration of customer workloads into shared IaaS hosting providers, allowing customers to move those workloads without modification, which includes leaving the virtual machine IP addresses

unchanged. For the hosting provider, supporting numerous customers who want to extend their existing network address space to the shared IaaS datacenter is a complex exercise of configuring and maintaining isolated VLANs for each customer to ensure the coexistence of potentially overlapping address spaces. With Hyper-V Network Virtualization, supporting overlapping addresses is made easier and requires less network reconfiguration by the hosting provider.

In addition, physical infrastructure maintenance and upgrades can be done without causing a down time of customer workloads. With Hyper-V Network Virtualization, virtual machines on a specific host, rack, subnet, VLAN, or entire cluster can be migrated without requiring a physical IP address change or major reconfiguration.

- **Enables easier moves for workloads to a shared IaaS cloud**

With Hyper-V Network Virtualization, IP addresses and virtual machine configurations remain unchanged. This enables IT organizations to more easily move workloads from their datacenters to a shared IaaS hosting provider with minimal reconfiguration of the workload or their infrastructure tools and policies. In cases where there is connectivity between two datacenters, IT administrators can continue to use their tools without reconfiguring them.

- **Enables live migration across subnets**

Live migration of virtual machine workloads traditionally has been limited to the same IP subnet or VLAN because crossing subnets required the virtual machine's guest operating system to change its IP address. This address change breaks existing communication and disrupts the services running on the virtual machine. With Hyper-V Network Virtualization, workloads can be live migrated from servers running Windows Server 2016 in one subnet to servers running Windows Server 2016 in a different subnet without changing the workload IP addresses. Hyper-V Network Virtualization ensures that virtual machine location changes due to live migration are updated and synchronized among hosts that have ongoing communication with the migrated virtual machine.

- **Enables easier management of decoupled server and network administration**

Server workload placement is simplified because migration and placement of workloads are independent of the underlying physical network configurations. Server administrators can focus on managing services and servers, and network administrators can focus on overall network infrastructure and traffic management. This enables datacenter server administrators to deploy and migrate virtual machines without changing the IP addresses of the virtual machines. There is reduced overhead because Hyper-V Network Virtualization allows virtual machine placement to occur independently of network topology, reducing the need for network administrators to be involved with placements that might change the isolation boundaries.

- **Simplifies the network and improves server/network resource utilization**

The rigidity of VLANs and the dependency of virtual machine placement on a physical network infrastructure results in overprovisioning and underutilization. By breaking the dependency, the increased flexibility of virtual machine workload placement can simplify the network management and improve server and network resource utilization. Note that Hyper-V Network Virtualization supports VLANs in the context of the physical datacenter. For example, a datacenter may want all Hyper-V Network Virtualization traffic to be on a specific VLAN.

- **Is compatible with existing infrastructure and emerging technology**

Hyper-V Network Virtualization can be deployed in today's datacenter, yet it is compatible with emerging datacenter "flat network" technologies.

For example, HNV in Windows Server 2016 supports the VXLAN encapsulation format and the Open vSwitch Database Management Protocol (OVSDB) as the SouthBound Interface (SBI)..

- **Provides for interoperability and ecosystem readiness**

Hyper-V Network Virtualization supports multiple configurations for communication with existing resources, such as cross premise connectivity, storage area network (SAN), non-virtualized resource access, and so on. Microsoft is committed to working with ecosystem partners to support and enhance the experience of Hyper-V Network Virtualization in terms of performance, scalability, and manageability.

- **Policy-based configuration**

Network virtualization policies in Windows Server 2016 are configured through the Microsoft Network Controller. The network controller has a RESTful northbound API, and Windows PowerShell interface to configure policy. For more information about the Microsoft Network Controller, see [Network Controller](#).

Software requirements

Hyper-V Network Virtualization using the Microsoft Network Controller requires Windows Server 2016 and the Hyper-V role.

See also

To learn more about Hyper-V Network Virtualization in Windows Server 2016 see the following links:

CONTENT TYPE	REFERENCES
Community Resources	- Private Cloud Architecture Blog - Ask questions: cloudnetfb@microsoft.com
RFC	- VXLAN - RFC 7348
Related Technologies	- Network Controller - Hyper-V Network Virtualization Overview (Windows Server 2012 R2)

Hyper-V Network Virtualization Technical Details in Windows Server 2016

9/1/2018 • 23 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016

Server virtualization enables multiple server instances to run concurrently on a single physical host; yet server instances are isolated from each other. Each virtual machine essentially operates as if it is the only server running on the physical computer.

Network virtualization provides a similar capability, in which multiple virtual networks (potentially with overlapping IP addresses) run on the same physical network infrastructure and each virtual network operates as if it is the only virtual network running on the shared network infrastructure. Figure 1 shows this relationship.

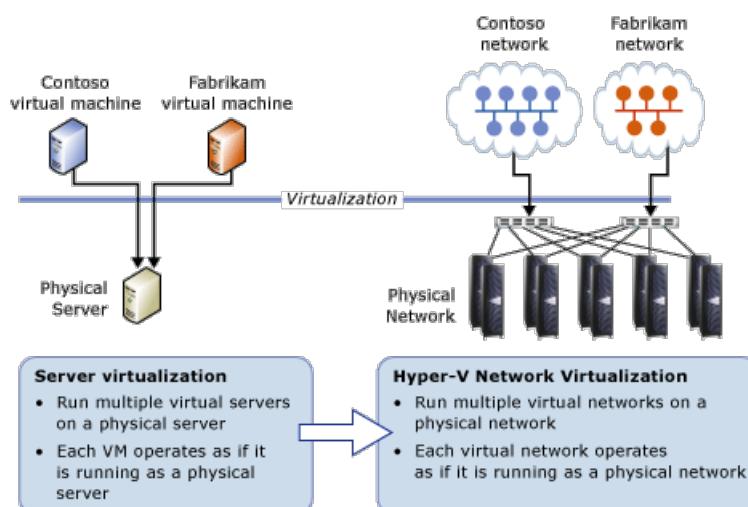


Figure 1: Server virtualization versus network virtualization

Hyper-V Network Virtualization Concepts

In Hyper-V Network Virtualization (HNV), a customer or tenant is defined as the "owner" of a set of IP subnets that are deployed in an enterprise or datacenter. A customer can be a corporation or enterprise with multiple departments or business units in a private datacenter which require network isolation, or a tenant in a public data center which is hosted by a service provider. Each customer can have one or more [Virtual networks](#) in the datacenter, and each virtual network consists of one or more [Virtual subnets](#).

There are two HNV implementations which will be available in Windows Server 2016: HNVv1 and HNVv2.

- **HNVv1**

HNVv1 is compatible with Windows Server 2012 R2 and System Center 2012 R2 Virtual Machine Manager (VMM). Configuration for HNVv1 relies on WMI management and Windows PowerShell cmdlets (facilitated through System Center VMM) to define isolation settings and Customer Address (CA) - virtual network - to Physical Address (PA) mappings and routing. No additional features have been added to HNVv1 in Windows Server 2016 and no new features are planned.

- SET Teaming and HNV V1 are not compatible by platform.

- o To use HA NVGRE gateways users need to either use LBFO team or No team. Or

- o Use Network Controller Deployed gateways with SET teamed switch.

- **HNVv2**

A significant number of new features are included in HNVv2 which is implemented using the Azure Virtual Filtering Platform (VFP) forwarding extension in the Hyper-V Switch. HNVv2 is fully integrated with Microsoft Azure Stack which includes the new Network Controller in the Software Defined Networking (SDN) Stack. Virtual network policy is defined through the Microsoft [Network Controller](#) using a RESTful NorthBound (NB) API and plumbed to a Host Agent via multiple SouthBound Interfaces (SBI) including OVSDB. The Host Agent programs policy in the VFP extension of the Hyper-V Switch where it is enforced.

IMPORTANT

This topic focuses on HNVv2.

Virtual network

- Each virtual network consists of one or more virtual subnets. A virtual network forms an isolation boundary where the virtual machines within a virtual network can only communicate with each other. Traditionally, this isolation was enforced using VLANs with a segregated IP address range and 802.1q Tag or VLAN ID. But with HNV, isolation is enforced using either NVGRE or VXLAN encapsulation to create overlay networks with the possibility of overlapping IP subnets between customers or tenants.
- Each virtual network has a unique Routing Domain ID (RDID) on the host. This RDID roughly maps to a Resource ID to identify the virtual network REST resource in the Network Controller. The virtual network REST resource is referenced using a Uniform Resource Identifier (URI) namespace with the appended Resource ID.

Virtual subnets

- A virtual subnet implements the Layer 3 IP subnet semantics for the virtual machines in the same virtual subnet. The virtual subnet forms a broadcast domain (similar to a VLAN) and isolation is enforced by using either the NVGRE Tenant Network ID (TNI) or VXLAN Network Identifier (VNI) field.
- Each virtual subnet belongs to a single virtual network (RDID), and it is assigned a unique Virtual Subnet ID (VSID) using either the TNI or VNI key in the encapsulated packet header. The VSID must be unique within the datacenter and is in the range 4096 to $2^{24}-2$.

A key advantage of the virtual network and routing domain is that it allows customers to bring their own network topologies (for example, IP subnets) to the cloud. Figure 2 shows an example where the Contoso Corp has two separate networks, the R&D Net and the Sales Net. Because these networks have different routing domain IDs, they cannot interact with each other. That is, Contoso R&D Net is isolated from Contoso Sales Net even though both are owned by Contoso Corp. Contoso R&D Net contains three virtual subnets. Note that both the RDID and VSID are unique within a datacenter.

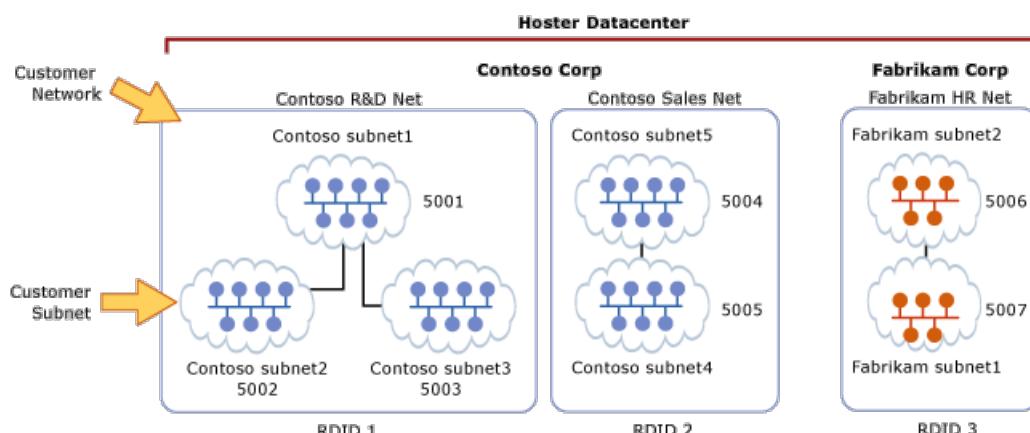


Figure 2: Customer networks and virtual subnets

Layer 2 Forwarding

In Figure 2, the virtual machines in VSID 5001 can have their packets forwarded to virtual machines that are also in VSID 5001 through the Hyper-V Switch. The incoming packets from a virtual machine in VSID 5001 are sent to a specific VPort on the Hyper-V Switch. Ingress rules (e.g. encap) and mappings (e.g. encapsulation header) are applied by the Hyper-V Switch for these packets. The packets are then forwarded either to a different VPort on the Hyper-V Switch (if the destination virtual machine is attached to the same host) or to a different Hyper-V switch on a different host (if the destination virtual machine is located on a different host).

Layer 3 Routing

Similarly, the virtual machines in VSID 5001 can have their packets routed to virtual machines in VSID 5002 or VSID 5003 by the HNV distributed router which is present in each Hyper-V host's VSwitch. Upon delivering the packet to the Hyper-V switch, HNV updates the VSID of the incoming packet to the VSID of the destination virtual machine. This will only happen if both VSIDs are in the same RDID. Therefore, virtual network adapters with RDID1 cannot send packets to virtual network adapters with RDID2 without traversing a gateway.

NOTE

In the packet flow description above, the term "virtual machine" actually means the virtual network adapter on the virtual machine. The common case is that a virtual machine only has a single virtual network adapter. In this case, the words "virtual machine" and "virtual network adapter" can conceptually mean the same thing.

Each virtual subnet defines a Layer 3 IP subnet and a Layer 2 (L2) broadcast domain boundary similar to a VLAN. When a virtual machine broadcasts a packet, HNV uses Unicast Replication (UR) to make a copy of the original packet and replace the destination IP and MAC with the addresses of each VM which are in the same VSID.

NOTE

When Windows Server 2016 ships, broadcast and subnet multicasts will be implemented using unicast replication. Cross-subnet multicast routing and IGMP are not supported.

In addition to being a broadcast domain, the VSID provides isolation. A virtual network adapter in HNV is connected to a Hyper-V switch port that will have ACL rules applied either directly to the port (virtualNetworkInterface REST resource) or to the virtual subnet (VSID) of which it is a part.

The Hyper-V switch port must have an ACL rule applied. This ACL could be ALLOW ALL, DENY ALL, or be more specific to only allow certain types of traffic based on 5-tuple (Source IP, Destination IP, Source Port, Destination Port, Protocol) matching.

NOTE

Hyper-V Switch Extensions will not work with HNVv2 in the new Software Defined Networking (SDN) stack. HNVv2 is implemented using the Azure Virtual Filtering Platform (VFP) switch extension which cannot be used in conjunction with any other 3rd-party switch extension.

Switching and Routing in Hyper-V Network Virtualization

HNVv2 implements correct Layer 2 (L2) switching and Layer 3 (L3) routing semantics to work just as a physical switch or router would work. When a virtual machine connected to an HNV virtual network attempts to establish a connection with another virtual machine in the same virtual subnet (VSID) it will first need to learn the CA MAC address of the remote virtual machine. If there is an ARP entry for the destination virtual machine's IP address in

the source virtual machine's ARP table, the MAC address from this entry is used. If an entry does not exist, the source virtual machine will send an ARP broadcast with a request for the MAC address corresponding to the destination virtual machine's IP address to be returned. The Hyper-V Switch will intercept this request and send it to the Host Agent. The Host Agent will look in its local database for a corresponding MAC address for the requested destination virtual machine's IP address.

NOTE

The Host Agent, acting as the OVSDB server, uses a variant of the VTEP schema to store CA-PA mappings, MAC table, and so on.

If a MAC address is available, the Host Agent injects an ARP response and sends this back to the virtual machine. After the virtual machine's networking stack has all the required L2 header information, the frame is sent to the corresponding Hyper-V Port on the V-Switch. Internally, the Hyper-V Switch tests this frame against N-tuple matching rules assigned to the V-Port and applies certain transformations to the frame based on these rules. Most importantly, a set of encapsulation transformations is applied to construct the encapsulation header using either NVGRE or VXLAN, depending on the policy defined at the Network Controller. Based on the policy programmed by the Host Agent, a CA-PA mapping is used to determine the IP address of the Hyper-V host where the destination virtual machine resides. The Hyper-V Switch ensures the correct routing rules and VLAN tags are applied to the outer packet so it reaches the remote PA address.

If a virtual machine connected to an HNV virtual network wants to create a connection with a virtual machine in a different virtual subnet (VSID), the packet needs to be routed accordingly. HNV assumes a star-topology where there is only one IP address in the CA space used as the next-hop to reach all IP prefixes (meaning one default route/gateway). Currently, this enforces a limitation to a single default route and non-default routes are not supported.

Routing Between Virtual Subnets

In a physical network, an IP subnet is a Layer 2 (L2) domain where computers (virtual and physical) can directly communicate with each other. The L2 domain is a broadcast domain where ARP entries (IP:MAC address map) are learned through ARP requests that are broadcast on all interfaces and ARP responses are sent back to the requesting host. The computer uses the MAC information learned from the ARP response to completely construct the L2 frame including Ethernet headers. However, if an IP address is in a different L3 subnet, the ARP request does not cross this L3 boundary. Instead, an L3 router interface (next-hop or default gateway) with an IP address in the source subnet must respond to these ARP requests with its own MAC address.

In standard Windows networking, an administrator can create static routes and assign these to a network interface. Additionally, a "default gateway" is usually configured to be the next-hop IP address on an interface where packets destined for the default route (0.0.0.0/0) are sent. Packets are sent to this default gateway if no specific routes exist. This is typically the router for your physical network. HNV uses a built-in router that is part of every host and has an interface in every VSID to create a distributed router for the virtual network(s).

Since HNV assumes a star topology, the HNV distributed router acts as a single default gateway for all traffic that is going between Virtual Subnets that are part of the same VSID network. The address used as the default gateway defaults to the lowest IP address in the VSID and is assigned to the HNV distributed router. This distributed router allows for a very efficient way for all traffic inside a VSID Network to be routed appropriately because each host can directly route the traffic to the appropriate host without needing an intermediary. This is particularly true when two virtual machines in the same VM Network but different Virtual Subnets are on the same physical host. As you will see later in this section, the packet never has to leave the physical host.

Routing between PA subnets

In contrast to HNVv1 which allocated one PA IP address for each Virtual Subnet (VSID), HNVv2 now uses one PA IP address per Switch-Embedded Teaming (SET) NIC team member. The default deployment assumes a two-NIC team and assigns two PA IP addresses per host. A single host has PA IPs assigned from the same Provider (PA)

logical subnet on the same VLAN. Two tenant VMs in the same virtual subnet may indeed be located on two different hosts which are connected to two different provider logical subnets. HNV will construct the outer IP headers for the encapsulated packet based on the CA-PA mapping. However, it relies on the host TCP/IP stack to ARP for the default PA gateway and then builds the outer Ethernet headers based on the ARP response. Typically, this ARP response comes from the SVI interface on the physical switch or L3 router where the host is connected. HNV therefore relies on the L3 router for routing the encapsulated packets between provider logical subnets / VLANs.

Routing Outside a Virtual Network

Most customer deployments will require communication from the HNV environment to resources that are not part of the HNV environment. Network Virtualization gateways are required to allow communication between the two environments. Infrastructures requiring an HNV Gateway include Private Cloud and Hybrid Cloud. Basically, HNV gateways are required for Layer 3 routing between internal and external (physical) networks (including NAT) or between different sites and/or clouds (private or public) which use an IPSec VPN or GRE tunnel.

Gateways can come in different physical form factors. They can be built on Windows Server 2016, incorporated into a Top of Rack (TOR) switch acting as a VXLAN Gateway, accessed through a Virtual IP (VIP) advertised by a load balancer, put into other existing network appliances, or can be a new stand-alone network appliance.

For more information about Windows RAS Gateway options, see [RAS Gateway](#).

Packet Encapsulation

Each virtual network adapter in HNV is associated with two IP addresses:

- **Customer Address (CA)** The IP address assigned by the customer, based on their intranet infrastructure. This address allows the customer to exchange network traffic with the virtual machine as if it had not been moved to a public or private cloud. The CA is visible to the virtual machine and reachable by the customer.
- **Provider Address (PA)** The IP address assigned by the hosting provider or the datacenter administrators based on their physical network infrastructure. The PA appears in the packets on the network that are exchanged with the server running Hyper-V that is hosting the virtual machine. The PA is visible on the physical network, but not to the virtual machine.

The CAs maintain the customer's network topology, which is virtualized and decoupled from the actual underlying physical network topology and addresses, as implemented by the PAs. The following diagram shows the conceptual relationship between virtual machine CAs and network infrastructure PAs as a result of network virtualization.

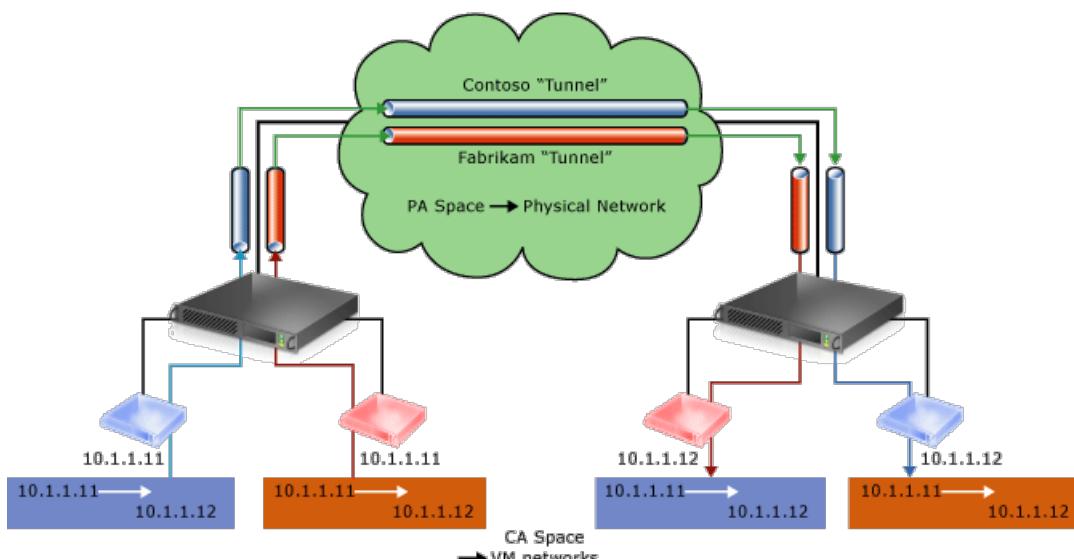


Figure 6: Conceptual diagram of network virtualization over physical infrastructure

In the diagram, customer virtual machines are sending data packets in the CA space, which traverse the physical network infrastructure through their own virtual networks, or "tunnels". In the example above, the tunnels can be thought of as "envelopes" around the Contoso and Fabrikam data packets with green shipping labels (PA addresses) to be delivered from the source host on the left to the destination host on the right. The key is how the hosts determine the "shipping addresses" (PA's) corresponding to the Contoso and the Fabrikam CA's, how the "envelope" is put around the packets, and how the destination hosts can unwrap the packets and deliver to the Contoso and Fabrikam destination virtual machines correctly.

This simple analogy highlighted the key aspects of network virtualization:

- Each virtual machine CA is mapped to a physical host PA. There can be multiple CAs associated with the same PA.
- Virtual machines send data packets in the CA spaces, which are put into an "envelope" with a PA source and destination pair based on the mapping.
- The CA-PA mappings must allow the hosts to differentiate packets for different customer virtual machines.

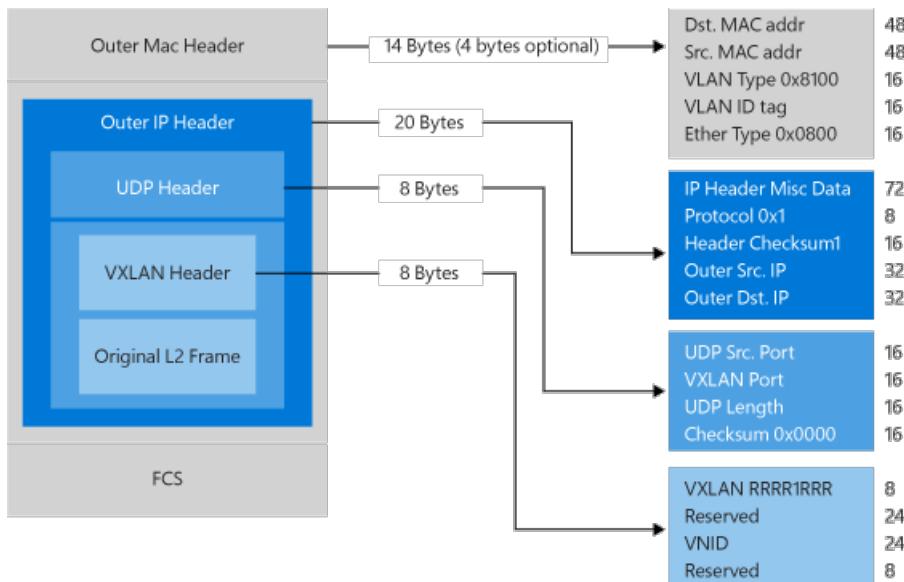
As a result, the mechanism to virtualize the network is to virtualize the network addresses used by the virtual machines. The network controller is responsible for the address mapping, and the host agent maintains the mapping database using the MS_VTEP schema. The next section describes the actual mechanism of address virtualization.

Network virtualization through address virtualization

HNV implements overlay tenant networks using either Network Virtualization Generic Routing Encapsulation (NVGRE) or the Virtual eXtensible Local Area Network (VXLAN). VXLAN is the default.

Virtual eXtensible Local Area Network (VXLAN)

The Virtual eXtensible Local Area Network (VXLAN) ([RFC 7348](#)) protocol has been widely adopted in the market place, with support from vendors like Cisco, Brocade, Arista, Dell, HP and others. The VXLAN protocol uses UDP as the transport. The IANA-assigned UDP destination port for VXLAN is 4789 and the UDP source port should be a hash of information from the inner packet to be used for ECMP spreading. After the UDP header, a VXLAN header is appended to the packet which includes a reserved 4-byte field followed by a 3-byte field for the VXLAN Network Identifier (VNI) - VSID - followed by another reserved 1-byte field. After the VXLAN header, the original CA L2 frame (without the CA Ethernet frame FCS) is appended.



Generic Routing Encapsulation (NVGRE)

This network virtualization mechanism uses the Generic Routing Encapsulation (NVGRE) as part of the tunnel header. In NVGRE, the virtual machine's packet is encapsulated inside another packet. The header of this new

packet has the appropriate source and destination PA IP addresses in addition to the Virtual Subnet ID, which is stored in the Key field of the GRE header, as shown in Figure 7.

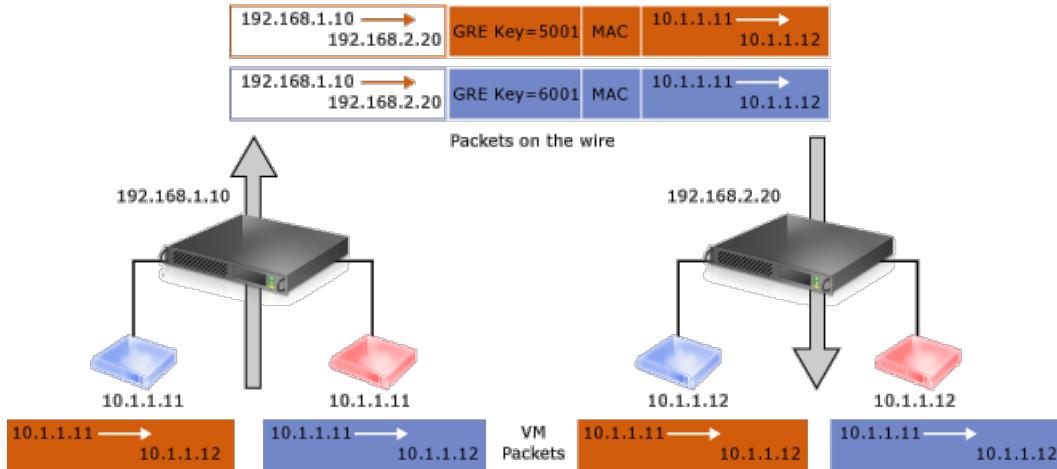


Figure 7: Network virtualization - NVGRE encapsulation

The Virtual Subnet ID allows hosts to identify the customer virtual machine for any given packet, even though the PA's and the CA's on the packets may overlap. This allows all virtual machines on the same host to share a single PA, as shown in Figure 7.

Sharing the PA has a big impact on network scalability. The number of IP and MAC addresses that need to be learned by the network infrastructure can be substantially reduced. For instance, if every end host has an average of 30 virtual machines, the number of IP and MAC addresses that need to be learned by the networking infrastructure is reduced by a factor of 30. The embedded Virtual Subnet IDs in the packets also enable easy correlation of packets to the actual customers.

The PA sharing scheme for Windows Server 2012 R2 is one PA per VSID per host. For Windows Server 2016 the scheme is one PA per NIC team member.

With Windows Server 2016 and later, HNV fully supports NVGRE and VXLAN out of the box; it does NOT require upgrading or purchasing new network hardware such as NICs (network adapters), switches, or routers. This is because these packets on the wire are regular IP packet in the PA space, which is compatible with today's network infrastructure. However, to get the best performance use supported NICs with the latest drivers that support task offloads.

Multi-tenant deployment example

The following diagram shows an example deployment of two customers located in a cloud datacenter with the CA-PA relationship defined by the network policies.

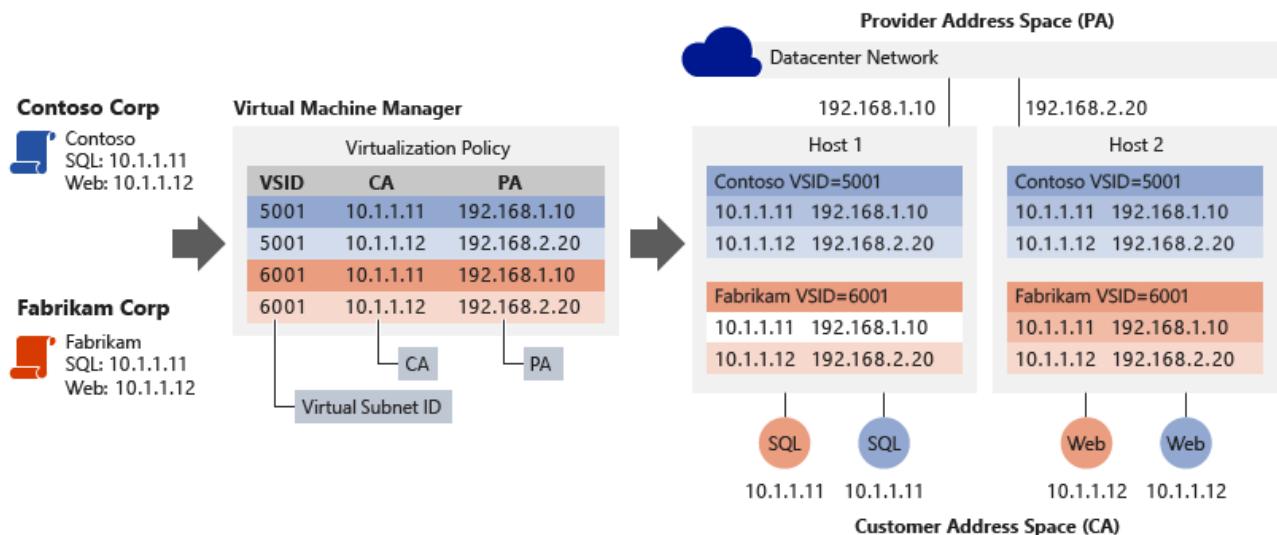


Figure 8: Multi-tenant deployment example

Consider the example in Figure 8. Prior to moving to the hosting provider's shared IaaS service:

- Contoso Corp ran a SQL Server (named **SQL**) at the IP address 10.1.1.11 and a web server (named **Web**) at the IP address 10.1.1.12, which uses its SQL Server for database transactions.
- Fabrikam Corp ran a SQL Server, also named **SQL** and assigned the IP address 10.1.1.11, and a web server, also named **Web** and also at the IP address 10.1.1.12, that uses its SQL Server for database transactions.

We will assume that the hosting service provider has previously created the provider (PA) logical network through the Network Controller to correspond to their physical network topology. The Network Controller allocates two PA IP addresses from the logical subnet's IP prefix where the hosts are connected. The network controller also indicates the appropriate VLAN tag to apply the IP addresses.

Using the Network Controller, Contoso Corp and Fabrikam Corp then create their virtual network and subnets which are backed by the provider (PA) logical network specified by the hosting service provider. Contoso Corp and Fabrikam Corp move their respective SQL Servers and web servers to the same hosting provider's shared IaaS service where, coincidentally, they run the **SQL** virtual machines on Hyper-V Host 1 and the **Web** (IIS7) virtual machines on Hyper-V Host 2. All virtual machines maintain their original intranet IP addresses (their CAs).

Both companies are assigned the following Virtual Subnet ID (VSID) by the Network Controller as indicated below. The Host Agent on each of the Hyper-V hosts receives the allocated PA IP addresses from the Network Controller and creates two PA host vNICs in a non-default network compartment. A network interface is assigned to each of these host vNICs where the PA IP address is assigned as shown below:

- Contoso Corp's virtual machines VSID and PAs : **VSID** is 5001, **SQL PA** is 192.168.1.10, **Web PA** is 192.168.2.20
- Fabrikam Corp's virtual machines VSID and PAs: **VSID** is 6001, **SQL PA** is 192.168.1.10, **Web PA** is 192.168.2.20

The Network Controller plumbs all network policy (including CA-PA mapping) to the SDN Host Agent which will maintain the policy in a persistent store (in OVSDB database tables).

When the Contoso Corp Web virtual machine (10.1.1.12) on Hyper-V Host 2 creates a TCP connection to the SQL Server at 10.1.1.11, the following happens:

- VM ARPs for the destination MAC address of 10.1.1.11
- The VFP extension in the vSwitch intercepts this packet and sends it to the SDN Host Agent
- The SDN Host Agent looks in its policy store for the MAC address for 10.1.1.11
- If a MAC is found, the Host Agent injects an ARP response back to the VM
- If a MAC is not found, no response is sent and the ARP entry in the VM for 10.1.1.11 is marked unreachable.
- The VM now constructs a TCP packet with the correct CA Ethernet and IP headers and sends it to the vSwitch
- The VFP forwarding extension in the vSwitch processes this packet through the VFP layers (described below) assigned to the source vSwitch port on which the packet was received and creates a new flow-entry in the VFP unified flow table
- The VFP engine performs rule matching or flow-table lookup for each layer (e.g. virtual network layer) based on the IP and Ethernet headers.
- The matched rule in the virtual network layer references a CA-PA mapping space and performs

encapsulation.

- The encapsulation type (either VXLAN or NVGRE) is specified in the VNet layer along with the VSID.
- In the case of VXLAN encapsulation, an outer UDP header is constructed with the VSID of 5001 in the VXLAN header.
An outer IP header is constructed with the source and destination PA address assigned to the Hyper-V Host 2 (192.168.2.20) and Hyper-V Host 1 (192.168.1.10) respectively based on the SDN Host Agent's policy store.
- This packet then flows to the PA routing layer in VFP.
- The PA routing layer in VFP will reference the network compartment used for PA-space traffic and a VLAN ID and use the TCP/IP stack of the host to forward the PA packet to Hyper-V Host 1 correctly.
- Upon receipt of the encapsulated packet, Hyper-V Host 1 receives the packet in the PA network compartment and forward it to the vSwitch.
- The VFP processes the packet through its VFP layers and create a new flow-entry in the VFP unified flow table.
- The VFP engine matches the ingres rules in the virtual network layer and strips off the outer encapsulated packet's Ethernet, IP, and VXLAN headers.
- The VFP engine then forwards the packet to the vSwitch port to which the destination VM is connected.

A similar process for traffic between the Fabrikam Corp **Web** and **SQL** virtual machines uses the HNV policy settings for the Fabrikam Corp. As a result, with HNV, Fabrikam Corp and Contoso Corp virtual machines interact as if they were on their original intranets. They can never interact with each other, even though they are using the same IP addresses.

The separate addresses (CAs and PAs), the policy settings of the Hyper-V hosts, and the address translation between the CA and the PA for inbound and outbound virtual machine traffic isolate these sets of servers using either the NVGRE Key or the VLXAN VNID. Furthermore, the virtualization mappings and transformation decouples the virtual network architecture from the physical network infrastructure. Although Contoso **SQL** and **Web** and Fabrikam **SQL** and **Web** reside in their own CA IP subnets (10.1.1/24), their physical deployment happens on two hosts in different PA subnets, 192.168.1/24 and 192.168.2/24, respectively. The implication is that cross-subnet virtual machine provisioning and live migration become possible with HNV.

Hyper-V Network Virtualization architecture

In Windows Server 2016, HNVv2 is implemented using the Azure Virtual Filtering Platform (VFP) which is an NDIS filtering extension within the Hyper-V Switch. The key concept of VFP is that of a Match-Action flow engine with an internal API exposed to the SDN Host Agent for programming network policy. The SDN Host Agent itself receives network policy from the Network Controller over the OVSDDB and WCF SouthBound communication channels. Not only is virtual network policy (e.g. CA-PA mapping) programmed using VFP but additional policy such as ACLs, QoS, and so on.

The object hierarchy for the vSwitch and VFP forwarding extension is the following:

- vSwitch
 - External NIC Management
 - NIC Hardware Offloads
 - Global Forwarding rules
 - Port

- Egress forwarding layer for hair-pinning
- Space lists for mappings and NAT pools
- Unified Flow Table
- VFP Layer
 - Flow table
 - Group
 - Rule
 - Rules can reference spaces

In the VFP, a layer is created per policy type (for example, Virtual Network) and is a generic set of rule/flow tables. It does not have any intrinsic functionality until specific rules are assigned to that layer to implement such functionality. Each layer is assigned a priority and layers are assigned to a port by ascending priority. Rules are organized into groups based primarily on direction and IP address family. Groups are also assigned a priority and at most, one rule from a group can match a given flow.

The forwarding logic for the vSwitch with VFP extension is as follows:

- Ingress processing (ingress from the point of view of packet coming into a port)
- Forwarding
- Egress processing (egress from the point of view of packet leaving a port)

The VFP supports inner MAC forwarding for NVGRE and VXLAN encapsulation types as well as outer MAC VLAN based forwarding.

The VFP extension has a slow-path and fast-path for packet traversal. The first packet in a flow must traverse all rule groups in each layer and do a rule lookup which is an expensive operation. However, once a flow is registered in the unified flow table with a list of actions (based on the rules matched) all subsequent packets will be processed based on the unified flow table entries.

HNV policy is programmed by the host agent. Each virtual machine network adapter is configured with an IPv4 address. These are the CAs that will be used by the virtual machines to communicate with each other, and they are carried in the IP packets from the virtual machines. HNV encapsulates the CA frame in a PA frame based on the network virtualization policies stored in the host agent's database.

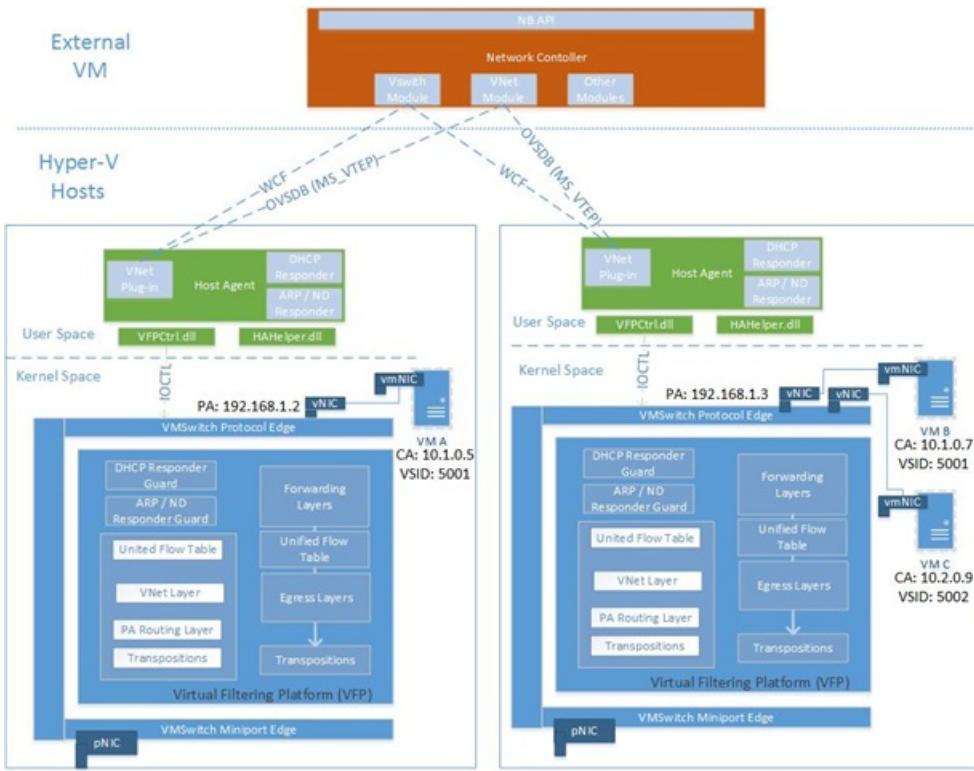


Figure 9: HNV Architecture

Summary

Cloud-based datacenters can provide many benefits such as improved scalability and better resource utilization. To realize these potential benefits requires a technology that fundamentally addresses the issues of multi-tenant scalability in a dynamic environment. HNV was designed to address these issues and also improve the operational efficiency of the datacenter by decoupling the virtual network topology for the physical network topology. Building on an existing standard, HNV runs in today's datacenter and operates with your existing VXLAN infrastructure. Customers with HNV can now consolidate their datacenters into a private cloud or seamlessly extend their datacenters to a hosting server provider's environment with a hybrid cloud.

See also

To learn more about HNVv2 see the following links:

CONTENT TYPE	REFERENCES
Community Resources	<ul style="list-style-type: none"> - Private Cloud Architecture Blog - Ask questions: cloudnetfb@microsoft.com
RFC	<ul style="list-style-type: none"> - NVGRE Draft RFC - VXLAN - RFC 7348
Related Technologies	<ul style="list-style-type: none"> - For Hyper-V Network Virtualization technical details in Windows Server 2012 R2 , see Hyper-V Network Virtualization technical details - Network Controller

What's New in Hyper-V Network Virtualization in Windows Server 2016

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

This topic describes the Hyper-V Network Virtualization (HNV) functionality that is new or changed in Windows Server 2016.

Updates in HNV

HNV offers enhanced support in the following areas:

FEATURE/FUNCTIONALITY	NEW OR IMPROVED	DESCRIPTION
Programmable Hyper-V switch	New	HNV policy is programmable through the Microsoft Network Controller.
VXLAN encapsulation support	New	HNV now supports VXLAN encapsulation.
Software Load Balancer (SLB) interoperability	New	HNV is fully integrated with the Microsoft Software Load Balancer.
Compliant IEEE Ethernet headers	Improved	Compliant with IEEE Ethernet standards

Programmable Hyper-V switch

HNV is a fundamental building block of Microsoft's updated Software Defined Networking (SDN) solution, and is fully integrated into the SDN stack.

Microsoft's new Network Controller pushes HNV policies down to a Host Agent running on each host using Open vSwitch Database Management Protocol (OVSDB) as the SouthBound Interface (SBI). The Host Agent stores this policy using a customization of the [VTEP schema](#) and programs complex flow rules into a performant flow engine in the Hyper-V switch.

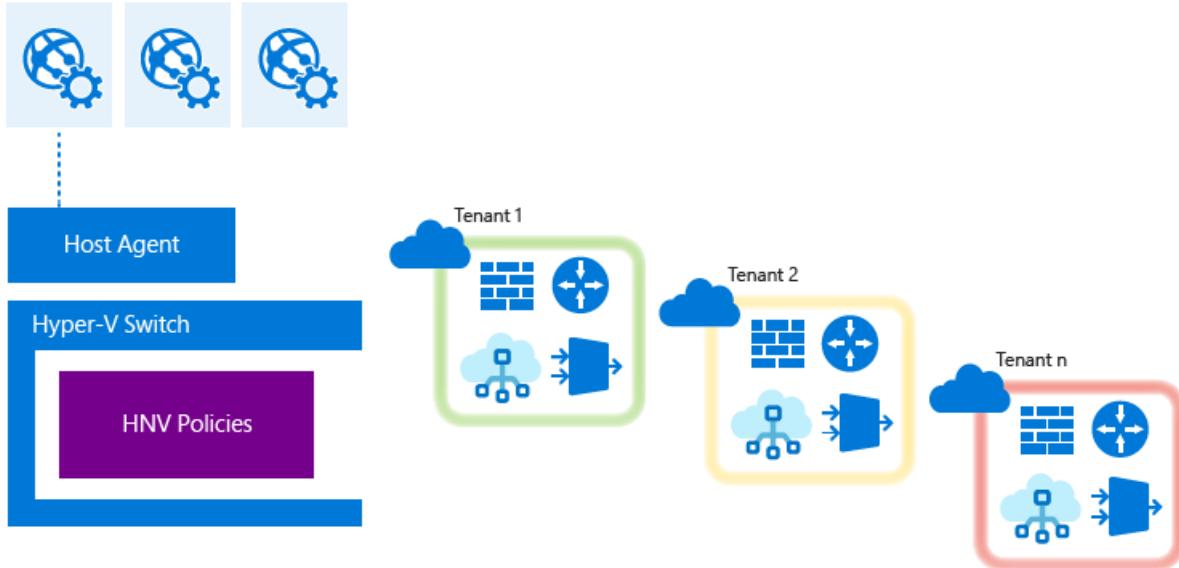
The flow engine inside the Hyper-V switch is the same engine used in Microsoft Azure™, which has been proven at hyper-scale in the Microsoft Azure public cloud. Additionally, the entire SDN stack up through the Network Controller, and Network Resource Provider (details coming soon) is consistent with Microsoft Azure, thus bringing the power of the Microsoft Azure public cloud to our enterprise and hosting service provider customers.

NOTE

For more information about OVSDB, see [RFC 7047](#).

The Hyper-V switch supports both stateless and stateful flow rules based on simple 'match action' within Microsoft's flow engine.

Network Controllers



VXLAN encapsulation support

The Virtual eXtensible Local Area Network (VXLAN - [RFC 7348](#)) protocol has been widely adopted in the market place, with support from vendors like Cisco, Brocade, Dell, HP and others. HNV also now supports this encapsulation scheme using MAC distribution mode through the Microsoft Network Controller to program mappings for tenant overlay network IP addresses (Customer Address, or CA) to the physical underlay network IP addresses (Provider Address, or PA). Both NVGRE and VXLAN Task Offloads are supported for improved performance through third-party drivers.

Software Load Balancer (SLB) interoperability

Windows Server 2016 includes a software load balancer (SLB) with full support for virtual network traffic and seamless interaction with HNV. The SLB is implemented through the performant flow engine in the data plane v-Switch and controlled by the Network Controller for Virtual IP (VIP) / Dynamic IP (DIP) mappings.

Compliant IEEE Ethernet headers

HNV implements correct L2 Ethernet headers to ensure interoperability with third-party virtual and physical appliances that depend on industry-standard protocols. Microsoft ensures that all transmitted packets have compliant values in all fields to ensure this interoperability. In addition, support for Jumbo Frames (MTU > 1780) in the physical L2 network will be required to account for packet overhead introduced by encapsulation protocols (NVGRE, VXLAN) while ensuring guest Virtual Machines attached to an HNV Virtual Network maintain a 1514 MTU.

See also

- [Hyper-V Network Virtualization Overview](#)
- [Hyper-V Network Virtualization technical details](#)
- [Software Defined Networking](#)

Internal DNS Service (iDNS) for SDN

9/1/2018 • 6 minutes to read • [Edit Online](#)

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

If you work for a Cloud Service Provider (CSP) or Enterprise that is planning to deploy Software Defined Networking (SDN) in Windows Server 2016, you can provide DNS services to your hosted tenant workloads by using Internal DNS (iDNS), which is integrated with SDN.

Hosted virtual machines (VMs) and applications require DNS to communicate within their own networks and with external resources on the Internet. With iDNS, you can provide tenants with DNS name resolution services for their isolated, local name space and for Internet resources.

Because the iDNS service is not accessible from tenant Virtual Networks, other than through the iDNS proxy, the server is not vulnerable to malicious activities on tenant networks.

Key Features

Following are the key features for iDNS.

- Provides shared DNS name resolution services for tenant workloads
- Authoritative DNS service for name resolution and DNS registration within the tenant name space
- Recursive DNS service for resolution of Internet names from tenant VMs.
- If desired, you can configure simultaneous hosting of fabric and tenant names
- A cost-effective DNS solution - tenants do not need to deploy their own DNS infrastructure
- High availability with Active Directory integration, which is required.

In addition to these features, if you are concerned about keeping your AD integrated DNS servers open to the Internet, you can deploy iDNS servers behind another recursive resolver in the perimeter network.

Because iDNS is a centralized server for all DNS queries, a CSP or Enterprise can also implement tenant DNS firewalls, apply filters, detect malicious activities, and audit transactions at a central location

iDNS Infrastructure

The iDNS infrastructure includes iDNS Servers and iDNS proxy.

iDNS Servers

iDNS includes a set of DNS servers that host tenant-specific data, such as VM DNS Resource Records.

iDNS servers are the authoritative servers for their internal DNS zones, and also act as a resolver for public names when tenant VMs attempt to connect to external resources.

All of the host names for VMs on Virtual Networks are stored as DNS Resource Records under the same zone. For example, if you deploy iDNS for a zone named contoso.local, the DNS Resource Records for the VMs on that network are stored in the contoso.local zone.

Tenant VM Fully Qualified Domain Names (FQDNs) consist of the computer name and the DNS suffix string for the Virtual Network, in GUID format. For example, if you have a tenant VM named TENANT1 that is on the Virtual Network contoso.local, the VM's FQDN is TENANT1.vn-guid.contoso.local, where *vn-guid* is the DNS suffix string for the Virtual Network.

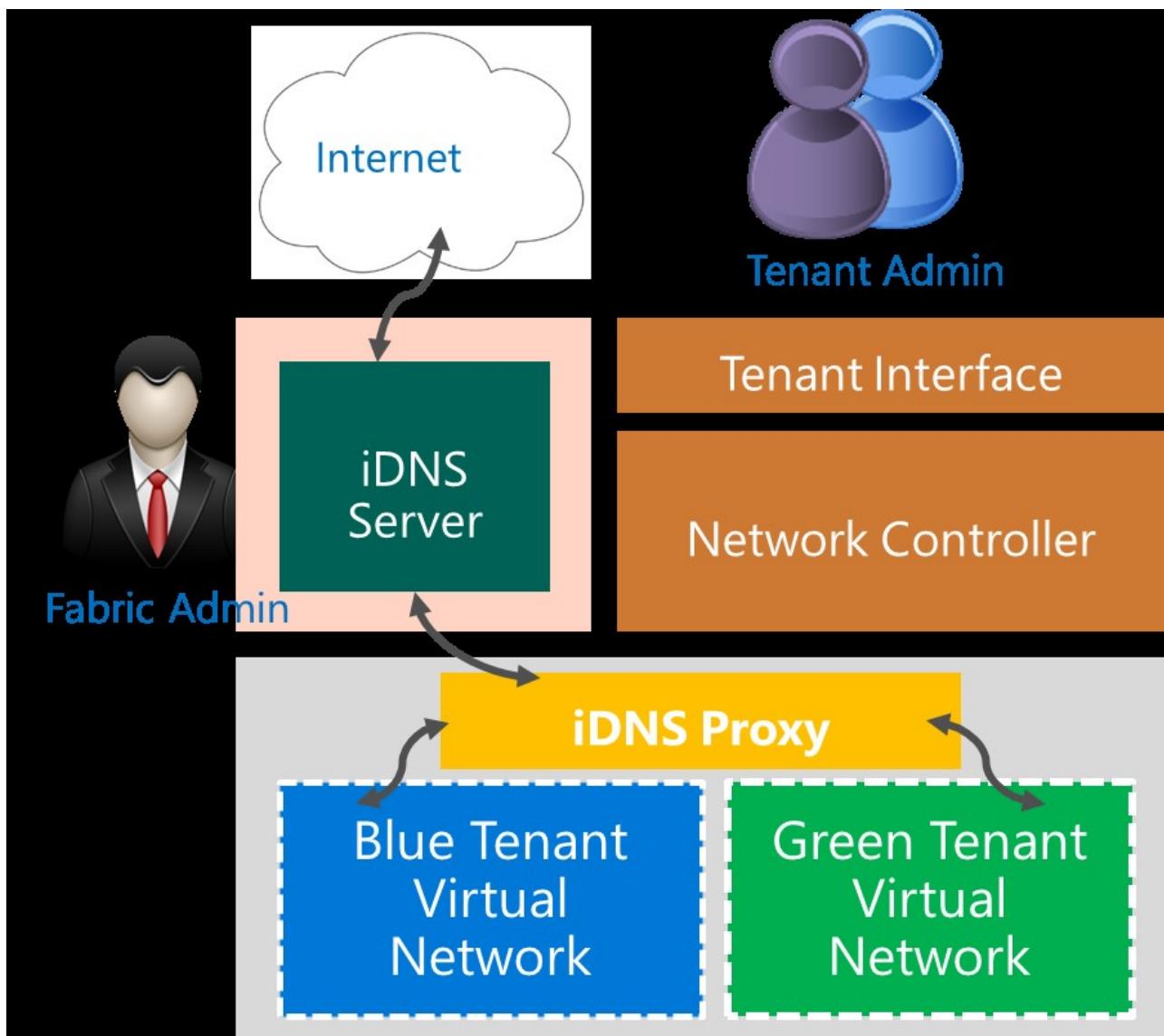
NOTE

If you are a fabric administrator, you can use your CSP or Enterprise DNS infrastructure as iDNS servers instead of deploying new DNS servers specifically for use as iDNS servers. Whether you deploy new servers for iDNS or you use your existing infrastructure, iDNS relies on Active Directory to provide high availability. Your iDNS servers must therefore be integrated with Active Directory.

iDNS Proxy

iDNS proxy is a Windows service that runs on every host, and which forwards tenant Virtual Network DNS traffic to the iDNS Server and the Internet.

The following illustration depicts DNS traffic paths from tenant Virtual Networks through the iDNS proxy to the iDNS Server and the Internet.



How to Deploy iDNS

When you deploy SDN in Windows Server 2016 by using scripts, iDNS is automatically included in your deployment.

For more information, see the following topics.

- [Deploy a Software Defined Network infrastructure using scripts](#)

Understanding iDNS Deployment Steps

You can use this section to gain an understanding of how iDNS is installed and configured when you deploy SDN using scripts.

Following is a summary of the steps needed to deploy iDNS.

NOTE

If you have deployed SDN by using scripts, you do not need to perform any of these steps. The steps are provided for information and troubleshooting purposes only.

Step 1: Deploy DNS

You can deploy a DNS server by using the following example Windows PowerShell command.

```
Install-WindowsFeature DNS -IncludeManagementTools
```

Step 2: Configure iDNS information in Network Controller

This script segment is a REST call that is made by the administrator to Network Controller, informing it about the iDNS zone configuration - such as the IP address of the iDnsServer and the zone that is used to host the iDNS names.

```
Url: https://<url>/networking/v1/iDnsServer/configuration
Method: PUT
{
  "properties": {
    "connections": [
      {
        "managementAddresses": [
          "10.0.0.9"
        ],
        "credential": {
          "resourceRef": "/credentials/iDnsServer-Credentials"
        },
        "credentialType": "usernamePassword"
      }
    ],
    "zone": "contoso.local"
  }
}
```

NOTE

This is an excerpt from the section **Configuration ConfigureIDns** in SDNExpress.ps1. For more information, see [Deploy a Software Defined Network infrastructure using scripts](#).

Step 3: Configure the iDNS Proxy Service

The the iDNS Proxy Service runs on each of the Hyper-V hosts, providing the bridge between the virtual networks of tenants and the physical network where the iDNS servers are located. The following registry keys must be created on every Hyper-V host.

DNS port: Fixed port 53

- Registry Key =

HKLM\SYSTEM\CurrentControlSet\Services\NcHostAgent\Parameters\Plugins\Vnet\InfraServices\DnsProxyService"

- ValueName = "Port"
- ValueData = 53
- ValueType = "Dword"

DNS Proxy Port: Fixed port 53

- Registry Key =

HKLM\SYSTEM\CurrentControlSet\Services\NcHostAgent\Parameters\Plugins\Vnet\InfraServices\DnsProxyService"
- ValueName = "ProxyPort"
- ValueData = 53
- ValueType = "Dword"

DNS IP: Fixed IP address configured on the network interface, in case the tenant chooses to use the iDNS service

- Registry Key =

HKLM\SYSTEM\CurrentControlSet\Services\NcHostAgent\Parameters\Plugins\Vnet\InfraServices\DnsProxyService"
- ValueName = "IP"
- ValueData = "169.254.169.254"
- ValueType = "String"

Mac Address: Media Access Control address of the DNS server

- Registry Key =

HKLM\SYSTEM\CurrentControlSet\Services\NcHostAgent\Parameters\Plugins\Vnet\InfraServices\DnsProxyService"
- ValueName = "MAC"
- ValueData = "aa-bb-cc-aa-bb-cc"
- ValueType = "String"

IDNS Server Address: A comma separated list of iDNS Servers.

- Registry Key: HKLM\SYSTEM\CurrentControlSet\Services\DNSProxy\Parameters
- ValueName = "Forwarders"
- ValueData = "10.0.0.9"
- ValueType = "String"

NOTE

This is an excerpt from the section **Configuration ConfigureIDnsProxy** in SDNExpress.ps1. For more information, see [Deploy a Software Defined Network infrastructure using scripts](#).

Step 4: Restart the Network Controller Host Agent Service

You can use the following Windows PowerShell command to restart the Network Controller Host Agent Service.

```
Restart-Service nhostagent -Force
```

For more information, see [Restart-Service](#).

Enable firewall rules for the DNS proxy service

You can use the following Windows PowerShell command to create a firewall rule that allows exceptions for the proxy to communicate with the VM and the iDNS server.

```
Enable-NetFirewallRule -DisplayGroup 'DNS Proxy Firewall'
```

For more information, see [Enable-NetFirewallRule](#).

Validate the iDNS Service

To validate the iDNS Service, you must deploy a sample tenant workload.

For more information, see [Create a VM and Connect to a Tenant Virtual Network or VLAN](#).

If you want the tenant VM to use the iDNS service, you must leave the VM network interfaces DNS Server configuration blank and allow the interfaces to use DHCP.

After the VM with such a network interface is initiated, it automatically receives a configuration that allows the VM to use iDNS, and the VM immediately starts performing name resolution by using the iDNS service.

If you configure the tenant VM to use the iDNS service by leaving network interface DNS Server and Alternate DNS Server information blank, Network Controller provides the VM with an IP address, and performs a DNS name registration on behalf of the VM with the iDNS Server.

Network Controller also informs the iDNS proxy about the VM and the required details to perform name resolution for the VM.

When the VM initiates a DNS query, the proxy acts as a forwarder of the query from the Virtual Network to the iDNS service.

The DNS proxy also ensures that the tenant VM queries are isolated. If the iDNS server is authoritative for the query, the iDNS server responds with an authoritative response. If the iDNS server is not authoritative for the query, it performs a DNS recursion to resolve Internet names.

NOTE

This information is included in the section **Configuration AttachToVirtualNetwork** in SDNExpressTenant.ps1. For more information, see [Deploy a Software Defined Network infrastructure using scripts](#).

Network Controller

9/1/2018 • 5 minutes to read • [Edit Online](#)

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

New in Windows Server 2016, Network Controller provides a centralized, programmable point of automation to manage, configure, monitor, and troubleshoot virtual and physical network infrastructure in your datacenter.

Using Network Controller, you can automate the configuration of network infrastructure instead of performing manual configuration of network devices and services.

NOTE

In addition to this topic, the following Network Controller documentation is available.

- [Network Controller High Availability](#)
- [Installation and Preparation Requirements for Deploying Network Controller](#)
- [Deploy Network Controller using Windows PowerShell](#)
- [Install the Network Controller server role using Server Manager](#)
- [Post-Deployment Steps for Network Controller](#)
- [Network Controller Cmdlets](#)

Network Controller Overview

Network Controller is a highly available and scalable server role, and provides one application programming interface (API) that allows Network Controller to communicate with the network, and a second API that allows you to communicate with Network Controller.

You can deploy Network Controller in both domain and non-domain environments. In domain environments, Network Controller authenticates users and network devices by using Kerberos; in non-domain environments, you must deploy certificates for authentication.

IMPORTANT

Do not deploy the Network Controller server role on physical hosts. To deploy Network Controller, you must install the Network Controller server role on a Hyper-V virtual machine (VM) that is installed on a Hyper-V host. After you have installed Network Controller on VMs on three different Hyper-V hosts, you must enable the Hyper-V hosts for Software Defined Networking (SDN) by adding the hosts to Network Controller using the Windows PowerShell command **New-NetworkControllerServer**. By doing so, you are enabling the SDN Software Load Balancer to function. For more information, see [New-NetworkControllerServer](#).

Network Controller communicates with network devices, services, and components by using the Southbound API. With the Southbound API, Network Controller can discover network devices, detect service configurations, and gather all of the information you need about the network. In addition, the Southbound API gives Network Controller a pathway to send information to the network infrastructure, such as configuration changes that you have made.

The Network Controller Northbound API provides you with the ability to gather network information from Network Controller and use it to monitor and configure the network.

The Network Controller Northbound API allows you to configure, monitor, troubleshoot, and deploy new devices on the network by using Windows PowerShell, the Representational State Transfer (REST) API, or a management application with a graphical user interface, such as System Center Virtual Machine Manager.

NOTE

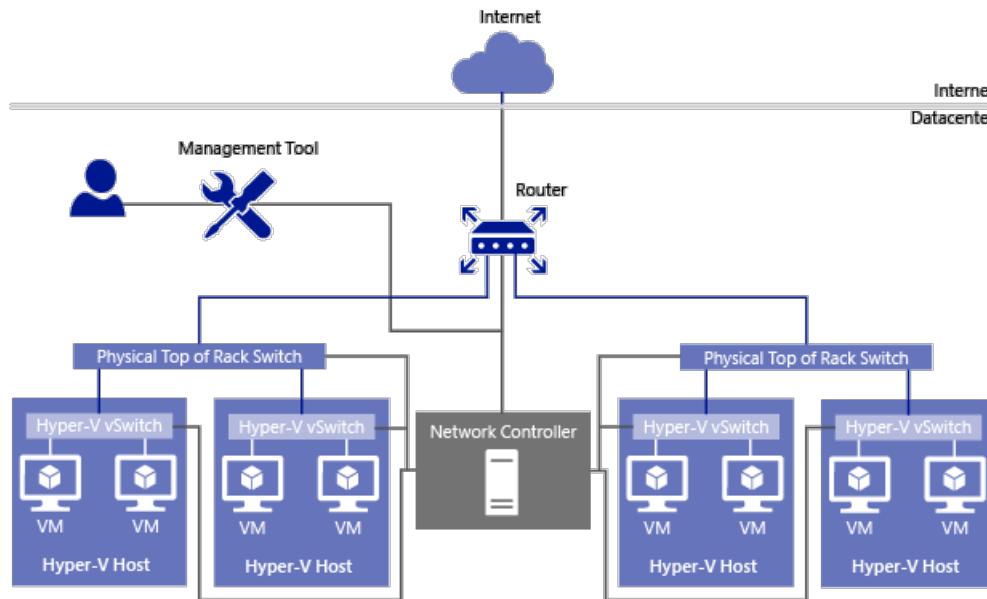
The Network Controller Northbound API is implemented as a REST interface.

You can manage your datacenter network with Network Controller by using management applications, such as System Center Virtual Machine Manager (SCVMM), and System Center Operations Manager (SCOM), because Network Controller allows you to configure, monitor, program, and troubleshoot the network infrastructure that is under its control.

Using Windows PowerShell, the REST API, or a management application, you can use Network Controller to manage the following physical and virtual network infrastructure:

- Hyper-V VMs and virtual switches
- Datacenter Firewall
- Remote Access Service (RAS) Multitenant Gateways, Virtual Gateways, and gateway pools
- Software Load Balancers

In the following illustration, an Administrator uses a Management Tool that interacts directly with Network Controller. Network Controller provides information about the network infrastructure, including both virtual and physical infrastructure, to the Management Tool, and makes configuration changes according to the Administrator's actions when using the tool.



If you are deploying Network Controller in a test lab environment, you can run the Network Controller server role on a Hyper-V virtual machine (VM) that is installed on a Hyper-V host.

For high availability in larger datacenters, you can deploy a cluster by using three VMs that are installed on three or more Hyper-V hosts. For more information, see [Network Controller High Availability](#).

Network Controller Features

The following Network Controller features allow you to configure and manage virtual and physical network devices and services.

- [Firewall Management](#)
- [Software Load Balancer Management](#)
- [Virtual Network Management](#)
- [RAS Gateway Management](#)

IMPORTANT

Network Controller Backup and Restore is not currently available in Windows Server 2016.

Firewall Management

This Network Controller feature allows you to configure and manage allow/deny firewall Access Control rules for your workload VMs for both East/West and North/South network traffic in your datacenter. The firewall rules are plumbed in the vSwitch port of workload VMs, and so they are distributed across your workload in the datacenter. Using the Northbound API, you can define the firewall rules for both incoming and outgoing traffic from the workload VM. You can also configure each firewall rule to log the traffic that was allowed or denied by the rule.

For more information, see [Datacenter Firewall Overview](#).

Software Load Balancer Management

This Network Controller feature allows you to enable multiple servers to host the same workload, providing high availability and scalability.

For more information, see [Software Load Balancing \(SLB\) for SDN](#).

Virtual Network Management

This Network Controller feature allows you to deploy and configure Hyper-V Network Virtualization, including the Hyper-V Virtual Switch and virtual network adapters on individual VMs, and to store and distribute virtual network policies.

Network Controller supports both Network Virtualization Generic Routing Encapsulation (NVGRE) and Virtual Extensible Local Area Network (VXLAN).

RAS Gateway Management

This Network Controller feature allows you to deploy, configure, and manage virtual machines (VMs) that are members of a RAS Gateway pool, providing gateway services to your tenants. Network Controller allows you to automatically deploy VMs running RAS Gateway with the following gateway features:

NOTE

In System Center Virtual Machine Manager, RAS Gateway is named Windows Server Gateway.

- Add and remove gateway VMs from the cluster and specify the level of backup required.
- Site-to-site virtual private network (VPN) gateway connectivity between remote tenant networks and your datacenter using IPsec.
- Site-to-site VPN gateway connectivity between remote tenant networks and your datacenter using Generic Routing Encapsulation (GRE).
- Layer 3 forwarding capability.
- Border Gateway Protocol (BGP) routing, which allows you to manage the routing of network traffic between your tenants' VM networks and their remote sites.

Network Controller can place different connections of a tenant on separate gateways. You can use a single public IP for all gateway connections or have different public IPs for a subset of the connections. Network Controller logs all gateway configuration and state changes, which can be used for auditing and troubleshooting purposes.

For more information on BGP, see [Border Gateway Protocol \(BGP\)](#).

For more information on the RAS Gateway, see [RAS Gateway for SDN](#).

Network Controller Deployment Options

To deploy Network Controller by using System Center Virtual Machine Manager (VMM), see [Set up an SDN Network Controller in the VMM fabric](#).

To deploy Network Controller using scripts, see [Deploy a Software Defined Network Infrastructure Using Scripts](#).

To deploy Network Controller using Windows PowerShell, see [Deploy Network Controller using Windows PowerShell](#)

Network Controller High Availability

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

You can use this topic to learn about Network Controller high availability and scalability configuration for Software Defined Networking (SDN).

When you deploy SDN in your datacenter, you can use Network Controller to centrally deploy, monitor, and manage many network elements, including RAS Gateways, Software Load Balancers, virtual networking policies for tenant communication, Datacenter Firewall policies, Quality of Service (QoS) for SDN policies, hybrid networking policies, and more.

Because Network Controller is the cornerstone of SDN management, it is critical for Network Controller deployments to provide high availability and the ability for you to easily scale up or down Network Controller nodes with your datacenter needs.

Although you can deploy Network Controller as a single machine cluster, for high availability and failover you must deploy Network Controller in a multiple machine cluster with a minimum of three machines.

NOTE

You can deploy Network Controller on either server computers or on virtual machines (VMs) that are running Windows Server 2016 Datacenter edition. If you deploy Network Controller on VMs, the VMs must be running on Hyper-V hosts that are also running Datacenter edition. Network Controller is not available on Windows Server 2016 Standard edition.

Network Controller as a Service Fabric Application

To achieve high availability and scalability, Network Controller relies on Service Fabric. Service Fabric provides a distributed systems platform to build scalable, reliable, and easily-managed applications.

As a platform, Service Fabric provides functionality that is required for building a scalable distributed system. It provides service hosting on multiple operating system instances, synchronizing state information between instances, electing a leader, failure detection, load balancing, and more.

NOTE

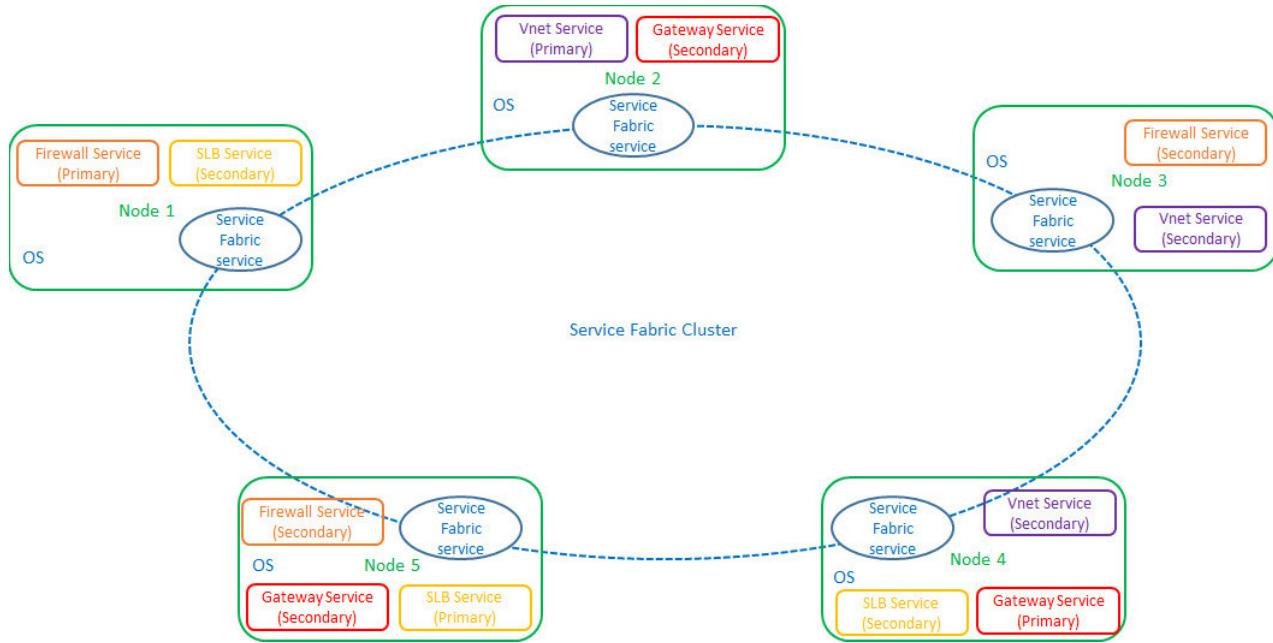
For information about Service Fabric in Azure, see [Overview of Azure Service Fabric](#).

When you deploy Network Controller on multiple machines, Network Controller runs as a single Service Fabric application on a Service Fabric cluster. You can form a Service Fabric cluster by connecting a set of operating system instances.

The Network Controller application is comprised of multiple stateful Service Fabric services. Each service is responsible for a network function, such as physical network management, virtual network management, firewall management, or gateway management.

Each Service Fabric service has one primary replica and two secondary replicas. The primary service replica processes requests, while the two secondary service replicas provide high availability in circumstances where the primary replica is disabled or unavailable for some reason.

The following illustration depicts a Network Controller Service Fabric cluster with five machines. Four services are distributed across the five machines: Firewall Service, Gateway Service, Software Load Balancing (SLB) service, and virtual network (Vnet) service. Each of the four services includes one primary service replica and two secondary service replicas.



Advantages of using Service Fabric

Following are the primary advantages for using Service Fabric for Network Controller clusters.

High Availability and Scalability

Because Network Controller is the core of a datacenter network, it must both be resilient to failure and be scalable enough to allow agile changes in datacenter networks over time. The following features provide these abilities:

- **Fast failover.** Service Fabric provides extremely fast failover. Multiple hot secondary service replicas are always available. If an operating system instance becomes unavailable due to hardware failure, one of the secondary replicas is immediately promoted to primary replica.
- **Agility of scale.** You can easily and quickly scale these reliable services from a few instances up to thousands of instances and then back down to a few instances, depending on your resource needs.

Persistent storage

The Network Controller application has large storage requirements for its configuration and state. The application also must be usable across planned and unplanned outages. For this purpose, Service Fabric provides a Key-Value Store (KVS) that is a replicated, transactional and persisted store.

Modularity

Network Controller is designed with a modular architecture, with each of the network services, such as the virtual networks service and firewall service, built-in as individual services.

This application architecture provides the following benefits.

1. Network Controller modularity allows independent development of each of the supported services, as needs evolve. For example, the Software Load Balancing service can be updated without affecting any of the other services or the normal operation of Network Controller.
2. Network Controller modularity allows the addition of new services, as the network evolves. New services can be added to Network Controller without impacting existing services.

NOTE

In Windows Server 2016, the addition of third party services to Network Controller is not supported.

Service Fabric modularity uses service model schemas to maximize the ease of developing, deploying, and servicing an application.

Network Controller Deployment Options

To deploy Network Controller by using System Center Virtual Machine Manager (VMM), see [Set up an SDN network controller in the VMM fabric](#).

To deploy Network Controller using scripts, see [Deploy a Software Defined Network Infrastructure Using Scripts](#).

To deploy Network Controller using Windows PowerShell, see [Deploy Network Controller using Windows PowerShell](#)

For more information about Network Controller, see [Network Controller](#).

Install the Network Controller Server Role Using Server Manager

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

This topic provides instructions on how to install the Network Controller server role by using Server Manager.

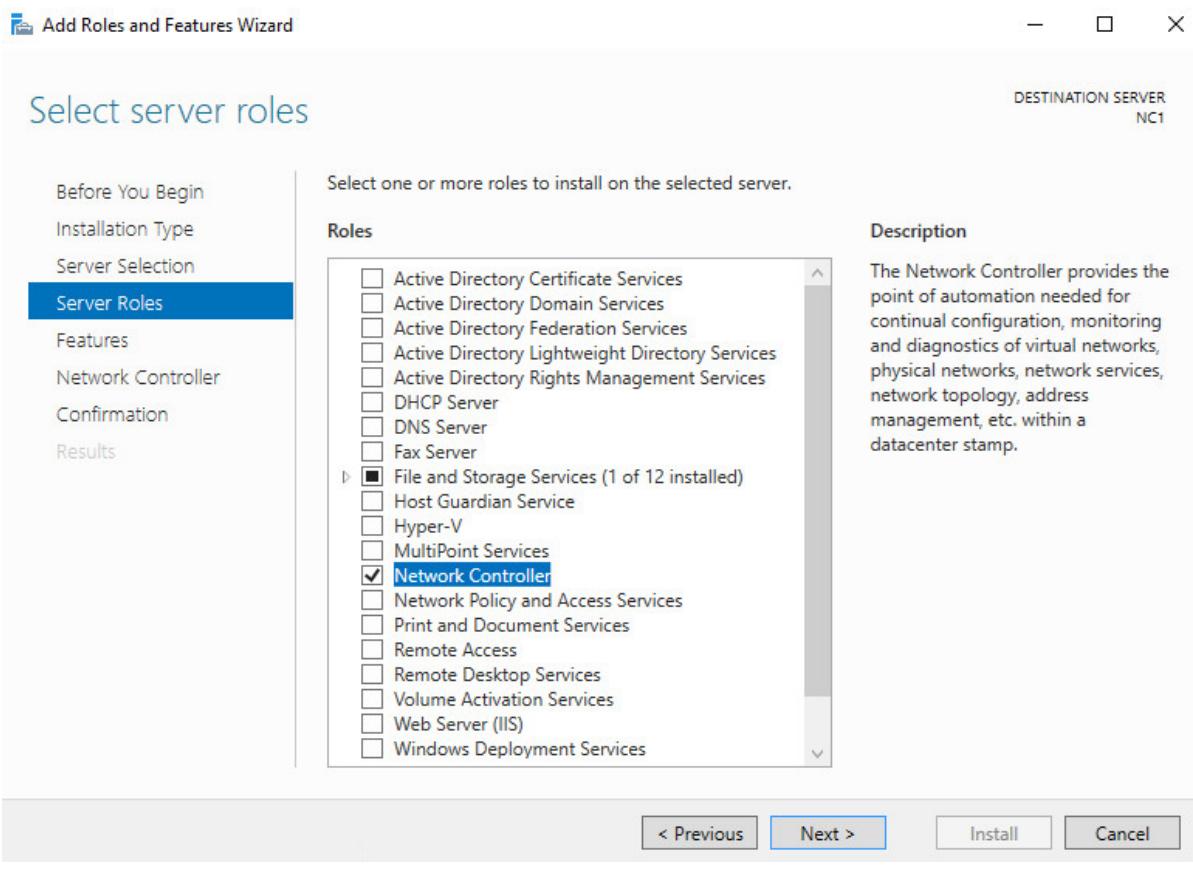
IMPORTANT

Do not deploy the Network Controller server role on physical hosts. To deploy Network Controller, you must install the Network Controller server role on a Hyper-V virtual machine (VM) that is installed on a Hyper-V host. After you have installed Network Controller on VMs on three different Hyper-V hosts, you must enable the Hyper-V hosts for Software Defined Networking (SDN) by adding the hosts to Network Controller using the Windows PowerShell command **New-NetworkControllerServer**. By doing so, you are enabling the SDN Software Load Balancer to function. For more information, see [New-NetworkControllerServer](#).

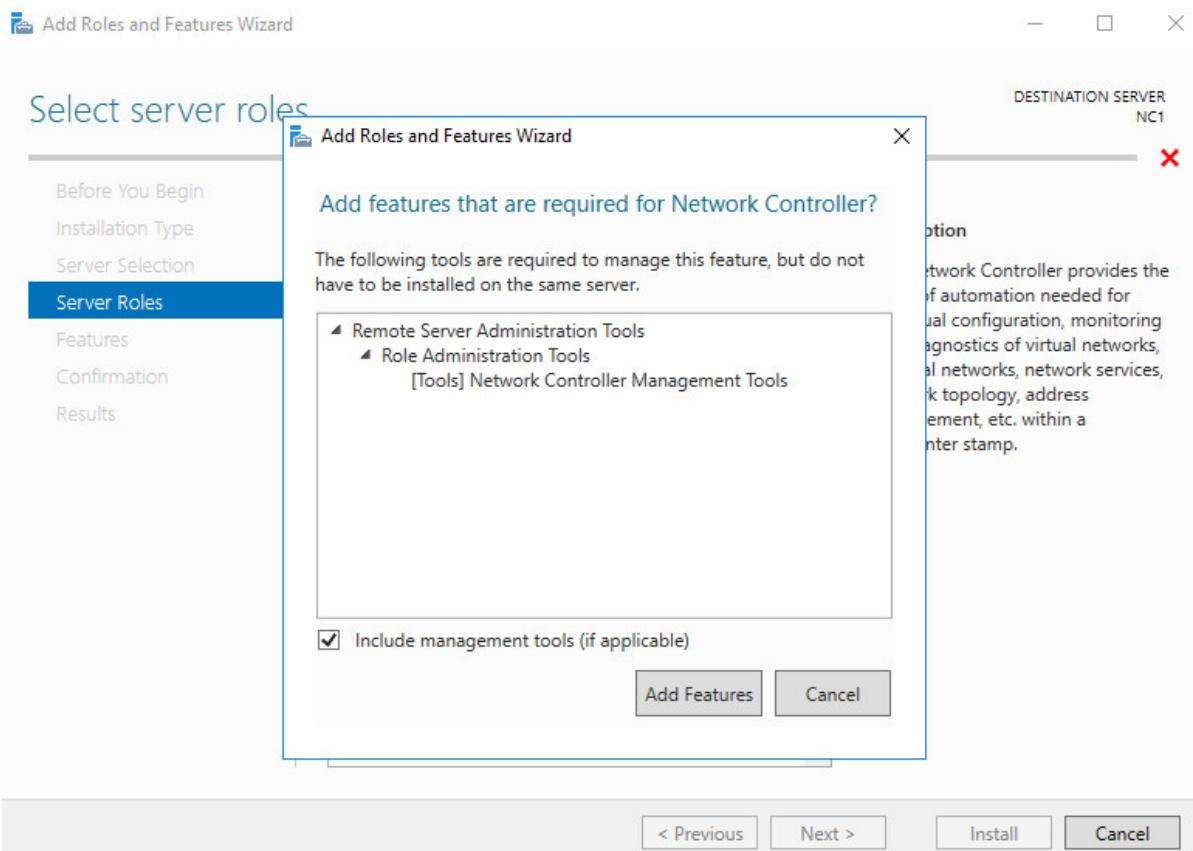
After you install Network Controller, you must use Windows PowerShell commands for additional Network Controller configuration. For more information, see [Deploy Network Controller using Windows PowerShell](#).

To install Network Controller

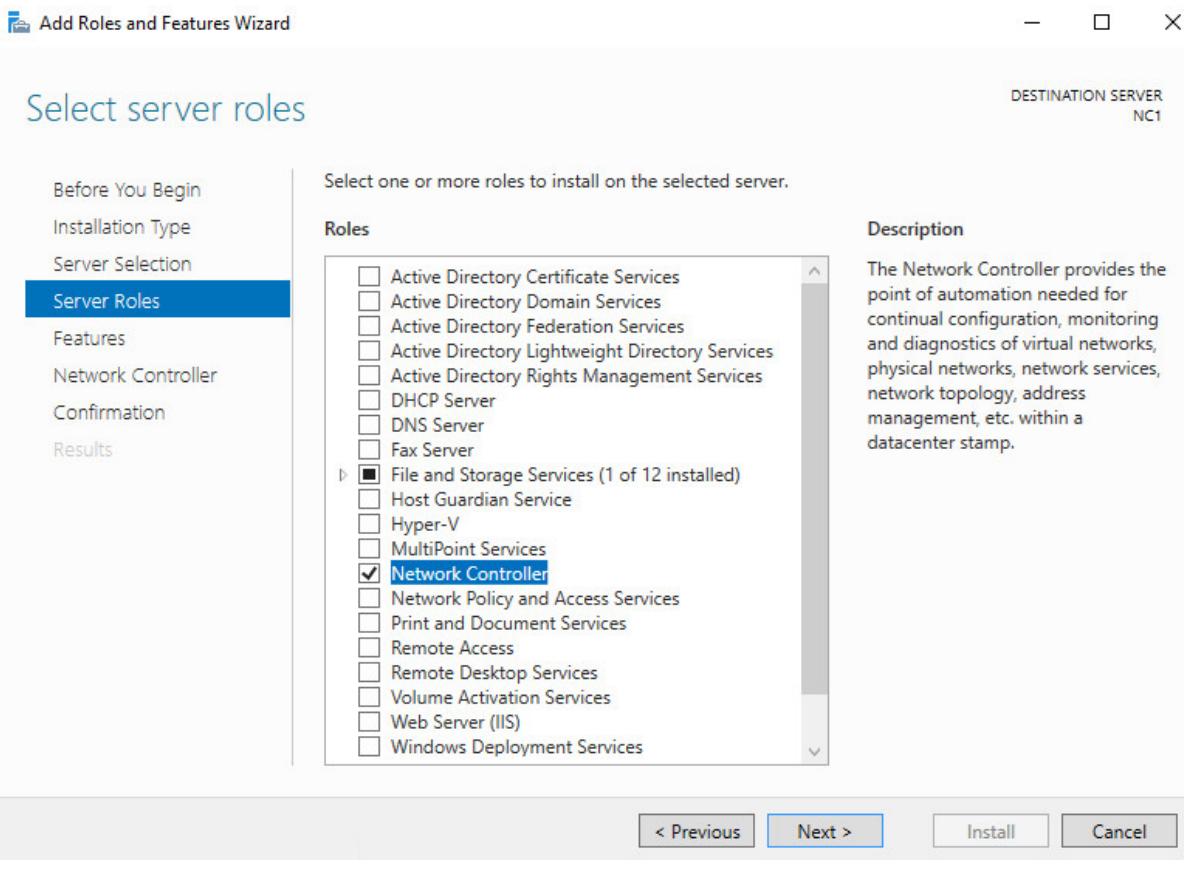
1. In Server Manager, click **Manage**, and then click **Add Roles and Features**. The Add Roles and Features wizard opens. Click **Next**.
2. In **Select Installation Type**, keep the default setting and click **Next**.
3. In **Select Destination Server**, choose the server where you want to install Network Controller, and then click **Next**.
4. In **Select Server Roles**, in **Roles**, click **Network Controller**.



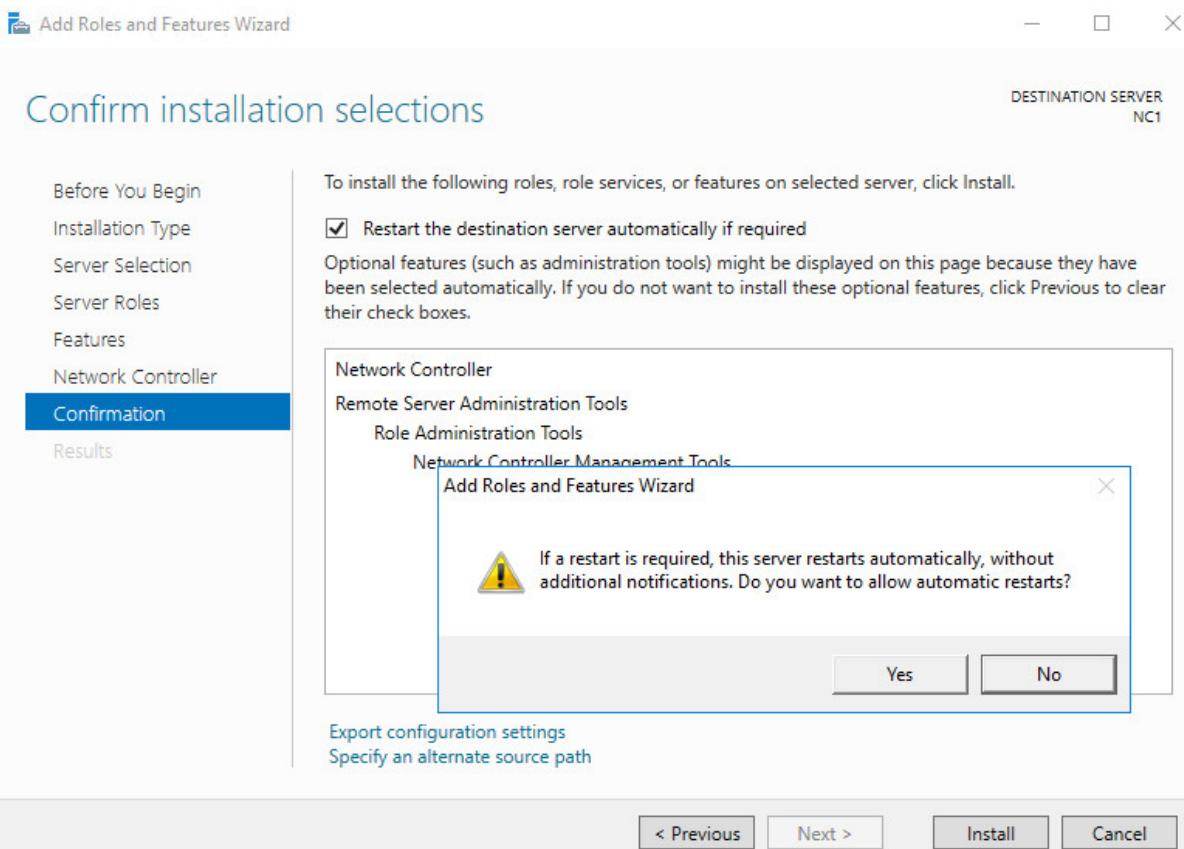
5. The **Add features that are required for Network Controller** dialog box opens. Click **Add Features**.



6. In **Select Server Roles**, click **Next**.

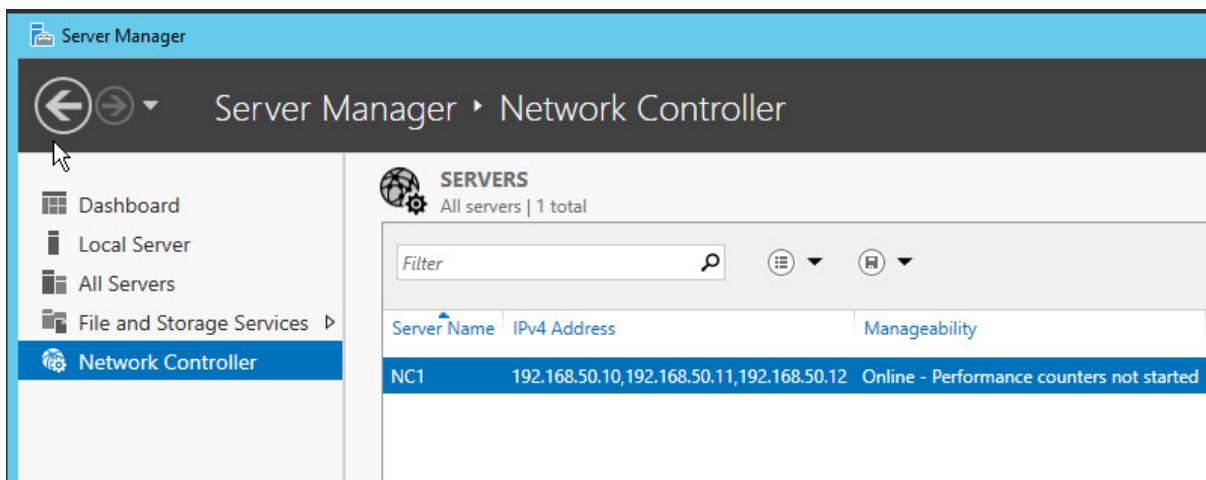


7. In **Select Features**, click **Next**.
8. In **Network Controller** click **Next**.
9. In **Confirm installation selections**, review your choices. Installation of Network Controller requires that you restart the computer after the wizard runs. Because of this, click **Restart the destination server automatically if required**. The **Add Roles and Features Wizard** dialog box opens. Click **Yes**.



10. In **Confirm installation selections**, click **Install**.

11. The Network Controller server role installs on the destination server, and then the server restarts.
12. After the computer restarts, log on to the computer and verify Network Controller installation by viewing Server Manager.



See Also

[Network Controller](#)

Post-Deployment Steps for Network Controller

3/23/2018 • 2 minutes to read • [Edit Online](#)

When you install Network Controller, you can choose Kerberos or non-Kerberos deployments.

For non-Kerberos deployments, you must configure certificates.

Configure certificates for non-Kerberos deployments

If the computers or virtual machines (VMs) for Network Controller and the management client are not domain-joined, you must configure certificate-based authentication by completing the following steps.

- Create a certificate on the Network Controller for Computer authentication. The certificate subject name must be same as the DNS name of the Network Controller computer or VM.
- Create a certificate on the management client. This certificate must be trusted by the Network Controller.
- Enroll a certificate on the Network Controller computer or VM. The certificate must meet the following requirements.
 - Both the Server Authentication purpose and the Client Authentication purpose must be configured in Enhanced Key Usage (EKU) or Application Policies extensions. The object identifier for Server Authentication is 1.3.6.1.5.5.7.3.1. The object identifier for Client Authentication is 1.3.6.1.5.5.7.3.2.
 - The certificate subject name should resolve to:
 - The IP address of the Network Controller computer or VM, if Network Controller is deployed on a single computer or VM.
 - The REST IP address, if Network Controller is deployed on multiple computers, multiple VMs, or both.
 - This certificate must be trusted by all the REST clients. The certificate must also be trusted by the Software Load Balancing (SLB) Multiplexer (MUX) and the southbound host computers that are managed by Network Controller.
 - The certificate can be enrolled by a Certification Authority (CA) or can be a self-signed certificate. Self-signed certificates are not recommended for production deployments, but are acceptable for test lab environments.
 - The same certificate must be provisioned on all the Network Controller nodes. After creating the certificate on one node, you can export the certificate (with private key) and import it on the other nodes.

For more information, see [Network Controller](#).

Network Function Virtualization

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

You can use this topic to learn about Network Function Virtualization, which allows you to deploy virtual networking appliances such as Datacenter Firewall, multitenant RAS Gateway, and Software Load Balancing (SLB) multiplexer (MUX).

NOTE

In addition to this topic, the following Network Function Virtualization documentation is available.

- [Datacenter Firewall Overview](#)
- [RAS Gateway for SDN](#)
- [Software Load Balancing \(SLB\) for SDN](#)

In today's software defined datacenters, network functions that are being performed by hardware appliances (such as load balancers, firewalls, routers, switches, and so on) are increasingly being virtualized as virtual appliances. This "network function virtualization" is a natural progression of server virtualization and network virtualization. Virtual appliances are quickly emerging and creating a brand new market. They continue to generate interest and gain momentum in both virtualization platforms and cloud services.

Microsoft included a standalone gateway as a virtual appliance starting with Windows Server 2012 R2 . For more information, see [Windows Server Gateway](#). Now with Windows Server 2016 Microsoft continues to expand and invest in the network function virtualization market.

Virtual appliance benefits

A virtual appliance is dynamic and easy to change because it is a pre-built, customized virtual machine. It can be one or more virtual machines packaged, updated, and maintained as a unit. Together with software defined networking (SDN), you get the agility and flexibility needed in today's cloud-based infrastructure. For example:

- SDN presents the network as a pooled and dynamic resource.
- SDN facilitates tenant isolation.
- SDN maximizes scale and performance.
- Virtual appliances enable seamless capacity expansion and workload mobility.
- Virtual appliances minimize operational complexity.
- Virtual appliances let customers easily acquire, deploy, and manage pre-integrated solutions.
 - Customers can easily move the virtual appliance anywhere in the cloud.
 - Customers can scale virtual appliances up or down dynamically based on demand.

For more information about Microsoft SDN see [Software Defined Networking](#).

What network functions are being virtualized?

The marketplace for virtualized network functions is growing quickly. The following network functions are being

virtualized:

- **Security**

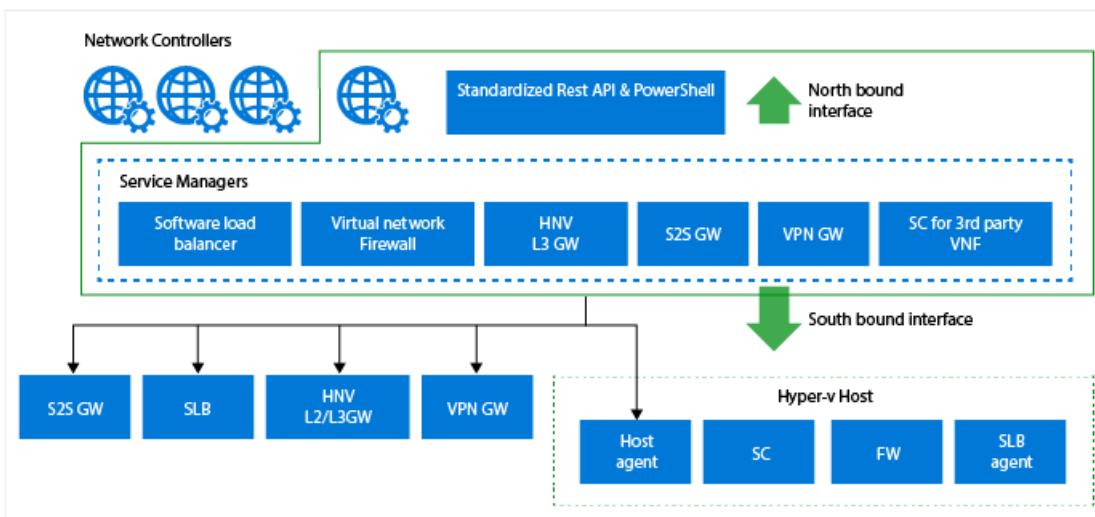
- Firewall
- Antivirus
- DDoS (Distributed Denial of Service)
- IPS/IDS (Intrusion Prevention System/Intrusion Detection System)

- **Application/WAN optimizers**

- **Edge**

- Site-to-site gateway
- L3 gateways
- Routers
- Switches
- NAT
- Load balancers (not necessarily at the edge)
- HTTP proxy

Why Microsoft is a great platform for virtual appliances



The Microsoft platform has been engineered to be a great platform to build and deploy virtual appliances. Here's why:

- Microsoft provides key virtualized network functions with Windows Server 2016.
- You can deploy a virtual appliance from the vendor of your choice.
- You can deploy, configure, and manage your virtual appliances with the Microsoft Network Controller which comes with Windows Server 2016. For more information about the Network Controller, see [Network Controller](#).
- Hyper-V can host the top guest operating systems that you need.

Network function virtualization in Windows Server 2016

Virtual appliances functions provided by Microsoft

The following virtual appliances are provided with Windows Server 2016:

Software load balancer

A layer-4 load balancer operating at datacenter scale. This is a similar version of Azure's load balancer that has been deployed at scale in the Azure environment. For more information about the Microsoft Software Load Balancer, see [Software Load Balancing \(SLB\) for SDN](#). For more information about Microsoft Azure Load Balancing Services, see [Microsoft Azure Load Balancing Services](#).

Gateway. RAS Gateway provides all combinations of the following gateway functions.

- **Site-to-Site gateway**

RAS Gateway provides a Border Gateway Protocol (BGP)-capable, multitenant gateway that allows your tenants to access and manage their resources over site-to-site VPN connections from remote sites, and that allows network traffic flow between virtual resources in the cloud and tenant physical networks. For more information about the RAS Gateway, see [RAS Gateway High Availability](#) and [RAS Gateway](#).

- **Forwarding gateway**

RAS Gateway routes traffic between virtual networks and the hosting provider physical network. For example, if tenants create one or more virtual networks, and need access to shared resources on the physical network at the hosting provider, the forwarding gateway can route traffic between the virtual network and the physical network to provide users working on the virtual network with the services that they need. For more information, see [RAS Gateway High Availability](#) and [RAS Gateway](#).

- **GRE tunnel gateways**

GRE based tunnels enable connectivity between tenant virtual networks and external networks. Since the GRE protocol is lightweight and support for GRE is available on most network devices, it becomes an ideal choice for tunneling where data encryption is not required. GRE support in Site to Site (S2S) tunnels solves the problem of forwarding between tenant virtual networks and tenant external networks using a multi-tenant gateway. For more information about GRE tunnels, see [GRE Tunneling in Windows Server 2016](#).

Routing control plane with BGP

Hyper-V Network Virtualization (HNV) Routing Control is the logical, centralized entity in the control plane, which carries all the Customer Address plane routes and dynamically learns and then updates the distributed RAS Gateway routers in the virtual network. For more information, see [RAS Gateway High Availability](#) and [RAS Gateway](#).

Distributed multi-tenant firewall

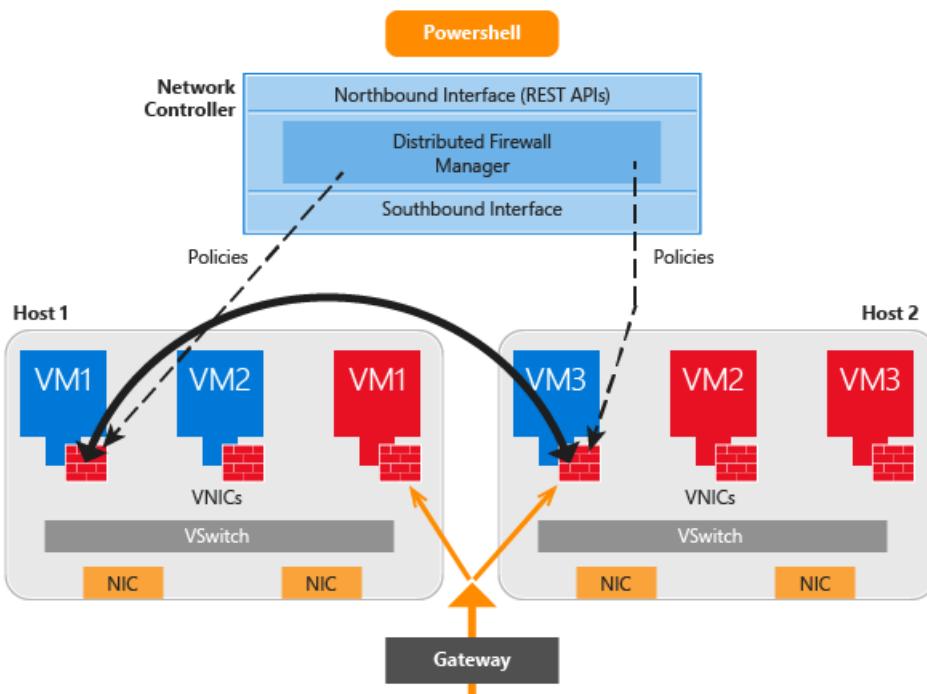
The firewall protects the network layer of virtual networks. The policies are enforced at the SDN-vSwitch port of each tenant VM. It protects all traffic flows: east-west and north-south. The policies are pushed through the tenant portal and the Network Controller distributes them to all applicable hosts. For more information about the distributed multi-tenant firewall, see [Datacenter Firewall Overview](#).

Datacenter Firewall Overview

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

Datacenter Firewall is a new service included with Windows Server 2016. It is a network layer, 5-tuple (protocol, source and destination port numbers, source and destination IP addresses), stateful, multitenant firewall. When deployed and offered as a service by the service provider, tenant administrators can install and configure firewall policies to help protect their virtual networks from unwanted traffic originating from Internet and intranet networks.



The service provider administrator or the tenant administrator can manage the Datacenter Firewall policies via the network controller and the northbound APIs.

The Datacenter Firewall offers the following advantages for cloud service providers:

- A highly scalable, manageable, and diagnosable software-based firewall solution that can be offered to tenants
- Freedom to move tenant virtual machines to different compute hosts without breaking tenant firewall policies
 - Deployed as a vSwitch port host agent firewall
 - Tenant virtual machines get the policies assigned to their vSwitch host agent firewall
 - Firewall rules are configured in each vSwitch port, independent of the actual host running the virtual machine
- Offers protection to tenant virtual machines independent of the tenant guest operating system

The Datacenter Firewall offers the following advantages for tenants:

- Ability to define firewall rules to help protect Internet facing workloads on virtual networks

- Ability to define firewall rules to help protect traffic between virtual machines on the same L2 virtual subnet as well as between virtual machines on different L2 virtual subnets
- Ability to define firewall rules to help protect and isolate network traffic between tenant on-premises networks and their virtual networks at the service provider

RAS Gateway for SDN

9/1/2018 • 3 minutes to read • [Edit Online](#)

Applies to: Windows Server (Semi-Annual Channel), Windows Server 2016## RAS Gateway for SDN

RAS Gateway is a software-based, multitenant, Border Gateway Protocol (BGP) capable router designed for Cloud Service Providers (CSPs) and Enterprises that host multiple tenant virtual networks using Hyper-V Network Virtualization. RAS Gateways routes network traffic between the physical network and VM network resources, regardless of the location. You can route the network traffic at the same physical location or many different locations.

Multitenancy is the ability of a cloud infrastructure to support the virtual machine workloads of multiple tenants, yet isolate them from each other, while all of the workloads run on the same infrastructure. The multiple workloads of an individual tenant can interconnect and be managed remotely, but these systems do not interconnect with the workloads of other tenants, nor can other tenants remotely manage them.

NOTE

In addition to this topic, the following RAS Gateway topics are available.

- [What's New in RAS Gateway](#)
- [RAS Gateway Deployment Architecture](#)
- [RAS Gateway High Availability](#)
- [Border Gateway Protocol \(BGP\)](#)
- [BGP Windows PowerShell Command Reference](#)

Prerequisites for installing RAS Gateway for SDN

You cannot use the Windows interface to install Remote Access when you want to deploy RAS Gateway in multitenant mode for use with SDN. Instead, you must use Windows PowerShell.

But before you can install RAS Gateway by using Windows PowerShell, you must use Windows PowerShell to add the **RemoteAccess** Windows feature. To do so, run the following command at the Windows PowerShell prompt.

```
Add-WindowsFeature -Name RemoteAccess -IncludeAllSubFeature -IncludeManagementTools
```

This command adds the **RemoteAccess** feature and the Windows PowerShell commands for the feature.

After you have added **RemoteAccess** to your server, you can install Remote Access as a RAS Gateway with multitenant mode and Border Gateway Protocol (BGP).

For more information, see the Windows PowerShell reference topic [Install-RemoteAccess](#).

RAS Gateway Features

Following are RAS Gateway features in Windows Server 2016. You can deploy RAS Gateway in high availability pools that use all of these features at one time.

- **Site-to-site VPN.** This RAS Gateway feature allows you to connect two networks at different physical locations across the Internet by using a site-to-site VPN connection. For CSPs that host many tenants in their datacenter, RAS Gateway provides a multitenant gateway solution that allows your tenants to access and manage their resources over site-to-site VPN connections from remote sites, and that allows network

traffic flow between virtual resources in your datacenter and their physical network.

- **Point-to-site VPN.** This RAS Gateway feature allows organization employees or administrators to connect to your organization's network from remote locations. For multitenant deployments, tenant network administrators can use point-to-site VPN connections to access virtual network resources at the CSP datacenter.
- **GRE Tunneling.** Generic Routing Encapsulation (GRE) based tunnels enable connectivity between tenant virtual networks and external networks. Since the GRE protocol is lightweight and support for GRE is available on most of network devices it becomes an ideal choice for tunneling where encryption of data is not required. GRE support in Site to Site (S2S) tunnels solves the problem of forwarding between tenant virtual networks and tenant external networks using a multi-tenant gateway, as described later in this topic.
- **Dynamic routing with Border Gateway Protocol (BGP).** BGP reduces the need for manual route configuration on routers because it is a dynamic routing protocol, and automatically learns routes between sites that are connected by using site-to-site VPN connections. If your organization has multiple sites that are connected by using BGP-enabled routers such as RAS Gateway, BGP allows the routers to automatically calculate and use valid routes to each other in the event of network disruption or failure. For more information, see [RFC 4271](#).

What's New in RAS Gateway

9/1/2018 • 5 minutes to read • [Edit Online](#)

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

You can use this topic to learn about new features for RAS Gateway, which is a software-based, multitenant, Border Gateway Protocol (BGP) capable router in Windows Server 2016. The RAS Gateway Multitenant BGP router is designed for Cloud Service Providers (CSPs) and Enterprises that host multiple tenant virtual networks using Hyper-V Network Virtualization.

NOTE

In Windows Server 2012 R2, RAS Gateway is named RRAS Gateway; and in System Center Virtual Machine Manager, RAS Gateway is named Windows Server Gateway.

This topic contains the following sections.

- [Site-to-site Connectivity Options](#)
- [Gateway Pools](#)
- [Gateway Pool Scalability](#)
- [M+N Gateway Pool Redundancy](#)
- [Route Reflector](#)

Site-to-site Connectivity Options

RAS Gateway now supports three types of VPN site-to-site connections: Internet Key Exchange version 2 (IKEv2) site-to-site virtual private networking (VPN), Layer 3 (L3) VPN, and Generic Routing Encapsulation (GRE) tunnels.

For more information about GRE, see [GRE Tunneling in Windows Server 2016](#).

Gateway Pools

In Windows Server 2016, you can create gateway pools of different types. Gateway pools contain many instances of RAS Gateway, and route network traffic between physical and virtual networks. Gateway pools can perform any of the individual gateway functions - Internet Key Exchange version 2 (IKEv2) site-to-site virtual private networking (VPN), Layer 3 (L3) VPN, and Generic Routing Encapsulation (GRE) tunnels - or the pool can perform all of these functions and act as a mixed pool.

You can create gateway pools using any logic that you prefer based on your infrastructure requirements. For example, you can create gateway pools based on any of the following characteristics.

- Tunnel types (IKEv2 VPN, L3 VPN, GRE VPN)
- Capacity
- Redundancy level (reliability based on your billing plan for tenants)
- Customized separation for customers

For more information, see [RAS Gateway High Availability](#).

Gateway Pool Scalability

You can easily scale a gateway pool up or down by adding or removing gateway VMs in the pool. Removal or addition of gateways does not disrupt the services that are provided by a pool. You can also add and remove entire pools of gateways.

For more information, see [RAS Gateway High Availability](#).

M+N Gateway Pool Redundancy

Every gateway pool is M+N redundant. This means that an 'M' number of active gateway virtual machines (VMs) are backed up by an 'N' number of standby gateway VMs. M+N redundancy provides you with more flexibility in determining the level of reliability that you require when you deploy RAS Gateway. Rather than using only one standby RAS Gateway per active RAS Gateway VM - which is the only configuration option with Windows Server 2012 R2 - you can now configure as many standby VMs as you require. The Network Controller Gateway Service Manager feature efficiently uses the standby RAS Gateway VM capacity to provide reliable failover if an active RAS Gateway VM fails or loses connectivity.

For more information, see [RAS Gateway High Availability](#).

Route Reflector

The Border Gateway Protocol (BGP) Route Reflector is now included with RAS Gateway, and provides an alternative to BGP full mesh topology that is required for route synchronization between routers. With full mesh synchronization, all BGP routers must connect with all other routers in the routing topology. When you use Route Reflector, however, the Route Reflector is the only router that connects with all of the other routers, called BGP clients, thereby simplifying route synchronization and reducing network traffic. The Route Reflector learns all routes, calculates best routes, and redistributes the best routes to its BGP clients.

With Windows Server 2016, you can configure an individual tenant's remote access tunnels to terminate on more than one RAS Gateway VM. This provides increased flexibility for Cloud Service Providers when faced with circumstances where one RAS Gateway VM cannot meet all of the bandwidth requirements of the tenant connections.

This capability, however, introduces the additional complexity of route management and effective synchronization of routes between the tenant remote sites and their virtual resources in the cloud datacenter. Providing tenants with connections to multiple RAS Gateways also introduces additional complexity in configuration at the Enterprise end, where each tenant site will have separate routing neighbors.

A BGP Route Reflector in the control plane addresses these problems and makes the CSP internal fabric deployment transparent to the Enterprise tenants. Following are some key points about the BGP Route Reflector that is included with RAS Gateway and integrated with Network Controller.

- A Route Reflector in a Software Defined Networking deployment is a logical entity that sits on the control plane between the RAS Gateways and the Network Controller. It does not, however, participate in data plane routing.
- When you add a new tenant to your datacenter, Network Controller automatically configures the first tenant RAS Gateway as a Route Reflector.
- Each tenant has a corresponding Route Reflector, and it resides on one of the RAS Gateway VMs that are associated with that tenant.
- A tenant Route Reflector acts as the Route Reflector for all of the RAS Gateway VMs that are associated with the tenant. Tenant gateways other than the RAS Gateway Route Reflector are the Route Reflector Clients. The Route Reflector performs route synchronization between all Route Reflector Clients so that the actual

data path routing can occur.

- A Route Reflector does not provide route reflector services for the RAS Gateway upon which it is configured.
- A Route Reflector updates Network Controller with the Enterprise routes that correspond to the tenant's Enterprise sites. This allows Network Controller to configure the required Hyper-V Network Virtualization policies on the tenant virtual network for End-to-End Data Path access.
- If your Enterprise customers use BGP Routing in the Customer Address space, the RAS Gateway Route Reflector is the only external BGP (eBGP) neighbor for all of the sites of the corresponding tenant. This is true regardless of the Enterprise tenant's tunnel termination points. In other words, no matter which RAS Gateway VM in the CSP datacenter terminates the site-to-site VPN tunnel for a tenant site, the eBGP Peer for all the tenant sites is the Route Reflector.

For more information, see [RAS Gateway Deployment Architecture](#) and the Internet Engineering Task Force (IETF) Request for Comments topic [RFC 4456 BGP Route Reflection: An Alternative to Full Mesh Internal BGP \(IBGP\)](#).

RAS Gateway Deployment Architecture

9/1/2018 • 10 minutes to read • [Edit Online](#)

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

You can use this topic to learn about Cloud Service Provider (CSP) deployment of RAS Gateway, including RAS Gateway pools, Route Reflectors, and deploying multiple gateways for individual tenants.

The following sections provide brief overviews of some of the RAS Gateway new features so that you can understand how to use these features in the design of your gateway deployment.

In addition, an example deployment is provided, including information about the process of adding new tenants, route synchronization and data plane routing, gateway and Route Reflector failover, and more.

This topic contains the following sections.

- [Using RAS Gateway New Features to Design Your Deployment](#)
- [Example Deployment](#)
- [Adding New Tenants and Customer Address \(CA\) Space EBGP Peering](#)
- [Route Synchronization and Data Plane Routing](#)
- [How Network Controller Responds to RAS Gateway and Route Reflector Failover](#)
- [Advantages of Using New RAS Gateway Features](#)

Using RAS Gateway New Features to Design Your Deployment

RAS Gateway includes multiple new features that change and improve the way in which you deploy your gateway infrastructure in your datacenter.

BGP Route Reflector

The Border Gateway Protocol (BGP) Route Reflector capability is now included with RAS Gateway, and provides an alternative to BGP full mesh topology that is normally required for route synchronization between routers. With full mesh synchronization, all BGP routers must connect with all other routers in the routing topology. When you use Route Reflector, however, the Route Reflector is the only router that connects with all of the other routers, called BGP Route Reflector clients, thereby simplifying route synchronization and reducing network traffic. The Route Reflector learns all routes, calculates best routes, and redistributes the best routes to its BGP clients.

For more information, see [What's New in RAS Gateway](#).

Gateway Pools

In Windows Server 2016, you can create many gateway pools of different types. Gateway pools contain many instances of RAS Gateway, and route network traffic between physical and virtual networks.

For more information, see [What's New in RAS Gateway](#) and [RAS Gateway High Availability](#).

Gateway Pool Scalability

You can easily scale a gateway pool up or down by adding or removing gateway VMs in the pool. Removal or addition of gateways does not disrupt the services that are provided by a pool. You can also add and remove entire pools of gateways.

For more information, see [What's New in RAS Gateway](#) and [RAS Gateway High Availability](#).

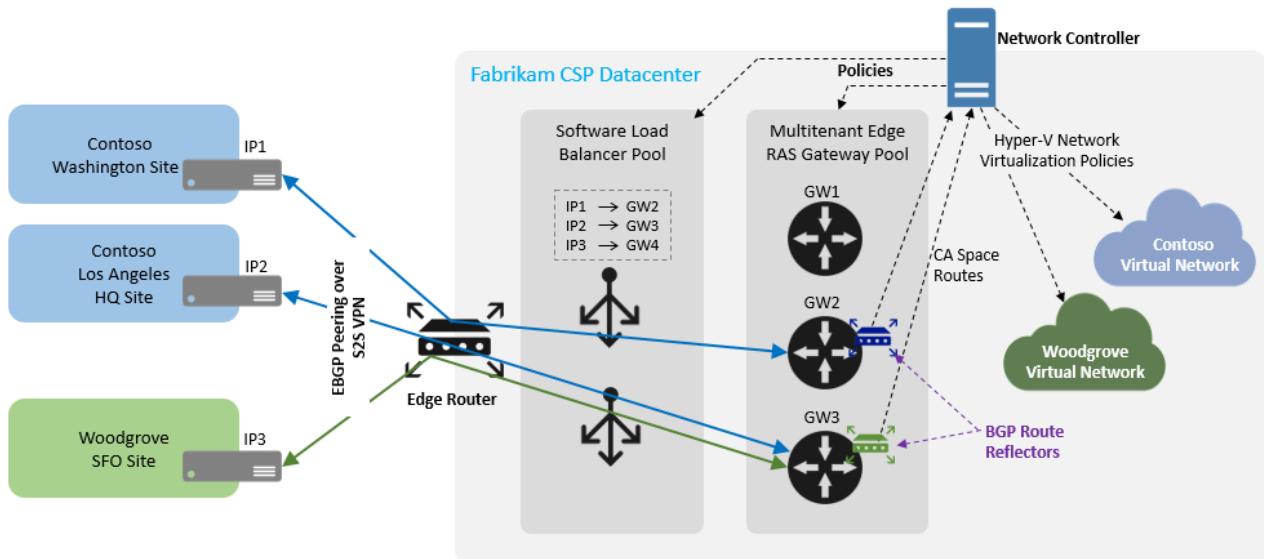
M+N Gateway Pool Redundancy

Every gateway pool is M+N redundant. This means that an 'M' number of active gateway VMs are backed up by an 'N' number of standby gateway VMs. M+N redundancy provides you with more flexibility in determining the level of reliability that you require when you deploy RAS Gateway.

For more information, see [What's New in RAS Gateway](#) and [RAS Gateway High Availability](#).

Example Deployment

The following illustration provides an example with eBGP peering over site-to-site VPN connections configured between two tenants, Contoso and Woodgrove, and the Fabrikam CSP datacenter.



In this example, Contoso requires additional gateway bandwidth, leading to the gateway infrastructure design decision to terminate the Contoso Los Angeles site on GW3 instead of GW2. Because of this, Contoso VPN connections from different sites terminate in the CSP datacenter on two different gateways.

Both of these gateways, GW2 and GW3, were the first RAS Gateways configured by Network Controller when the CSP added the Contoso and Woodgrove tenants to their infrastructure. Because of this, these two gateways are configured as Route Reflectors for these corresponding customers (or tenants). GW2 is the Contoso Route Reflector, and GW3 is the Woodgrove Route Reflector - in addition to being the CSP RAS Gateway termination point for the VPN connection with the Contoso Los Angeles HQ site.

NOTE

One RAS Gateway can route virtual and physical network traffic for up to one hundred different tenants, depending on the bandwidth requirements of each tenant.

As Route Reflectors, GW2 sends Contoso CA Space routes to Network Controller, and GW3 sends Woodgrove CA Space routes to Network Controller.

Network Controller pushes Hyper-V Network Virtualization policies to the Contoso and Woodgrove virtual networks, as well as RAS policies to the RAS Gateways and load balancing policies to the Multiplexers (MUXes) that are configured as a Software Load Balancing pool.

Adding New Tenants and Customer Address (CA) Space eBGP Peering

When you sign a new customer and add the customer as a new tenant in your datacenter, you can use the following

process, much of which is automatically performed by Network Controller and RAS Gateway eBGP routers.

1. Provision a new virtual network and workloads according to your tenant's requirements.
2. If required, configure remote connectivity between the remote tenant Enterprise site and their virtual network at your datacenter. When you deploy a site-to-site VPN connection for the tenant, Network Controller automatically selects an available RAS Gateway VM from the available gateway pool and configures the connection.
3. While configuring the RAS Gateway VM for the new tenant, Network Controller also configures the RAS Gateway as a BGP Router and designates it as the Route Reflector for the tenant. This is true even in circumstances where the RAS Gateway serves as a gateway, or as a gateway and Route Reflector, for other tenants.
4. Depending on whether CA space routing is configured to use statically configured networks or dynamic BGP routing, Network Controller configures the corresponding static routes, BGP neighbors, or both on the RAS Gateway VM and Route Reflector.

NOTE

- After Network Controller has configured a RAS Gateway and Route Reflector for the tenant, whenever the same tenant requires a new site-to-site VPN connection, Network Controller checks for the available capacity on this RAS Gateway VM. If the original gateway can service the required capacity, the new network connection is also configured on the same RAS Gateway VM. If the RAS Gateway VM cannot handle additional capacity, Network Controller selects a new available RAS Gateway VM and configures the new connection on it. This new RAS Gateway VM associated with the tenant becomes the Route Reflector client of the original tenant RAS Gateway Route Reflector.
- Because RAS Gateway pools are behind Software Load Balancers (SLBs), the tenants' site-to-site VPN addresses each use a single public IP address, called a virtual IP address (VIP), which is translated by the SLBs into a datacenter-internal IP address, called a dynamic IP address (DIP), for a RAS Gateway that routes traffic for the Enterprise tenant. This public-to-private IP address mapping by SLB ensures that the site-to-site VPN tunnels are correctly established between the Enterprise sites and the CSP RAS Gateways and Route Reflectors.

For more information about SLB, VIPs, and DIPs, see [Software Load Balancing \(SLB\) for SDN](#).

5. After the site-to-site VPN tunnel between the Enterprise site and the CSP datacenter RAS Gateway is established for the new tenant, the static routes that are associated with the tunnels are automatically provisioned on both the Enterprise and CSP sides of the tunnel.
6. With CA space BGP routing, the eBGP peering between the Enterprise sites and the CSP RAS Gateway Route Reflector is also established.

Route Synchronization and Data Plane Routing

After eBGP peering is established between Enterprise sites and the CSP RAS Gateway Route Reflector, the Route Reflector learns all of the Enterprise routes by using dynamic BGP routing. The Route Reflector synchronizes these routes between all of the Route Reflector clients so that they are all configured with the same set of routes.

Route Reflector also updates these consolidated routes, using route synchronization, to Network Controller. Network Controller then translates the routes into the Hyper-V Network Virtualization policies and configures the Fabric Network to ensure that End-to-End Data Path routing is provisioned. This process makes the tenant virtual network accessible from the tenant Enterprise sites.

For Data Plane routing, the packets that reach the RAS Gateway VMs are directly routed to the tenant's virtual network, because the required routes are now available with all of the participating RAS Gateway VMs.

Similarly, with the Hyper-V Network Virtualization policies in place, the tenant virtual network routes packets

directly to the RAS Gateway VMs (without requiring to know about the Route Reflector) and then to the Enterprise sites over the site-to-site VPN tunnels.

In addition, return traffic from the tenant virtual network to the remote tenant Enterprise site bypasses the SLBs, a process called Direct Server Return (DSR).

How Network Controller Responds to RAS Gateway and Route Reflector Failover

Following are two possible failover scenarios - one for RAS Gateway Route Reflector clients and one for RAS Gateway Route Reflectors - including information about how Network Controller handles failover for VMs in either configuration.

VM Failure of a RAS Gateway BGP Route Reflector Client

Network Controller takes the following actions when a RAS Gateway Route Reflector client fails.

NOTE

When a RAS Gateway is not a Route Reflector for a tenant's BGP infrastructure, it is a Route Reflector client in the tenant's BGP infrastructure.

- Network Controller selects an available standby RAS Gateway VM and provisions the new RAS Gateway VM with the configuration of the failed RAS Gateway VM.
- Network Controller updates the corresponding SLB configuration to ensure that the site-to-site VPN tunnels from tenant sites to the failed RAS Gateway are correctly established with the new RAS Gateway.
- Network Controller configures the BGP Route Reflector client on the new gateway.
- Network Controller configures the new RAS Gateway BGP Route Reflector client as active. The RAS Gateway immediately starts peering with the tenant's Route Reflector to share routing information and to enable eBGP peering for the corresponding Enterprise site.

VM Failure for a RAS Gateway BGP Route Reflector

Network Controller takes the following actions when a RAS Gateway BGP Route Reflector fails.

- Network Controller selects an available standby RAS Gateway VM and provisions the new RAS Gateway VM with the configuration of the failed RAS Gateway VM.
- Network Controller configures the Route Reflector on the new RAS Gateway VM, and assigns the new VM the same IP address that was used by the failed VM, thereby providing route integrity despite the VM failure.
- Network Controller updates the corresponding SLB configuration to ensure that the site-to-site VPN tunnels from tenant sites to the failed RAS Gateway are correctly established with the new RAS Gateway.
- Network Controller configures the new RAS Gateway BGP Route Reflector VM as active.
- The Route Reflector immediately becomes active. The site-to-site VPN tunnel to the Enterprise is established, and the Route Reflector uses eBGP peering and exchanges routes with the Enterprise site routers.
- After BGP route selection, the RAS Gateway BGP Route Reflector updates tenant Route Reflector clients in the datacenter, and synchronizes routes with Network Controller, making the End-to-End Data Path available for tenant traffic.

Advantages of Using New RAS Gateway Features

Following are a few of the advantages of using these new RAS Gateway features when designing your RAS Gateway deployment.

RAS Gateway scalability

Because you can add as many RAS Gateway VMs as you need to RAS Gateway pools, you can easily scale your RAS Gateway deployment to optimize performance and capacity. When you add VMs to a pool, you can configure these RAS Gateways with site-to-site VPN connections of any kind (IKEv2, L3, GRE), eliminating capacity bottlenecks with no down time.

Simplified Enterprise Site Gateway Management

When your tenant has multiple Enterprise sites, the tenant can configure all sites with one remote site-to-site VPN IP address and a single remote neighbor IP address - your CSP datacenter RAS Gateway BGP Route Reflector VIP for that tenant. This simplifies gateway management for your tenants.

Fast Remediation of Gateway Failure

To ensure a fast failover response, you can configure the BGP Keepalive parameter time between edge routes and the control router to a short time interval, such as less than or equal to ten seconds. With this short keep alive interval, if a RAS Gateway BGP edge router fails, the failure is quickly detected and Network Controller follows the steps provided in previous sections. This advantage might reduce the need for a separate failure detection protocol, such as Bidirectional Forwarding Detection (BFD) protocol.

RAS Gateway High Availability

9/1/2018 • 11 minutes to read • [Edit Online](#)

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

You can use this topic to learn about high availability configurations for the RAS Multitenant Gateway for Software Defined Networking (SDN).

This topic contains the following sections.

- [RAS Gateway Overview](#)
- [Gateway Pools Overview](#)
- [RAS Gateway Deployment Overview](#)
- [RAS Gateway Integration with Network Controller](#)

RAS Gateway Overview

If your organization is a Cloud Service Provider (CSP) or an Enterprise with multiple tenants, you can deploy RAS Gateway in multitenant mode to provide network traffic routing to and from virtual and physical networks, including the Internet.

You can deploy RAS Gateway in multitenant mode as an edge gateway to route tenant customer network traffic to tenant virtual networks and resources.

When you deploy multiple instances of RAS Gateway VMs that provide high availability and failover, you are deploying a gateway pool. In Windows Server 2012 R2, all the gateway VMs formed a single pool, which made a logical separation of the gateway deployment a little difficult. Windows Server 2012 R2 gateway offered a 1:1 redundancy deployment for the gateway VMs, which resulted in under-utilization of the available capacity for site-to-site (S2S) VPN connections.

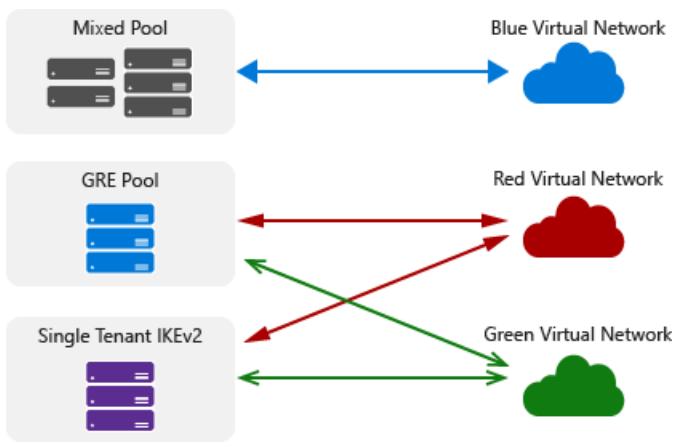
This issue is resolved in Windows Server 2016, which provides multiple Gateway Pools - which can be of different types for logical separation. The new mode of M+N redundancy allows for a more efficient failover configuration.

For more overview information about RAS Gateway, see [RAS Gateway](#).

Gateway Pools Overview

In Windows Server 2016, you can deploy gateways in one or more pools.

The following illustration shows different types of gateway pools that provide traffic routing between virtual networks.



Each pool has the following properties:

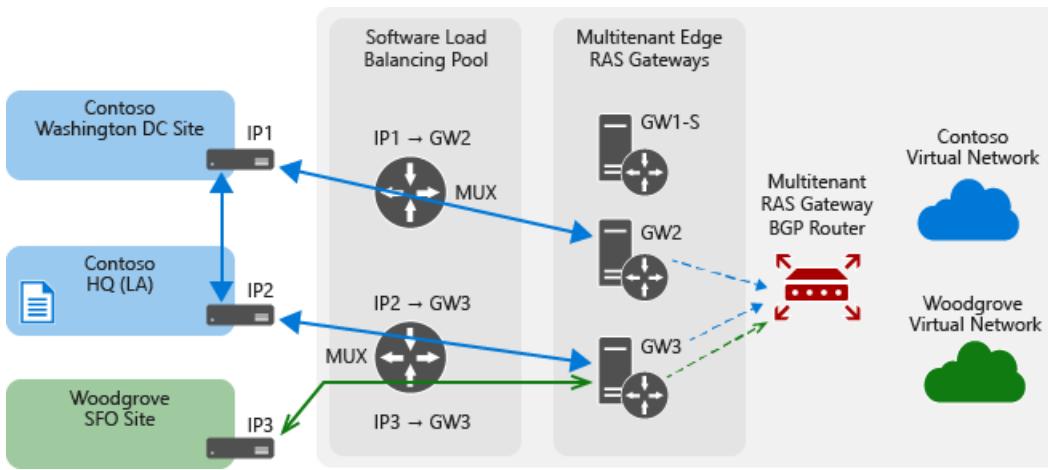
- Each pool is M+N redundant. This means that an 'M' number of active gateway VMs are backed up by an 'N' number of standby gateway VMs. The value of N (standby gateways) is always less than or equal to M (active gateways).
- A pool can perform any of the individual gateway functions - Internet Key Exchange version 2 (IKEv2) S2S, Layer 3 (L3), and Generic Routing Encapsulation (GRE) - or the pool can perform all of these functions.
- You can assign a single public IP address to all pools or to a subset of pools. Doing so greatly reduces the number of public IP addresses that you must use, because it is possible to have all tenants connect to the cloud on a single IP address. The section below on High Availability and Load balancing describes how this works.
- You can easily scale a gateway pool up or down by adding or removing gateway VMs in the pool. Removal or addition of gateways does not disrupt the services that are provided by a pool. You can also add and remove entire pools of gateways.
- Connections of a single tenant can terminate on multiple pools and multiple gateways in a pool. However, if a tenant has connections terminating in an **All** type gateway pool, it cannot subscribe to other **All** type or individual type gateway pools.

Gateway pools also provide the flexibility to enable additional scenarios:

- Single-tenant pools - you can create one pool for use by one tenant.
- If you are selling cloud services through partner (reseller) channels, you can create separate sets of pools for every reseller.
- Multiple pools can provide the same gateway function but different capacities. For example, you can create a gateway pool that supports both high throughput and low throughput IKEv2 S2S connections.

RAS Gateway Deployment Overview

The following illustration demonstrates a typical Cloud Service Provider (CSP) deployment of RAS Gateway.



With this type of deployment, the gateway pools are deployed behind a Software Load Balancer (SLB), which enables the CSP to assign a single public IP address for the entire deployment. Multiple gateway connections of a tenant can terminate on multiple gateway pools - and also on multiple gateways within a pool. This is illustrated through IKEv2 S2S connections in the above diagram, but the same is applicable to other gateway functions too, such as L3 and GRE gateways.

In the illustration, the MT BGP device is a RAS Multitenant Gateway with BGP. Multitenant BGP is used for dynamic routing. The routing for a tenant is centralized - a single point, called the route reflector (RR), handles the BGP peering for all tenant sites. The RR itself is distributed across all gateways in a pool. This results in a configuration where the connections of a tenant (data path) terminate on multiple gateways, but the RR for the tenant (BGP peering point - control path) is on only one of the gateways.

The BGP router is separated out in the diagram to depict this centralized routing concept. The gateway BGP implementation also provides transit routing, which enables the cloud to act as a transit point for routing between two tenant sites. These BGP capabilities are applicable to all gateway functions.

RAS Gateway Integration with Network Controller

RAS Gateway is fully integrated with Network Controller in Windows Server 2016. When RAS Gateway and Network Controller are deployed, Network Controller performs the following functions.

- Deployment of the gateway pools
- Configuration of tenant connections on each gateway
- Switching network traffic flows to a standby gateway in the event of a gateway failure

The following sections provide detailed information about RAS Gateway and Network Controller.

- [Provisioning and Load balancing of Gateway Connections \(IKEv2, L3, and GRE\)](#)
- [High Availability for IKEv2 S2S](#)
- [High Availability for GRE](#)
- [High Availability for L3 Forwarding Gateways](#)

Provisioning and Load Balancing of Gateway Connections (IKEv2, L3, and GRE)

When a tenant requests a gateway connection, the request is sent to Network Controller. Network Controller is configured with information about all of the gateway pools, including the capacity of each pool and every gateway in every pool. Network Controller selects the correct pool and gateway for the connection. This selection is based on the bandwidth requirement for the connection. Network Controller uses a "best fit" algorithm to pick connections efficiently in a pool. The BGP peering point for the connection is also designated at this time if this is the first connection of the tenant.

After Network Controller selects a RAS Gateway for the connection, Network Controller provisions the necessary configuration for the connection on the gateway. If the connection is an IKEv2 S2S connection, Network Controller also provisions a Network Address Translation (NAT) rule on the SLB pool; this NAT rule on the SLB pool directs connection requests from the tenant to the designated gateway. Tenants are differentiated by the source IP, which is expected to be unique.

NOTE

L3 and GRE connections bypass the SLB and connect directly with the designated RAS Gateway. These connections require that the remote endpoint router (or other third party device) must be correctly configured to connect with the RAS Gateway.

If BGP routing is enabled for the connection, then BGP peering is initiated by RAS Gateway - and routes are exchanged between on-premises and cloud gateways. The routes that are learned by BGP (or that are statically configured routes if BGP is not used) are sent to Network Controller. Network Controller then plumbs the routes down to the Hyper-V hosts upon which the tenant VMs are installed. At this point, tenant traffic can be routed to the correct on-premises site. Network Controller also creates associated Hyper-V Network Virtualization policies that specify gateway locations, and plumbs them down to the Hyper-V hosts.

High Availability for IKEv2 S2S

A RAS Gateway in a pool consists of both connections and BGP peering of different tenants. Every pool has 'M' active gateways and 'N' standby gateways.

Network Controller handles the failure of gateways in the following manner.

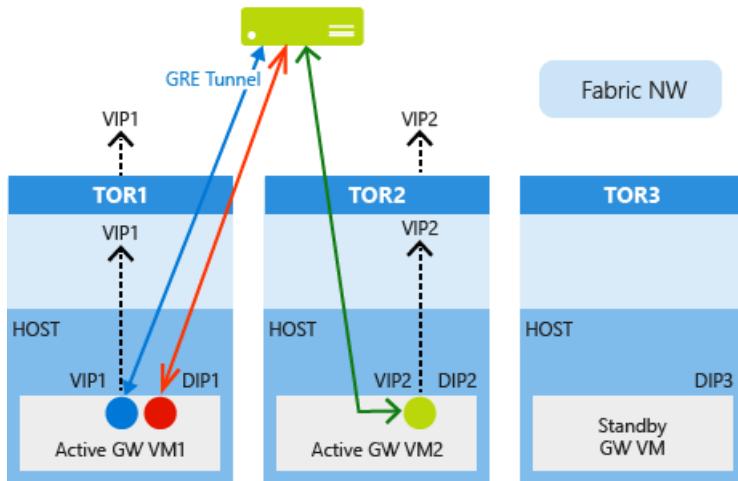
- Network Controller constantly pings the gateways in all pools and can detect a gateway that is failed or failing. Network Controller can detect the following types of RAS Gateway failures.
 - RAS Gateway VM failure
 - Failure of the Hyper-V host upon which the RAS Gateway is running
 - RAS Gateway service failure

Network Controller stores the configuration of all deployed active gateways. Configuration consists of connection settings and routing settings.

- When a gateway fails, it impacts tenant connections on the gateway, as well as tenant connections that are located on other gateways but whose RR resides on the failed gateway. The down time of the latter connections is less than the former. When Network Controller detects a failed gateway, it performs the following tasks.
 - Removes the routes of the impacted connections from the compute hosts.
 - Removes the Hyper-V Network Virtualization policies on these hosts.
 - Selects a standby gateway, converts it into an active gateway, and configures the gateway.
 - Changes the NAT mappings on the SLB pool to point connections to the new gateway.
- Simultaneously, as the configuration comes up on the new active gateway, the IKEv2 S2S connections and BGP peering are re-established. The connections and BGP peering can be initiated by either the cloud gateway or the on-premises gateway. The gateways refresh their routes and send them to Network Controller. After Network Controller learns the new routes discovered by the gateways, Network Controller sends the routes and the associated Hyper-V Network Virtualization policies to the Hyper-V hosts where the VMs of the failure-impacted tenants reside. This Network Controller activity is similar to the circumstance of a new connection setup, only it occurs on a larger scale.

High Availability for GRE

The process of RAS Gateway failover response by Network Controller - including failure detection, copying connection and routing configuration to the standby gateway, failover of BGP/static routing of the impacted connections (including the withdrawal and re-plumbing of routes on compute hosts and BGP re-peering), and reconfiguration of Hyper-V Network Virtualization policies on compute hosts - is the same for GRE gateways and connections. The re-establishment of GRE connections happens differently, however, and the high availability solution for GRE has some additional requirements.



At the time of gateway deployment, every RAS Gateway VM is assigned a Dynamic IP address (DIP). In addition, every gateway VM is also assigned a virtual IP address (VIP) for GRE high availability. VIPs are assigned only to gateways in pools that can accept GRE connections, and not to non-GRE pools. The VIPs assigned are advertised to the top of rack (TOR) switches using BGP, which then further advertises the VIPs into the cloud physical network. This makes the gateways reachable from the remote routers or third party devices where the other end of the GRE connection resides. This BGP peering is different than the tenant-level BGP peering for the exchange of tenant routes.

At the time of GRE connection provisioning, Network Controller selects a gateway, configures a GRE endpoint on the selected gateway, and returns back the VIP address of the assigned gateway. This VIP is then configured as the destination GRE tunnel address on the remote router.

When a gateway fails, Network Controller copies the VIP address of the failed gateway and other configuration data to the standby gateway. When the standby gateway becomes active, it advertises the VIP to its TOR switch and further into the physical network. Remote routers continue to connect GRE tunnels to the same VIP and the routing infrastructure ensures that packets are routed to the new active gateway.

High Availability for L3 Forwarding Gateways

A Hyper-V Network Virtualization L3 forwarding gateway is a bridge between the physical infrastructure in the datacenter and the virtualized infrastructure in the Hyper-V Network Virtualization cloud. On a multitenant L3 forwarding gateway, each tenant uses its own VLAN tagged logical network for connectivity with the tenant's physical network.

When a new tenant creates a new L3 gateway, Network Controller Gateway Service Manager selects an available gateway VM and configures a new tenant interface with a highly available Customer Address (CA) space IP address (from the tenant's VLAN tagged logical network). The IP address is used as the peer IP address on the remote (physical network) gateway, and is the Next-Hop to reach the tenant's Hyper-V Network Virtualization network.

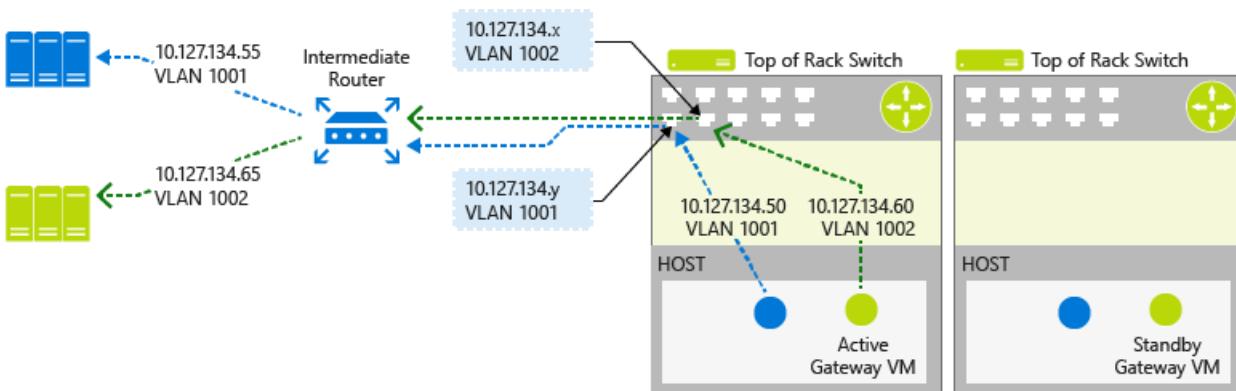
Unlike IPsec or GRE network connections, the TOR switch will not learn the tenant's VLAN tagged network dynamically. The routing for the tenant's VLAN tagged network needs to be configured on the TOR switch and all the intermediate switches and routers between physical infrastructure and the gateway to ensure end to end connectivity. Following is an example CSP Virtual Network configuration as depicted in the illustration below.

NETWORK	SUBNET	VLAN ID	DEFAULT GATEWAY
Contoso L3 Logical Network	10.127.134.0/24	1001	10.127.134.1
Woodgrove L3 Logical Network	10.127.134.0/24	1002	10.127.134.1

Following are example tenant gateway configurations as depicted in the illustration below.

TENANT NAME	L3 GATEWAY IP ADDRESS	VLAN ID	PEER IP ADDRESS
Contoso	10.127.134.50	1001	10.127.134.55
Woodgrove	10.127.134.60	1002	10.127.134.65

Following is the illustration of these configurations in a CSP datacenter.



The gateway failures, failure detection, and the gateway failover process in the context of an L3 forwarding gateway is similar to the processes for IKEv2 and GRE RAS Gateways. The differences are in the way the external IP addresses are handled.

When the gateway VM state becomes unhealthy, Network Controller selects one of the standby gateways from the pool and re-provisions the network connections and routing on the standby gateway. While moving the connections, the L3 Forwarding gateway's highly available CA space IP address is also moved to the new gateway VM along with the CA space BGP IP address of the tenant.

Because the L3 Peering IP address is moved to the new gateway VM during the failover, the remote physical infrastructure is again able to connect to this IP address and, subsequently, reach the Hyper-V Network Virtualization workload. For BGP dynamic routing, as the CA space BGP IP address is moved to the new gateway VM, the remote BGP Router can re-establish peering and learn all Hyper-V Network Virtualization routes again.

NOTE

You must separately configure the TOR switches and all of the intermediate routers in order to use the VLAN tagged logical network for tenant communication. In addition, L3 failover is restricted to only the racks which are configured in this way. Because of this, the L3 gateway pool must be carefully configured and manual configuration must be completed separately.

Software Load Balancing (SLB) for SDN

9/1/2018 • 10 minutes to read • [Edit Online](#)

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

You can use this topic to learn about Software Load Balancing for Software Defined Networking in Windows Server 2016.

Cloud Service Providers (CSPs) and Enterprises that are deploying Software Defined Networking (SDN) in Windows Server 2016 can use Software Load Balancing (SLB) to evenly distribute tenant and tenant customer network traffic among virtual network resources. The Windows Server SLB enables multiple servers to host the same workload, providing high availability and scalability.

Windows Server SLB includes the following capabilities.

- Layer 4 (L4) load balancing services for 'North-South' and 'East-West' TCP/UDP traffic.
- Public and Internal network traffic load balancing.
- Supports dynamic IP addresses (DIPs) on virtual Local Area Networks (VLANS) and on virtual networks that you create by using Hyper-V Network Virtualization.
- Health probe support.
- Ready for cloud scale, including scale-out capability, and scale up capability for multiplexers and Host Agents.

For more information, see [Software Load Balancing Features](#) in this topic.

NOTE

Multitenancy for VLANs is not supported by Network Controller, however you can use VLANs with SLB for service provider managed workloads, such as the datacenter infrastructure and high density Web servers.

Using Windows Server SLB, you can scale out your load balancing capabilities using SLB VMs on the same Hyper-V compute servers that you use for your other VM workloads. Because of this, SLB supports the rapid creation and deletion of load balancing endpoints that is required for CSP operations. In addition, Windows Server SLB supports tens of gigabytes per cluster, provides a simple provisioning model, and is easy to scale out and in.

How SLB works

SLB works by mapping virtual IP addresses (VIPs) to dynamic IP addresses (DIPs) that are part of a cloud service set of resources in the datacenter.

VIPs are single IP addresses that provide public access to a pool of load balanced VMs. For example, VIPs are IP addresses that are exposed on the Internet so that tenants and tenant customers can connect to tenant resources in the cloud datacenter.

DIPs are the IP addresses of the member VMs of a load balanced pool behind the VIP. DIPs are assigned within the cloud infrastructure to the tenant resources.

VIPs are located in the SLB Multiplexer (MUX). The MUX consists of one or more virtual machines (VMs). Network Controller provides each MUX with each VIP, and each MUX in turn uses Border Gateway Protocol (BGP) to advertise each VIP to routers on the physical network as a /32 route. BGP allows the physical network

routers to:

- Learn that a VIP is available on each MUX, even if the MUXes are on different subnets in a layer 3 network.
- Spread the load for each VIP across all available MUXes using Equal Cost Multi-Path (ECMP) routing.
- Automatically detect a MUX failure or removal and stop sending traffic to the failed MUX.
- Spread the load from the failed or removed MUX across the healthy MUXes.

When public traffic arrives from the Internet, the SLB MUX examines the traffic, which contains the VIP as a destination, and maps and rewrites the traffic so that it will arrive at an individual DIP. For inbound network traffic, this transaction is performed in a two-step process that is split between the MUX virtual machines (VMs) and the Hyper-V host where the destination DIP is located:

- Load balance - the MUX uses the VIP to select a DIP, encapsulates the packet, and forwards the traffic to the Hyper-V host where the DIP is located.
- Network Address Translation (NAT) - the Hyper-V host removes encapsulation from the packet, translates the VIP to a DIP, remaps the ports, and forwards the packet to the DIP VM.

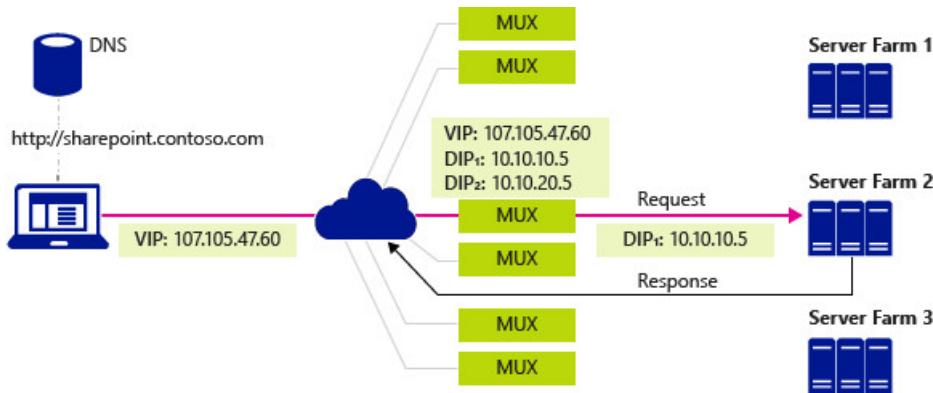
The MUX knows how to map VIPs to the correct DIPs because of load balancing policies that you define by using Network Controller. These rules include Protocol, Front-end Port, Back-end port, and distribution algorithm (5, 3, or 2 tuples).

When tenant VMs respond and send outbound network traffic back to the Internet or remote tenant locations, because the NAT is performed by the Hyper-V host, the traffic bypasses the MUX and goes directly to the edge router from the Hyper-V host. This MUX bypass process is called Direct Server Return (DSR).

And after the initial network traffic flow is established, the inbound network traffic bypasses the SLB MUX completely.

In the following illustration, a client computer performs a DNS query for the IP address of a company Sharepoint site - in this case, a fictional company named Contoso. The following process occurs.

- The DNS server returns the VIP 107.105.47.60 to the client.
- The client sends an HTTP request to the VIP.
- The physical network has multiple paths available to reach the VIP located on any MUX. Each router along the way uses ECMP to pick the next segment of the path until the request arrives at a MUX.
- The MUX that receives the request checks configured policies, and sees that there are two DIPs available, 10.10.10.5 and 10.10.20.5, on a virtual network to handle the request to the VIP 107.105.47.60
- The MUX selects the DIP 10.10.10.5 and encapsulates the packets using VXLAN so it can send it to the host containing the DIP using the hosts physical network address.
- The host receives the encapsulated packet and inspects it. It removes the encapsulation and rewrites the packet so the destination is now the DIP 10.10.10.5 instead of the VIP and sends the traffic to DIP VM.
- The request has now reached the Contoso Sharepoint site in Server Farm 2. The server generates a response and sends it to the client, using its own IP address as the source.
- The host intercepts the outgoing packet in the virtual switch which remembers that the client, now the destination, made the original request to the VIP. The host rewrites the source of the packet to be the VIP so that to the client does not see the DIP address.
- The host forwards the packet directly to the default gateway for the physical network which uses its standard routing table to forward the packet on to the client which eventually receives the response.



Load balancing internal datacenter traffic

When load balancing network traffic internal to the datacenter, such as between tenant resources that are running on different servers and are members of the same virtual network, the Hyper-V Virtual Switch to which the VMs are connected performs NAT.

With internal traffic load balancing, the first request is sent to and processed by the MUX, which selects the appropriate DIP and routes the traffic to the DIP. From that point forward, the established traffic flow bypasses the MUX and goes directly from VM to VM.

Health probes

SLB includes health probes to validate the health of the network infrastructure, including the following.

- TCP probe to port
- HTTP probe to port and URL

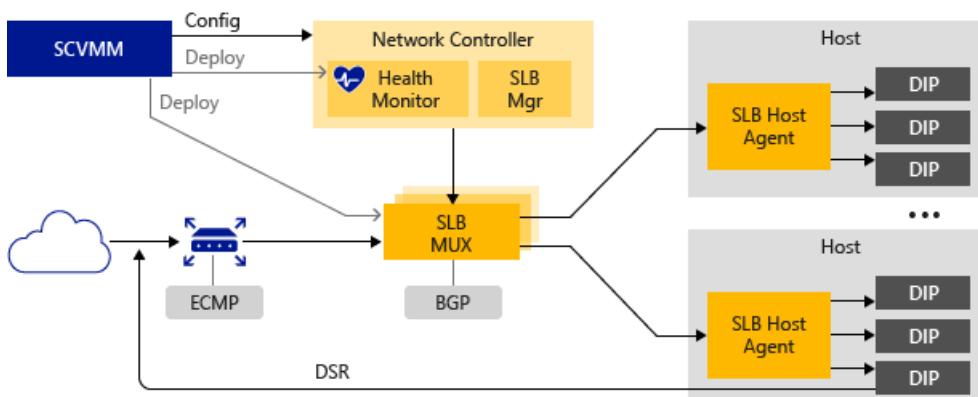
Unlike a traditional load balancer appliance where the probe originates on the appliance and travels across the wire to the DIP, the SLB probe originates on the host where the DIP is located and goes directly from the SLB host agent to the DIP, further distributing the work across the hosts.

Software Load Balancing Infrastructure

To deploy Windows Server SLB, you must first deploy Network Controller in Windows Server 2016 and one or more SLB MUX VMs.

In addition, you must configure Hyper-V hosts with the SDN-enabled Hyper-V Virtual Switch and ensure that the SLB Host Agent is running. The routers that serve the hosts must support equal cost multipath (ECMP) routing and Border Gateway Protocol (BGP) and must be configured to accept BGP peering requests from the SLB MUXes.

Following is an overview of the SLB infrastructure.



The following sections provide more information about these elements of the SLB infrastructure.

SCVMM

With System Center 2016, you can configure Network Controller on Windows Server 2016, including the SLB Manager and Health Monitor. You can also use System Center to deploy SLB MUXes and to install SLB Host Agents on computers that are running Windows Server 2016 and Hyper-V.

For more information about System Center 2016, see [System Center 2016](#).

NOTE

If you do not want to use System Center 2016, you can use Windows PowerShell or another management application to install and configure Network Controller and other SLB infrastructure. For more information, see [Deploy Network Controller using Windows PowerShell](#).

Network Controller

Network Controller hosts the SLB Manager and performs the following actions for SLB.

- Processes SLB commands that come in through the Northbound API from System Center, Windows PowerShell, or another network management application.
- Calculates policy for distribution to Hyper-V hosts and SLB MUXes.
- Provides the health status of the SLB infrastructure.

SLB MUX

The SLB MUX processes inbound network traffic and maps VIPs to DIPs, then forwards the traffic to the correct DIP. Each MUX also uses BGP to publish VIP routes to edge routers. BGP Keep Alive notifies MUXes when a MUX fails, which allows active MUXes to redistribute the load in case of a MUX failure - essentially providing load balancing for the load balancers.

Hosts that are running Hyper-V

You can use SLB with computers that are running Windows Server 2016 and Hyper-V. The VMs on the Hyper-V host can run any operating system that is supported by Hyper-V.

SLB Host Agent

When you deploy SLB, you must use System Center, Windows PowerShell, or another management application to deploy the SLB Host Agent on every Hyper-V host computer. You can install the SLB Host Agent on all versions of Windows Server 2016 that provide Hyper-V support, including Nano Server.

The SLB Host Agent listens for SLB policy updates from Network Controller. In addition, the host agent programs rules for SLB into the SDN-enabled Hyper-V Virtual Switches that are configured on the local computer.

SDN Enabled Hyper-V Virtual Switch

For a virtual switch to be compatible with SLB, you must use Hyper-V Virtual Switch Manager or Windows PowerShell commands to create the switch, and then you must enable Virtual Filtering Platform (VFP) for the virtual switch.

For information on enabling VFP on virtual switches, see the Windows PowerShell commands [Get-VMSystemSwitchExtension](#) and [Enable-VMSSwitchExtension](#).

The SDN enabled Hyper-V Virtual Switch performs the following actions for SLB.

- Processes the data path for SLB.
- Receives inbound network traffic from the MUX.
- Bypasses the MUX for outbound network traffic, sending it to the router using DSR.

- Runs on Nano Server instances of Hyper-V.

BGP Enabled Router

The BGP router performs the following actions for SLB.

- Routes inbound traffic to the MUX using ECMP.
- For outbound network traffic, uses the route provided by the host.
- Listens for route updates for VIPs from SLB MUX.
- Removes SLB MUXes from the SLB rotation if Keep Alive fails.

Software Load Balancing Features

Following are some of the features and capabilities of SLB.

Core functionality

- SLB provides Layer 4 load balancing services for 'North-South' and 'East-West' TCP/UDP traffic
- You can use SLB on a Hyper-V Network Virtualization-based network
- You can use SLB with a VLAN-based network for DIP VMs connected to a SDN Enabled Hyper-V Virtual Switch.
- One SLB instance can handle multiple tenants
- SLB and DIP support a scalable and low-latency return path, as implemented by Direct Server Return (DSR)
- SLB functions when you are also using Switch Embedded Teaming (SET) or Single Root Input/Output Virtualization (SR-IOV)
- SLB includes Internet Protocol version 4 (IPv4) support
- For site-to-site gateway scenarios, SLB provides NAT functionality to enable all site-to-site connections to utilize a single public IP
- You can install SLB, including the Host Agent and the MUX, on Windows Server 2016, Full, Core, and Nano Install.

Scale and performance

- Ready for cloud scale, including scale-out capability, and scale up capability for MUXes and Host Agents.
- One active SLB Manager Network Controller module can support 8 MUX instances

High availability

- You can deploy SLB to more than 2 nodes in an active/active configuration
- MUXes can be added and removed from the MUX pool without impacting the SLB service. This maintains SLB availability when individual MUXes are being patched.
- Individual MUX instances have an uptime of 99%
- Health monitoring data is available to management entities

Alignment

- You can deploy and configure SLB with SCVMM

- SLB provides a multitenant unified edge by seamlessly integrating with Microsoft appliances such as the RAS Multitenant Gateway, Datacenter Firewall, and Route Reflector.

Switch Embedded Teaming for SDN

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

SET is an alternative NIC Teaming solution that you can use in environments that include Hyper-V and the Software Defined Networking (SDN) stack in Windows Server 2016. SET integrates some NIC Teaming functionality into the Hyper-V Virtual Switch.

SET allows you to group between one and eight physical Ethernet network adapters into one or more software-based virtual network adapters. These virtual network adapters provide fast performance and fault tolerance in the event of a network adapter failure.

For more information, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#).

Container Networking Overview

9/11/2018 • 2 minutes to read • [Edit Online](#)

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

In this topic, we give you an overview of the networking stack for Windows containers and we include links to additional guidance about creating, configuring, and managing container networks.

Windows Server Containers are a lightweight operating system virtualization method separating applications or services from other services that run on the same container host. Windows containers function similarly to virtual machines. When enabled, each container has a separate view of the operating system, processes, file system, registry, and IP addresses, which you can connect to virtual networks.

A Windows container shares a kernel with the container host and all containers running on the host. Because of the shared kernel space, these containers require the same kernel version and configuration. Containers provide application isolation through process and namespace isolation technology.

IMPORTANT

Windows containers do not provide a hostile security boundary and should not be used to isolate untrusted code.

With Windows containers, you can deploy a Hyper-V host, where you create one or more virtual machines on the VM hosts. Inside the VM hosts, containers get created, and the networking access is through a virtual switch running inside the virtual machine. You can use reusable images stored in a repository to deploy the operating system and services into containers. Each container has a virtual network adapter which connects to a virtual switch, forwarding inbound and outbound traffic. You can attach container endpoints to a local host network (such as NAT), the physical network or overlay virtual network created through the SDN stack.

For enforcing isolation between containers on the same host, you create a network compartment for each Windows Server and Hyper-V container. Windows Server containers use a Host vNIC to attach to the virtual switch. Hyper-V Containers use a Synthetic VM NIC (not exposed to the Utility VM) to attach to the virtual switch.

Related topics

- [Windows Container Networking](#): Learn how to create and manage container networks for non-overlay/SDN deployments.
- [Connect container endpoints to a tenant virtual network](#): Learn how to create and manage container networks for overlay virtual networks with SDN.

Plan a Software Defined Network Infrastructure

9/21/2018 • 13 minutes to read • [Edit Online](#)

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

Learn about deployment planning for a Software Defined Network infrastructure, including the hardware and software prerequisites.

Prerequisites

This topic describes a number of hardware and software prerequisites, including:

- **Configured security groups, log file locations, and dynamic DNS registration** You must prepare your datacenter for Network Controller deployment, which requires one or more computers or VMs and one computer or VM. Before you can deploy Network Controller, you must configure the security groups, log file locations (if needed), and dynamic DNS registration. If you have not prepared your datacenter for Network Controller deployment, see [Installation and Preparation Requirements for Deploying Network Controller](#) for details.
- **Physical network** You need access to your physical network devices to configure VLANs, Routing, BGP, Data Center Bridging (ETS) if using an RDMA technology, and Data Center Bridging (PFC) if using a RoCE based RDMA technology. This topic shows manual switch configuration as well as BGP Peering on Layer-3 switches / routers or a Routing and Remote Access Server (RRAS) virtual machine.
- **Physical compute hosts** These hosts run Hyper-V and are required to host SDN infrastructure and tenant virtual machines. Specific network hardware is required in these hosts for best performance, which is described later in the **Network hardware** section.

Physical network and compute host configuration

Each physical compute host requires network connectivity through one or more network adapters attached to a physical switch port(s). A Layer-2 **VLAN** supports networks divided into multiple logical network segments.

TIP

Use VLAN 0 for logical networks in access mode or untagged.

IMPORTANT

Windows Server 2016 Software Defined Networking supports IPv4 addressing for the underlay and the overlay. IPv6 is not supported.

Logical networks

Management and HNV Provider

All physical compute hosts must access the Management logical network and the HNV Provider logical network. For IP address planning purposes, each physical compute host must have at least one IP address assigned from the Management logical network. The network controller requires a reserved IP address to serve as the REST IP address.

A DHCP server can automatically assign IP addresses for the Management network, or you can manually assign

static IP address. The SDN stack automatically assigns IP addresses for the HNV Provider logical network for the individual Hyper-V hosts from an IP Pool specified through and managed by the Network Controller.

NOTE

The Network Controller assigns an HNV Provider IP address to a physical compute host only after the Network Controller Host Agent receives network policy for a specific tenant virtual machine.

IF...	THEN...
The logical networks use VLANs,	the physical compute host must connect to a trunked switch port that has access to these VLANs. It's important to note that the physical network adapters on the computer host must not have any VLAN filtering activated.
Using Switched-Embedded Teaming (SET) and have multiple NIC team members, such as network adapters,	you must connect all of the NIC team members for that particular host to the same Layer-2 broadcast domain.
The physical compute host is running additional infrastructure virtual machines, such as Network Controller, SLB/MUX, or Gateway,	that host must have an additional IP address assigned from the Management logical network for each of the virtual machines hosted. Also, each SLB/MUX infrastructure virtual machine must have an IP address reserved for the HNV Provider logical network. Failure to have an IP address reserved may result in duplicate IP addresses on your network.

For information about Hyper-V Network Virtualization (HNV), which you can use to virtualize networks in a Microsoft SDN deployment, see [Hyper-V Network Virtualization](#).

Gateways and the Software Load Balancer

Additional logical networks need to be created and provisioned for gateway and SLB usage. Make sure to obtain the correct IP prefixes, VLAN IDs, and gateway IP addresses for these networks.

Transit logical network	The RAS Gateway and SLB/MUX use the Transit logical network to exchange BGP peering information and North/South (external-internal) tenant traffic. The size of this subnet will typically be smaller than the others. Only physical compute hosts that run RAS Gateway or SLB/MUX virtual machines need to have connectivity to this subnet with these VLANs trunked and accessible on the switch ports to which the compute hosts' network adapters are connected. Each SLB/MUX or RAS Gateway virtual machine is statically assigned one IP address from the Transit logical network.
Public VIP logical network	The Public VIP logical network is required to have IP subnet prefixes that are routable outside of the cloud environment (typically Internet routable). These will be the front-end IP addresses used by external clients to access resources in the virtual networks including the front end VIP for the Site-to-site gateway.
Private VIP logical network	The Private VIP logical network is not required to be routable outside of the cloud as it is used for VIPs that are only accessed from internal cloud clients, such as the SLB Manager or private services.

GRE VIP logical network	The GRE VIP network is a subnet that exists solely for defining VIPs that are assigned to gateway virtual machines running on your SDN fabric for a S2S GRE connection type. This network does not need to be pre-configured in your physical switches or router and need not have a VLAN assigned.
--------------------------------	---

Sample network topology

Change the sample IP subnet prefixes and VLAN IDs for your environment.

NETWORK NAME	SUBNET	MASK	VLAN ID ON TRUCK	GATEWAY	RESERVATIONS (EXAMPLES)
Management	10.184.108.0	24	7	10.184.108.1	10.184.108.1 – Router 10.184.108.4 - Network Controller 10.184.108.10 - Compute host 110.184.108.11 - Compute host 210.184.108.X - Compute host X
HNV Provider	10.10.56.0	23	11	10.10.56.1	10.10.56.1 – Router 10.10.56.2 - SLB/MUX1
Transit	10.10.10.0	24	10	10.10.10.1	10.10.10.1 – Router
Public VIP	41.40.40.0	27	NA	41.40.40.1	41.40.40.1 – Router 41.40.40.2 - SLB/MUX VIP41.40.40.3 - IPSec S2S VPN VIP
Private VIP	20.20.20.0	27	NA	20.20.20.1	20.20.20.1 - Default GW (router)
GRE VIP	31.30.30.0	24	NA	31.30.30.1	31.30.30.1 - Default GW

Logical networks required for RDMA-based storage

If using RDMA-based storage, define a VLAN and subnet for each physical adapter (two adapters per node) in your compute and storage hosts.

IMPORTANT

For Quality of Service (QoS) to be appropriately applied, physical switches require a tagged VLAN for RDMA traffic.

Network Name	Subnet	Mask	VLAN ID on Truck	Gateway	Reservations (Examples)
Storage1	10.60.36.0	25	8	10.60.36.1	10.60.36.1 - Router 10.60.36.X - Compute host X 10.60.36.Y - Compute host Y 10.60.36.V - Compute cluster 10.60.36.W - Storage cluster
Storage2	10.60.36.128	25	9	10.60.36.129	10.60.36.129 - Router 10.60.36.X - Compute host X 10.60.36.Y - Compute host Y 10.60.36.V - Compute cluster 10.60.36.W - Storage cluster

Routing infrastructure

If you are deploying your SDN infrastructure using scripts, the Management, HNV Provider, Transit, and VIP subnets must be routable to each other on the physical network.

Routing information (e.g. next-hop) for the VIP subnets is advertised by the SLB/MUX and RAS Gateways into the physical network using internal BGP peering. The VIP logical networks do not have a VLAN assigned and is not pre-configured in the Layer-2 switch (e.g. Top-of-Rack switch).

You need to create a BGP peer on the router that is used by your SDN infrastructure to receive routes for the VIP logical networks advertised by the SLB/MUXes and RAS Gateways. BGP peering only needs to occur one way (from SLB/MUX or RAS Gateway to external BGP peer). Above the first layer of routing you can use static routes or another dynamic routing protocol such as OSPF, however, as previously stated, the IP subnet prefix for the VIP logical networks do need to be routable from the physical network to the external BGP peer.

BGP peering is typically configured in a managed switch or router as part of the network infrastructure. The BGP peer could also be configured on a Windows Server with the Remote Access Server (RAS) role installed in a Routing Only mode. This BGP router peer in the network infrastructure must be configured to have its own ASN and allow peering from an ASN that is assigned to the SDN components (SLB/MUX and RAS Gateways). You must obtain the following information from your physical router, or from the network administrator in control of

that router:

- Router ASN
- Router IP address
- ASN for use by SDN components (can be any AS number from the private ASN range)

NOTE

Four byte ASNs are not supported by the SLB/MUX. You must allocate two byte ASNs to the SLB/MUX and the router to which it connects. You can use 4 byte ASNs elsewhere in your environment.

You or your network administrator must configure the BGP router peer to accept connections from the ASN and IP address or subnet address of the Transit logical network that your RAS gateway and SLB/MUXes are using.

For more information, see [Border Gateway Protocol \(BGP\)](#).

Default gateways

Machines that are configured to connect to multiple networks, such as the physical hosts and gateway virtual machines must only have one default gateway configured. Configure the default gateway on the adapter used to reach the Internet.

For virtual machines, follow these rules to decide which network to use as the default gateway:

1. Use the Transit logical network as the default gateway if a virtual machine connects to the Transit network, or if it is multi-homed to the Transit network or any other network.
2. Use the Management network as the default gateway if a virtual machine only connects to the Management network.
3. Use the HNV Provider network for SLB/MUXes and RAS Gateways. Do not use the HNV Provider network as a default gateway.
4. Do not connect virtual machines directly to the Storage1, Storage2, Public VIP or Private VIP networks.

For Hyper-V hosts and storage nodes, use the Management network as the default gateway. The storage networks must never have a default gateway assigned.

Network hardware

You can use the following sections to plan network hardware deployment.

Network Interface Cards (NICs)

The network interface cards (NICs) used in your Hyper-V hosts and storage hosts require specific capabilities to achieve the best performance.

Remote Direct Memory Access (RDMA) is a kernel bypass technique that makes it possible to transfer large amounts of data without using the host CPU, which frees the CPU to perform other work.

Switch Embedded Teaming (SET) is an alternative NIC Teaming solution that you can use in environments that include Hyper-V and the Software Defined Networking (SDN) stack in Windows Server 2016. SET integrates some NIC Teaming functionality into the Hyper-V Virtual Switch.

For more information, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#).

To account for the overhead in tenant virtual network traffic caused by VXLAN or NVGRE encapsulation headers, the MTU of the Layer-2 fabric network (switches and hosts) must be set to greater than or equal to 1674 Bytes (including Layer-2 Ethernet headers).

NICs that support the new *EncapOverhead* advanced adapter keyword sets the MTU automatically through the network controller Host Agent. NICs that do not support the new *EncapOverhead* keyword need to set the MTU size manually on each physical host using the *JumboPacket* (or equivalent) keyword.

Switches

When selecting a physical switch and router for your environment, make sure it supports the following set of capabilities:

- Switchport MTU settings (required)
- MTU set to ≥ 1674 Bytes (including L2-Ethernet Header)
- L3 protocols (required)
- ECMP
- BGP (IETF RFC 4271)-based ECMP

Implementations should support the MUST statements in the following IETF standards.

- RFC 2545: "BGP-4 Multiprotocol extensions for IPv6 Inter-Domain Routing"
- RFC 4760: "Multiprotocol Extensions for BGP-4"
- RFC 4893: "BGP Support for Four-octet AS Number Space"
- RFC 4456: "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)"
- RFC 4724: "Graceful Restart Mechanism for BGP"

The following tagging protocols are required.

- VLAN - Isolation of various types of traffic
- 802.1q trunk

The following items provide Link control.

- Quality of service (PFC only required if using RoCE)
- Enhanced Traffic Selection (802.1Qaz)
- Priority Based Flow Control (802.1p/Q and 802.1Qbb)

The following items provide availability and redundancy.

- Switch availability (required)
- A highly available router is required to perform gateway functions. You can do this by using a multi-chassis switch\ router or technologies like VRRP.

The following items provide management capabilities.

Monitoring

- SNMP v1 or SNMP v2 (required if using Network Controller for physical switch monitoring)
- SNMP MIBs (required if you are using Network Controller for physical switch monitoring)
- MIB-II (RFC 1213), LLDP, Interface MIB (RFC 2863), IF-MIB, IP-MIB, IP-FORWARD-MIB, Q-BRIDGE-MIB, BRIDGE-MIB, LLDB-MIB, Entity-MIB, IEEE8023-LAG-MIB

The following diagrams show a sample four-node setup. For clarity purposes, the first diagram shows just the network controller, the second shows the network controller plus the software load balancer, and the third diagram shows the network controller, software load balancer, and the gateway.

These diagrams do not show storage networks and vNICs. If you plan to use SMB-based storage, these are required.

Both the infrastructure and tenant virtual machines can be redistributed across any physical compute host (assuming the correct network connectivity exists for the correct logical networks).

Switch configuration examples

To help configure your physical switch or router, a set of sample configuration files for a variety of switch models and vendors are available at the [Microsoft SDN Github repository](#). A detailed readme and tested command line interface (CLI) commands for specific switches are provided.

Compute

All Hyper-V hosts must have Windows Server 2016 installed, Hyper-V enabled, and an external Hyper-V virtual switch created with at least one physical adapter connected to the Management logical network. The host must be reachable via a Management IP address assigned to the Management Host vNIC.

Any storage type that is compatible with Hyper-V, shared or local may be used.

TIP

It is convenient if you use the same name for all your virtual switches, but it is not mandatory. If you plan to deploy with scripts, see the comment associated with the `vSwitchName` variable in the config.psd1 file.

Host compute requirements

The following table shows the minimum hardware and software requirements for the four physical hosts used in the example deployment.

HOST	HARDWARE REQUIREMENTS	SOFTWARE REQUIREMENTS
Physical Hyper-v host	4-Core 2.66 GHz CPU 32 GB of RAM 300 GB Disk Space 1 Gb/s (or faster) physical network adapter	OS: Windows Server 2016 Hyper-V Role installed

SDN infrastructure virtual machine role requirements

ROLE	VCPU REQUIREMENTS	MEMORY REQUIREMENTS	DISK REQUIREMENTS
Network controller (three node)	4 vCPUs	4 GB min (8 GB recommended)	75 GB for the OS drive
SLB/MUX (three node)	8 vCPUs	8 GB recommended	75 GB for the OS drive
RAS Gateway (single pool of three node gateways, two active, one passive)	8 vCPUs	8 GB recommended	75 GB for the OS drive
RAS Gateway BGP router for SLB/MUX peering (alternatively use ToR switch as BGP Router)	2 vCPUs	2 GB	75 GB for the OS drive

If you use VMM for deployment, additional infrastructure virtual machine resources are required for VMM and

other non-SDN infrastructure. For additional information, see [Minimum Hardware Recommendations for System Center Technical Preview](#).

Extending your infrastructure

The sizing and resource requirements for your infrastructure are dependent on the tenant workload virtual machines that you plan to host. The CPU, memory, and disk requirements for the infrastructure virtual machines (for example: network controller, SLB, gateway, etc.) are listed in the previous table. You can add more of these infrastructure virtual machines to scale out as needed. However, any tenant virtual machines running on the Hyper-V hosts have their own CPU, memory, and disk requirements that you must consider.

When the tenant workload virtual machines begin to consume too many resources on the physical Hyper-V hosts, you can extend your infrastructure by adding additional physical hosts. This can be done with Virtual Machine Manager or by using PowerShell scripts (depending on how you initially deployed the infrastructure) to create new server resources through the network controller. If you need to add additional IP addresses for the HNV Provider network, you can create new logical subnets (with corresponding IP Pools) that the hosts can use.

See Also

[Installation and Preparation Requirements for Deploying Network Controller Software Defined Networking \(SDN\)](#)

Requirements for Deploying Network Controller

9/21/2018 • 3 minutes to read • [Edit Online](#)

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

Prepare your datacenter for Network Controller deployment, which requires one or more computers or VMs and one computer or VM. Before you can deploy Network Controller, you must configure the security groups, log file locations (if needed), and dynamic DNS registration.

Network Controller requirements

Network Controller deployment requires one or more computers or VMs that serve as the Network Controller, and one computer or VM to serve as a management client for Network Controller.

- All VMs and computers planned as Network Controller nodes must be running Windows Server 2016 Datacenter edition.
- Any computer or virtual machine (VM) upon which you install Network Controller must be running the Datacenter edition of Windows Server 2016.
- The management client computer or VM for Network Controller must be running Windows 10.

Configuration requirements

Before deploying Network Controller, you must configure the security groups, log file locations (if needed), and dynamic DNS registration.

Step 1. Configure your security groups

The first thing you want to do is create two security groups for Kerberos authentication.

You create groups for users who have permission to:

1. Configure Network Controller

You can name this group Network Controller Admins, for example.

2. Configure and manage the network by using Network Controller

You can name this group Network Controller Users, for example. Use Representational State Transfer (REST) to configure and manage Network Controller.

NOTE

All of the users you add must be members of the Domain Users group in Active Directory Users and Computers.

Step 2. Configure log file locations if needed

The next thing you want to do is configure the file locations to store Network Controller debug logs either on the Network Controller computer or VM or on a remote file share.

NOTE

If you store the logs in a remote file share, ensure that the share is accessible from the Network Controller.

Step 3. Configure dynamic DNS registration for Network Controller

Finally, the next thing you want to do is deploy Network Controller cluster nodes on the same subnet or different subnets.

IF...	THEN...
On the same subnet,	You must provide the Network Controller REST IP address.
On different subnets,	You must provide the Network Controller REST DNS name, which you create during the deployment process. You must also do the following: <ul style="list-style-type: none">• Configure DNS dynamic updates for the Network Controller DNS name on the DNS server.• Restrict the DNS dynamic updates to Network Controller nodes only.

NOTE

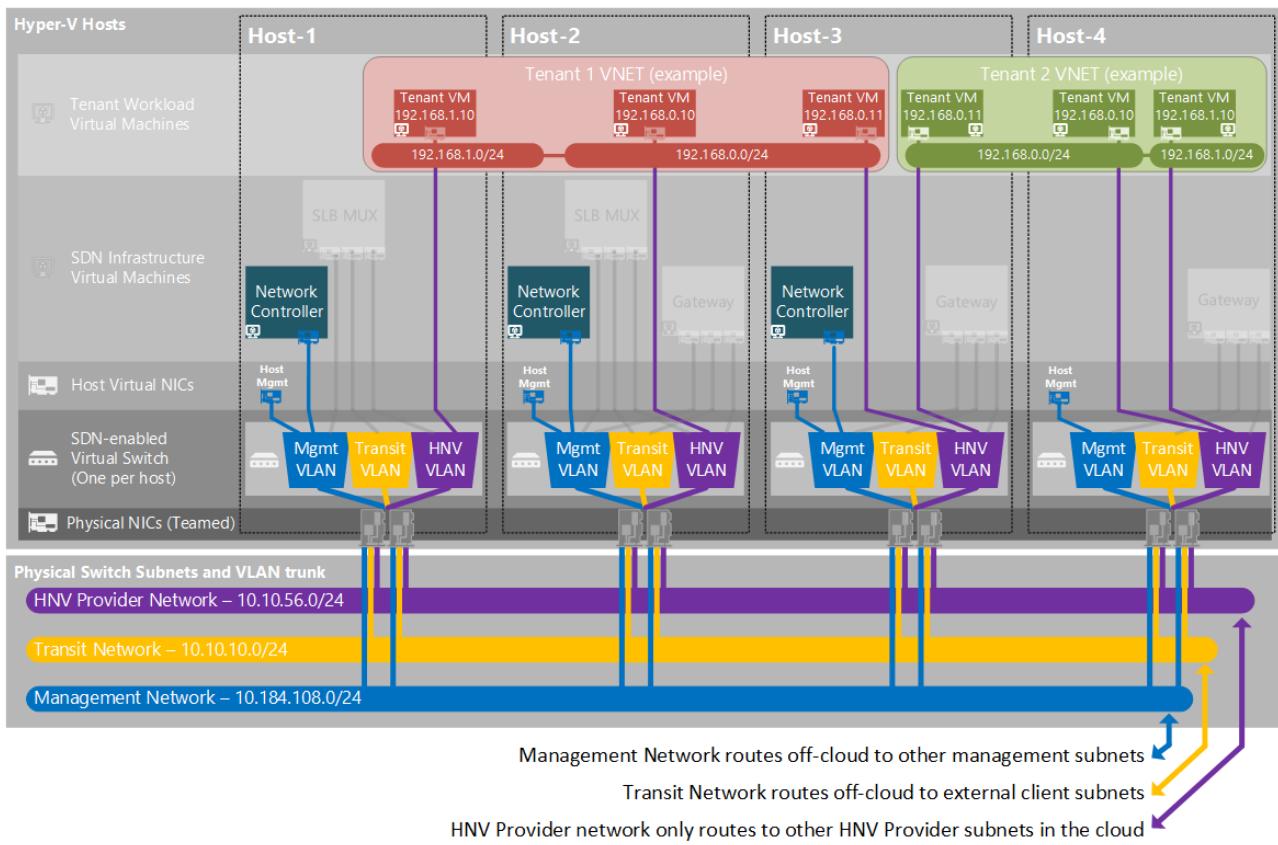
Membership in **Domain Admins**, or equivalent, is the minimum required to perform these procedures.

1. Allow DNS dynamic updates for a zone.
 - a. Open DNS Manager, and in the console tree, right-click the applicable zone, and then click **Properties**.
 - b. On the **General** tab, verify that the zone type is either **Primary** or **Active Directory-integrated**.
 - c. In **Dynamic updates**, verify that **Secure only** is selected, and then click **OK**.
2. Configure DNS zone security permissions for Network Controller nodes
 - a. Click the **Security** tab, and then click **Advanced**.
 - b. In **Advanced Security Settings**, click **Add**.
 - c. Click **Select a principal**.
 - d. In the **Select User, Computer, Service Account, or Group** dialog box, click **Object Types**.
 - e. In **Object Types**, select **Computers**, and then click **OK**.
 - f. In the **Select User, Computer, Service Account, or Group** dialog box, type the NetBIOS name of one of the Network Controller nodes in your deployment, and then click **OK**.
 - g. In **Permission Entry**, verify the following values:
 - **Type** = Allow
 - **Applies to** = This object and all descendant objects
 - h. In **Permissions**, select **Write all properties** and **Delete**, and then click **OK**.
3. Repeat for all computers and VMs in the Network Controller cluster.

Deployment options

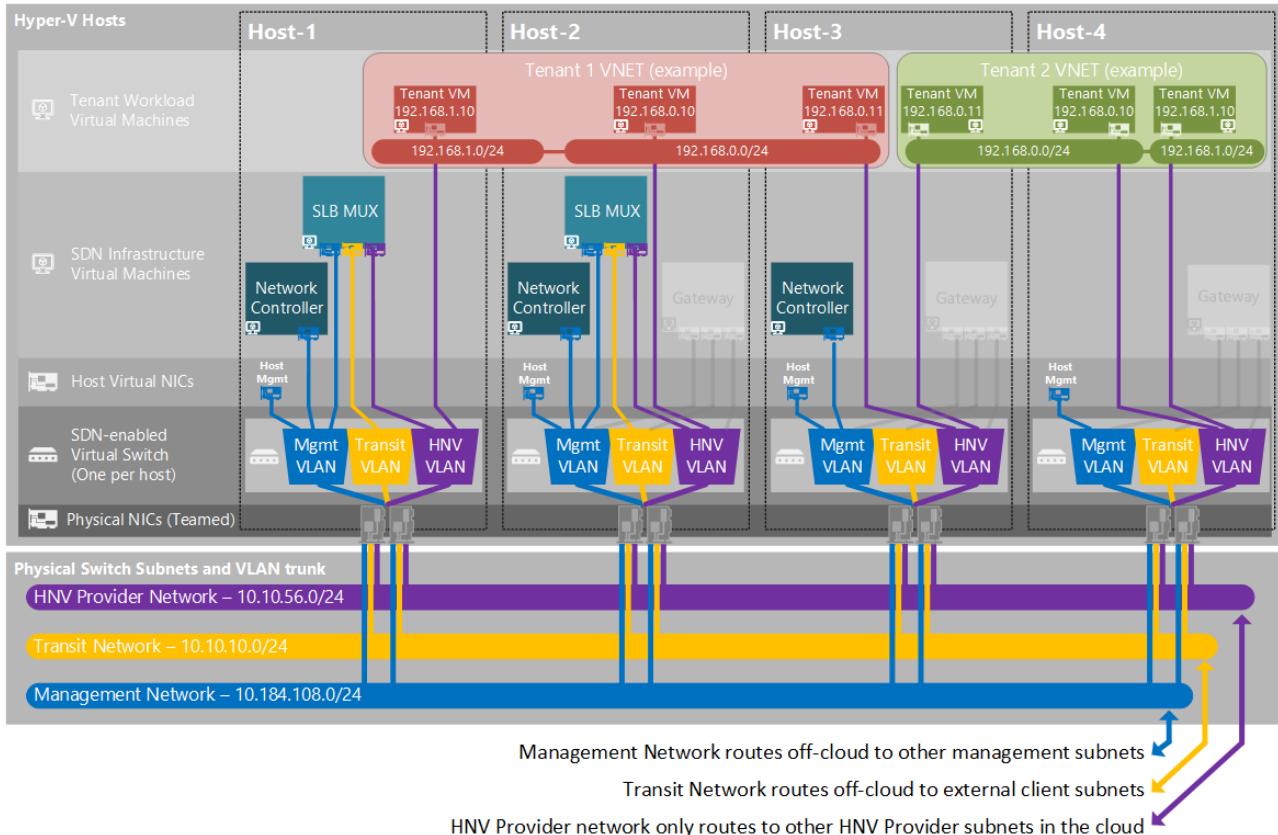
Network Controller deployment

The setup is highly available with three Network Controller nodes configured on virtual machines. Also shown is two tenants with Tenant 2's virtual network broken into two virtual subnets to simulate a web tier and a database tier.



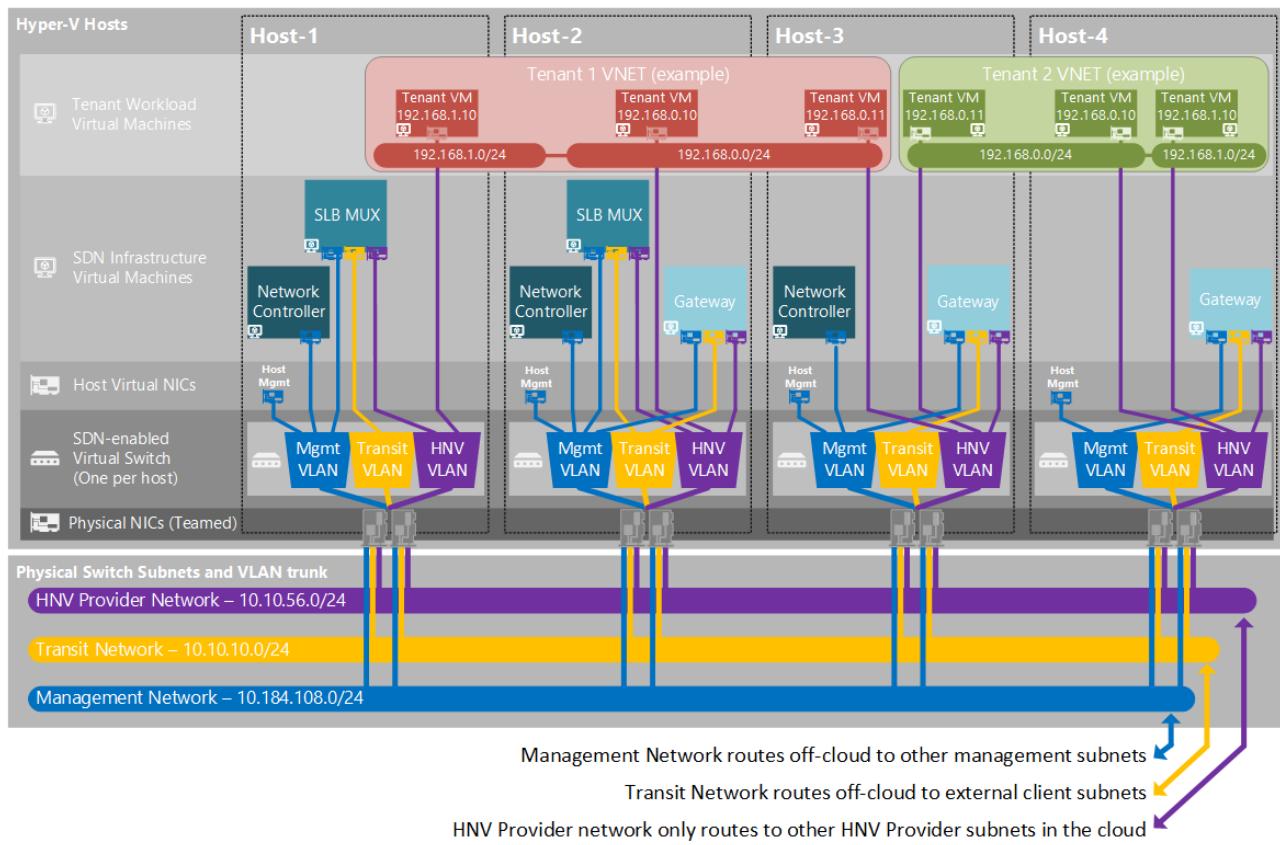
Network controller and software load balancer deployment

For high availability, there are two or more SLB/MUX nodes.



Network Controller, Software Load Balancer, and RAS Gateway deployment

There are three gateway virtual machines; two are active, and one is redundant.



For TP5-based deployment automation, Active Directory must be available and reachable from these subnets. For more information about Active Directory, see [Active Directory Domain Services Overview](#).

IMPORTANT

If you deploy using VMM, ensure your infrastructure virtual machines (VMM Server, AD/DNS, SQL Server, etc.) are not hosted on any of the four hosts shown in the diagrams.

Next steps

Plan a Software Defined Network Infrastructure.

Related topics

- [Network Controller](#)
- [Network Controller High Availability](#)
- [Deploy Network Controller using Windows PowerShell](#)
- [Install the Network Controller server role using Server Manager](#)

2 minutes to read

Deploy a Software Defined Network infrastructure

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

Deploy Microsoft's Software Defined Networking (SDN) infrastructure.

These deployments include all the technologies you need for a fully functional infrastructure, including Hyper-V Network Virtualization (HNV), network controllers, software load balancers (SLB/MUX), and gateways.

Set up SDN infrastructure in the VMM fabric

- [Set up a Software Defined Network \(SDN\) infrastructure in the VMM fabric](#)

Use this method if you want to incorporate System Center Virtual Machine Manager (VMM) to manage your SDN infrastructure.

Deploy SDN infrastructure using scripts

- [Deploy a Software Defined Network infrastructure using scripts](#)

Use this method if you do not want to use VMM to manage your SDN infrastructure, or if you have another management method.

Deploy individual SDN technologies instead of an entire infrastructure

If you want to deploy individual SDN technologies instead of an entire infrastructure, see:

[Deploy Software Defined Network Technologies using Windows PowerShell](#).

Related topics

- [Software Defined Networking \(SDN\)](#)
- [SDN Technologies](#)
- [Plan SDN](#)
- [Manage SDN](#)
- [Security for SDN](#)
- [Troubleshoot SDN](#)

Deploy a Software Defined Network infrastructure using scripts

9/21/2018 • 8 minutes to read • [Edit Online](#)

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

In this topic, you deploy a Microsoft Software Defined Network (SDN) infrastructure using scripts. The infrastructure includes a highly available (HA) network controller, an HA Software Load Balancer (SLB)/MUX, virtual networks, and associated Access Control Lists (ACLs). Additionally, another script deploys a tenant workload for you to validate your SDN infrastructure.

If you want your tenant workloads to communicate outside their virtual networks, you can setup SLB NAT rules, Site-to-Site Gateway tunnels, or Layer-3 Forwarding to route between virtual and physical workloads.

You can also deploy an SDN infrastructure using Virtual Machine Manager (VMM). For more information, see [Set up a Software Defined Network \(SDN\) infrastructure in the VMM fabric](#).

Pre-deployment

IMPORTANT

Before you begin deployment, you must plan and configure your hosts and physical network infrastructure. For more information, see [Plan a Software Defined Network Infrastructure](#).

All Hyper-V hosts must have Windows Server 2016 installed.

Deployment steps

Start by configuring the Hyper-V host's (physical servers) Hyper-V virtual switch and IP address assignment. Any storage type that is compatible with Hyper-V, shared or local may be used.

Install host networking

1. Install the latest network drivers available for your NIC hardware.
2. Install the Hyper-V role on all hosts (For more information, see [Get started with Hyper-V on Windows Server 2016](#).

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

3. Create the Hyper-V virtual switch.

Use the same switch name for all hosts, for example, **sdnSwitch**. Configure at least one network adapter or, if using SET, configure at least two network adapters. Maximum inbound spreading occurs when using two NICs.

```
New-VMSwitch "<switch name>" -NetAdapterName "<NetAdapter1>" [, "<NetAdapter2>" -EnableEmbeddedTeaming $True] -AllowManagementOS $True
```

TIP

You can skip steps 4 and 5 if you have separate Management NICs.

4. Refer to the planning topic ([Plan a Software Defined Network Infrastructure](#)) and work with your network administrator to obtain the VLAN ID of the Management VLAN. Attach the Management vNIC of the newly created Virtual Switch to the Management VLAN. This step can be omitted if your environment does not use VLAN tags.

```
Set-VMNetworkAdapterIsolation -ManagementOS -IsolationMode Vlan -DefaultIsolationID <Management VLAN> -  
AllowUntaggedTraffic $True
```

5. Refer to the planning topic ([Plan a Software Defined Network Infrastructure](#)) and work with your network administrator to use either DHCP or static IP assignments to assign an IP address to the Management vNIC of the newly created vSwitch. The following example shows how to create a static IP address and assign it to the Management vNIC of the vSwitch:

```
New-NetIPAddress -InterfaceAlias "vEthernet (<switch name>)" -IPAddress <IP> -DefaultGateway <Gateway  
IP> -AddressFamily IPv4 -PrefixLength <Length of Subnet Mask - for example: 24>
```

6. [Optional] Deploy a virtual machine to host Active Directory Domain Services ([Install Active Directory Domain Services \(Level 100\)](#)) and a DNS Server.

- a. Connect the Active Directory/DNS Server virtual machine to the Management VLAN:

```
```PowerShell  
Set-VMNetworkAdapterIsolation -VMName "<VM Name>" -Access -VlanId <Management VLAN> -
AllowUntaggedTraffic $True
```
```

- b. Install Active Directory Domain Services and DNS.

NOTE

The network controller supports both Kerberos and X.509 certificates for authentication. This guide uses both authentication mechanisms for different purposes (although only one is required).

7. Join all Hyper-V hosts to the domain. Ensure the DNS server entry for the network adapter that has an IP address assigned to the Management network points to a DNS server that can resolve the domain name.

```
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (<switch name>)" -ServerAddresses <DNS Server IP>
```

- a. Right-click **Start**, click **System**, and then click **Change Settings**.
- b. Click **Change**.
- c. Click **Domain** and specify the domain name.
- d. Click **OK**.
- e. Type the user name and password credentials when prompted.
- f. Restart the server.

Validation

Use the following steps to validate that host networking is setup correctly.

1. Ensure the VM Switch was created successfully:

```
Get-VMSwitch "<switch name>"
```

2. Verify that the Management vNIC on the VM Switch is connected to the Management VLAN:

NOTE

Relevant only if Management and Tenant traffic share the same NIC.

```
Get-VMNetworkAdapterIsolation -ManagementOS
```

3. Validate all Hyper-V hosts and external management resources, for example, DNS servers.

Ensure they are accessible via ping using their Management IP address and/or fully qualified domain name (FQDN).

```
ping <Hyper-V Host IP>
```

```
ping <Hyper-V Host FQDN>
```

4. Run the following command on the deployment host and specify the FQDN of each Hyper-V host to ensure the Kerberos credentials used provides access to all the servers.

```
winrm id -r:<Hyper-V Host FQDN>
```

Nano installation requirements and notes

If you use Nano as your Hyper-V hosts (physical servers) for the deployment, the following are additional requirements:

1. All Nano nodes need to have the DSC package installed with the language pack:

- Microsoft-NanoServer-DSC-Package.cab
- Microsoft-NanoServer-DSC-Package_en-us.cab

```
dism /online /add-package /packagepath:<Path> /loglevel:4
```

2. The SDN Express scripts must be run from a non-Nano host (Windows Server Core or Windows Server w/ GUI). PowerShell Workflows are not supported on Nano.
3. Invoking the Network Controller NorthBound API using PowerShell or NC REST Wrappers (which rely on Invoke-WebRequest and Invoke-RestMethod) must be done from a non-Nano host.

Run SDN Express scripts

1. Go to the [Microsoft SDN GitHub Repository](#) for the installation files.
2. Download the installation files from the repository to the designated deployment computer. Click **Clone or download** and then click **Download ZIP**.

NOTE

The designated deployment computer must be running Windows Server 2016 or later.

3. Expand the zip file and copy the **SDNExpress** folder to the deployment computer's `c:\` folder.

4. Share the `c:\SDNExpress` folder as "**SDNExpress**" with permission for **Everyone** to **Read/Write**.

5. Navigate to the `C:\SDNExpress` folder.

You see the following folders:

| FOLDER NAME | DESCRIPTION |
|-------------|---|
| AgentConf | Holds fresh copies of OVSDB schemas used by the SDN Host Agent on each Windows Server 2016 Hyper-V host to program network policy. |
| Certs | Temporary shared location for the NC certificate file. |
| Images | Empty, place your Windows Server 2016 vhdx image here |
| Tools | Utilities for troubleshooting and debugging. Copied to the hosts and virtual machines. We recommend you place Network Monitor or Wireshark here so it is available if needed. |
| Scripts | <p>Deployment scripts.</p> <ul style="list-style-type: none"> - SDNExpress.ps1
Deploys and configures the fabric, including the Network controller virtual machines, SLB Mux virtual machines, gateway pool(s) and the HNV gateway virtual machine(s) corresponding to the pool(s) . - FabricConfig.psd1
A configuration file template for the SDNExpress script. You will customize this for your environment. - SDNExpressTenant.ps1
Deploys a sample tenant workload on a virtual network with a load balanced VIP.
Also provisions one or more network connections (IPSec S2S VPN, GRE, L3) on the service provider edge gateways which are connected to the previously created tenant workload. The IPSec and GRE gateways are available for connectivity over the corresponding VIP IP Address, and the L3 forwarding gateway over the corresponding address pool.
This script can be used to delete the corresponding configuration with an Undo option as well. - TenantConfig.psd1
A template configuration file for tenant workload and S2S gateway configuration. - SDNExpressUndo.ps1
Cleans up the fabric environment and resets it to a starting state. - SDNExpressEnterpriseExample.ps1
Provisions one or more enterprise site environments with one Remote Access Gateway and (optionally) one corresponding enterprise virtual machine per site. The IPSec or GRE enterprise gateways connects to the corresponding VIP IP address of the service provider gateway to establish the S2S tunnels. The L3 Forwarding Gateway connects over the corresponding Peer IP Address. This script can be used to delete the corresponding configuration with an Undo option as well. - EnterpriseConfig.psd1
A template configuration file for the Enterprise site-to-site gateway and Client VM configuration. |

| FOLDER NAME | DESCRIPTION |
|-------------|--|
| TenantApps | Files used to deploy example tenant workloads. |

6. Verify the Windows Server 2016 VHDX file is in the **Images** folder.
7. Customize the SDNExpress\scripts\FabricConfig.psd1 file by changing the << Replace >> tags with specific values to fit your lab infrastructure including host names, domain names, usernames and passwords, and network information for the networks listed in the Planning Network topic.
8. Create a Host A record in DNS for the NetworkControllerRestName (FQDN) and NetworkControllerRestIP.
9. Run the script as a user with domain administrator credentials:

```
SDNExpress\scripts\SDNExpress.ps1 -ConfigurationDataFile FabricConfig.psd1 -Verbose
```

10. To undo all operations, run the following command:

```
SDNExpress\scripts\SDNExpressUndo.ps1 -ConfigurationDataFile FabricConfig.psd1 -Verbose
```

Validation

Assuming that the SDN Express script ran to completion without reporting any errors, you can perform the following step to ensure the fabric resources have been deployed correctly and are available for tenant deployment.

Use [Diagnostic Tools](#) to ensure there are no errors on any fabric resources in the network controller.

```
Debug-NetworkControllerConfigurationState -NetworkController <FQDN of Network Controller Rest Name>
```

Deploy a sample tenant workload with the software load balancer

Now that fabric resources have been deployed, you can validate your SDN deployment end-to-end by deploying a sample tenant workload. This tenant workload consists of two virtual subnets (web tier and database tier) protected via Access Control List (ACL) rules using the SDN distributed firewall. The web tier's virtual subnet is accessible through the SLB/MUX using a Virtual IP (VIP) address. The script automatically deploys two web tier virtual machines and one database tier virtual machine and connects these to the virtual subnets.

1. Customize the SDNExpress\scripts\TenantConfig.psd1 file by changing the << Replace >> tags with specific values (for example: VHD image name, network controller REST name, vSwitch Name, etc. as previously defined in the FabricConfig.psd1 file)
2. Run the script. For example:

```
SDNExpress\scripts\SDNExpressTenant.ps1 -ConfigurationDataFile TenantConfig.psd1 -Verbose
```

3. To undo the configuration, run the same script with the **undo** parameter. For example:

```
SDNExpress\scripts\SDNExpressTenant.ps1 -Undo -ConfigurationDataFile TenantConfig.psd1 -Verbose
```

Validation

To validate that the tenant deployment was successful, do the following:

1. Log into the database tier virtual machine and try to ping the IP address of one of the web tier virtual machines (ensure Windows Firewall is turned off in web tier virtual machines).
2. Check the network controller tenant resources for any errors. Run the following from any Hyper-V host with Layer-3 connectivity to the network controller:

```
Debug-NetworkControllerConfigurationState -NetworkController <FQDN of Network Controller REST Name>
```

3. To verify that the load balancer is running correctly, run the following from any Hyper-V host:

```
wget <VIP IP address>/unique.htm -disablekeepalive -usebasicparsing
```

where `<VIP IP address>` is the web tier VIP IP address you configured in the TenantConfig.psd1 file.

TIP

Search for the `VIPIP` variable in TenantConfig.psd1.

Run this multiple times to see the load balancer switch between the available DIPs. You can also observe this behavior using a web browser. Browse to `<VIP IP address>/unique.htm`. Close the browser and open a new instance and browse again. You will see the blue page and the green page alternate, except when the browser caches the page before the cache times out.

Deploy Software Defined Network technologies using Windows PowerShell

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the topics in this section to deploy individual SDN technologies using Windows PowerShell.

This section contains the following topics.

- [Deploy Network Controller using Windows PowerShell](#)

Deploy Network Controller using Windows PowerShell

9/21/2018 • 13 minutes to read • [Edit Online](#)

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

This topic provides instructions on using Windows PowerShell to deploy Network Controller on one or more virtual machines (VMs) that are running Windows Server 2016.

IMPORTANT

Do not deploy the Network Controller server role on physical hosts. To deploy Network Controller, you must install the Network Controller server role on a Hyper-V virtual machine (VM) that is installed on a Hyper-V host. After you have installed Network Controller on VMs on three different Hyper-V hosts, you must enable the Hyper-V hosts for Software Defined Networking (SDN) by adding the hosts to Network Controller using the Windows PowerShell command **New-NetworkControllerServer**. By doing so, you are enabling the SDN Software Load Balancer to function. For more information, see [New-NetworkControllerServer](#).

This topic contains the following sections.

- [Install the Network Controller server role](#)
- [Configure the Network Controller cluster](#)
- [Configure the Network Controller application](#)
- [Network Controller deployment validation](#)
- [Additional Windows PowerShell commands for Network Controller](#)
- [Sample Network Controller configuration script](#)
- [Post-Deployment Steps for Non-Kerberos Deployments](#)

Install the Network Controller server role

You can use this procedure to install the Network Controller server role on a virtual machine (VM).

IMPORTANT

Do not deploy the Network Controller server role on physical hosts. To deploy Network Controller, you must install the Network Controller server role on a Hyper-V virtual machine (VM) that is installed on a Hyper-V host. After you have installed Network Controller on VMs on three different Hyper-V hosts, you must enable the Hyper-V hosts for Software Defined Networking (SDN) by adding the hosts to Network Controller. By doing so, you are enabling the SDN Software Load Balancer to function.

Membership in **Administrators**, or equivalent, is the minimum required to perform this procedure.

NOTE

If you want to use Server Manager instead of Windows PowerShell to install Network Controller, see [Install the Network Controller server role using Server Manager](#)

To install Network Controller by using Windows PowerShell, type the following commands at a Windows PowerShell prompt, and then press ENTER.

```
Install-WindowsFeature -Name NetworkController -IncludeManagementTools
```

Installation of Network Controller requires that you restart the computer. To do so, type the following command, and then press ENTER.

```
Restart-Computer
```

Configure the Network Controller cluster

The Network Controller cluster provides high availability and scalability to the Network Controller application, which you can configure after creating the cluster, and which is hosted on top of the cluster.

NOTE

You can perform the procedures in the following sections either directly on the VM where you installed Network Controller, or you can use the Remote Server Administration Tools for Windows Server 2016 to perform the procedures from a remote computer that is running either Windows Server 2016 or Windows 10. In addition, membership in **Administrators**, or equivalent, is the minimum required to perform this procedure. If the computer or VM upon which you installed Network Controller is joined to a domain, your user account must be a member of **Domain Users**.

You can create a Network Controller cluster by creating a node object and then configuring the cluster.

Create a node object

You need to create a node object for each VM that is a member of the Network Controller cluster.

To create a node object, type the following command at the Windows PowerShell command prompt, and then press ENTER. Ensure that you add values for each parameter that are appropriate for your deployment.

```
New-NetworkControllerNodeObject -Name <string> -Server <String> -FaultDomain <string>-RestInterface <string> [-NodeCertificate <X509Certificate2>]
```

The following table provides descriptions for each parameter of the **New-NetworkControllerNodeObject** command.

| PARAMETER | DESCRIPTION |
|-----------|--|
| Name | The Name parameter specifies the friendly name of the server that you want to add to the cluster |
| Server | The Server parameter specifies the host name, Fully Qualified Domain Name (FQDN), or IP address of the server that you want to add to the cluster. For domain-joined computers, FQDN is required. |

| PARAMETER | DESCRIPTION |
|-----------------|---|
| FaultDomain | <p>The FaultDomain parameter specifies the failure domain for the server that you are adding to the cluster. This parameter defines the servers that might experience failure at the same time as the server that you are adding to the cluster. This failure might be due to shared physical dependencies such as power and networking sources. Fault domains typically represent hierarchies that are related to these shared dependencies, with more servers likely to fail together from a higher point in the fault domain tree. During runtime, Network Controller considers the fault domains in the cluster and attempts to spread out the Network Controller services so that they are in separate fault domains. This process helps ensure, in case of failure of any one fault domain, that the availability of that service and its state is not compromised. Fault domains are specified in a hierarchical format. For example: "Fd:/DC1/Rack1/Host1", where DC1 is the datacenter name, Rack1 is the rack name and Host1 is the name of the host where the node is placed.</p> |
| RestInterface | <p>The RestInterface parameter specifies the name of the interface on the node where the Representational State Transfer (REST) communication is terminated. This Network Controller interface receives Northbound API requests from the network's management layer.</p> |
| NodeCertificate | <p>The NodeCertificate parameter specifies the certificate that Network Controller uses for computer authentication. The certificate is required if you use certificate-based authentication for communication within the cluster; the certificate is also used for encryption of traffic between Network Controller services. The certificate subject name must be same as the DNS name of the node.</p> |

Configure the cluster

To configure the cluster, type the following command at the Windows PowerShell command prompt, and then press ENTER. Ensure that you add values for each parameter that are appropriate for your deployment.

```
Install-NetworkControllerCluster -Node <NetworkControllerNode[]> -ClusterAuthentication <ClusterAuthentication>
[-ManagementSecurityGroup <string>][-DiagnosticLogLocation <string>][-LogLocationCredential <PSCredential>] [-
CredentialEncryptionCertificate <X509Certificate2>][-Credential <PSCredential>][-CertificateThumbprint
<String>] [-UseSSL][-ComputerName <string>][-LogSizeLimitInMBs<UInt32>] [-LogTimeLimitInDays<UInt32>]
```

The following table provides descriptions for each parameter of the **Install-NetworkControllerCluster** command.

| PARAMETER | DESCRIPTION |
|-----------------------|---|
| ClusterAuthentication | <p>The ClusterAuthentication parameter specifies the authentication type that is used for securing the communication between nodes and is also used for encryption of traffic between Network Controller services. The supported values are Kerberos, X509 and None. Kerberos authentication uses domain accounts and can only be used if the Network Controller nodes are domain joined. If you specify X509-based authentication, you must provide a certificate in the NetworkControllerNode object. In addition, you must manually provision the certificate before you run this command.</p> |

| PARAMETER | DESCRIPTION |
|---------------------------------|--|
| ManagementSecurityGroup | The ManagementSecurityGroup parameter specifies the name of the security group that contains users that are allowed to run the management cmdlets from a remote computer. This is only applicable if ClusterAuthentication is Kerberos. You must specify a domain security group and not a security group on the local computer. |
| Node | The Node parameter specifies the list of Network Controller nodes that you created by using the New-NetworkControllerNodeObject command. |
| DiagnosticLogLocation | The DiagnosticLogLocation parameter specifies the share location where the diagnostic logs are periodically uploaded. If you do not specify a value for this parameter, the logs are stored locally on each node. Logs are stored locally in the folder %systemdrive%\Windows\tracing\SDNDiagnostics. Cluster logs are stored locally in the folder %systemdrive%\ProgramData\Microsoft\ServiceFabric\log\Traces. |
| LogLocationCredential | The LogLocationCredential parameter specifies the credentials that are required for accessing the share location where the logs are stored. |
| CredentialEncryptionCertificate | The CredentialEncryptionCertificate parameter specifies the certificate that Network Controller uses to encrypt the credentials that are used to access Network Controller binaries and the LogLocationCredential , if specified. The certificate must be provisioned on all of the Network Controller nodes before you run this command, and the same certificate must be enrolled on all of the cluster nodes. Using this parameter to protect Network Controller binaries and logs is recommended in production environments. Without this parameter, the credentials are stored in clear text and can be misused by any unauthorized user. |
| Credential | This parameter is required only if you are running this command from a remote computer. The Credential parameter specifies a user account that has permission to run this command on the target computer. |
| CertificateThumbprint | This parameter is required only if you are running this command from a remote computer. The CertificateThumbprint parameter specifies the digital public key certificate (X509) of a user account that has permission to run this command on the target computer. |
| UseSSL | This parameter is required only if you are running this command from a remote computer. The UseSSL parameter specifies the Secure Sockets Layer (SSL) protocol that is used to establish a connection to the remote computer. By default, SSL is not used. |
| ComputerName | The ComputerName parameter specifies the Network Controller node on which this command is run. If you do not specify a value for this parameter, the local computer is used by default. |

| PARAMETER | DESCRIPTION |
|--------------------|---|
| LogSizeLimitInMBs | This parameter specifies the maximum log size, in MB, that Network Controller can store. Logs are stored in circular fashion. If DiagnosticLogLocation is provided, the default value of this parameter is 40 GB. If DiagnosticLogLocation is not provided, the logs are stored on the Network Controller nodes and the default value of this parameter is 15 GB. |
| LogTimeLimitInDays | This parameter specifies the duration limit, in days, for which the logs are stored. Logs are stored in circular fashion. The default value of this parameter is 3 days. |

Configure the Network Controller application

To configure the Network Controller application, type the following command at the Windows PowerShell command prompt, and then press ENTER. Ensure that you add values for each parameter that are appropriate for your deployment.

```
Install-NetworkController -Node <NetworkControllerNode[]> -ClientAuthentication <ClientAuthentication> [-ClientCertificateThumbprint <string[]>] [-ClientSecurityGroup <string>] -ServerCertificate <X509Certificate2> [-RESTIPAddress <String>] [-RESTName <String>] [-Credential <PSCredential>][-CertificateThumbprint <String>] [-UseSSL]
```

The following table provides descriptions for each parameter of the **Install-NetworkController** command.

| PARAMETER | DESCRIPTION |
|-----------------------------|--|
| ClientAuthentication | The ClientAuthentication parameter specifies the authentication type that is used for securing the communication between REST and Network Controller. The supported values are Kerberos , X509 and None . Kerberos authentication uses domain accounts and can only be used if the Network Controller nodes are domain joined. If you specify X509-based authentication, you must provide a certificate in the NetworkControllerNode object. In addition, you must manually provision the certificate before you run this command. |
| Node | The Node parameter specifies the list of Network Controller nodes that you created by using the New-NetworkControllerNodeObject command. |
| ClientCertificateThumbprint | This parameter is required only when you are using certificate-based authentication for Network Controller clients. The ClientCertificateThumbprint parameter specifies the thumbprint of the certificate that is enrolled to clients on the Northbound layer. |
| ServerCertificate | The ServerCertificate parameter specifies the certificate that Network Controller uses to prove its identity to clients. The server certificate must include the Server Authentication purpose in Enhanced Key Usage extensions, and must be issued to Network Controller by a CA that is trusted by clients. |

| PARAMETER | DESCRIPTION |
|-----------------------|--|
| RESTIPAddress | <p>You do not need to specify a value for RESTIPAddress with a single node deployment of Network Controller. For multiple-node deployments, the RESTIPAddress parameter specifies the IP address of the REST endpoint in CIDR notation. For example, 192.168.1.10/24. The Subject Name value of ServerCertificate must resolve to the value of the RESTIPAddress parameter. This parameter must be specified for all multiple-node Network Controller deployments when all of the nodes are on the same subnet. If nodes are on different subnets, you must use the RestName parameter instead of using RESTIPAddress.</p> |
| RestName | <p>You do not need to specify a value for RestName with a single node deployment of Network Controller. The only time you must specify a value for RestName is when multiple-node deployments have nodes that are on different subnets. For multiple-node deployments, the RestName parameter specifies the FQDN for the Network Controller cluster.</p> |
| ClientSecurityGroup | <p>The ClientSecurityGroup parameter specifies the name of the Active Directory security group whose members are Network Controller clients. This parameter is required only if you use Kerberos authentication for ClientAuthentication. The security group must contain the accounts from which the REST APIs are accessed, and you must create the security group and add members before running this command.</p> |
| Credential | <p>This parameter is required only if you are running this command from a remote computer. The Credential parameter specifies a user account that has permission to run this command on the target computer.</p> |
| CertificateThumbprint | <p>This parameter is required only if you are running this command from a remote computer. The CertificateThumbprint parameter specifies the digital public key certificate (X509) of a user account that has permission to run this command on the target computer.</p> |
| UseSSL | <p>This parameter is required only if you are running this command from a remote computer. The UseSSL parameter specifies the Secure Sockets Layer (SSL) protocol that is used to establish a connection to the remote computer. By default, SSL is not used.</p> |

After you complete the configuration of the Network Controller application, your deployment of Network Controller is complete.

Network Controller deployment validation

To validate your Network Controller deployment, you can add a credential to the Network Controller and then retrieve the credential.

If you are using Kerberos as the ClientAuthentication mechanism, membership in the **ClientSecurityGroup** that you created is the minimum required to perform this procedure.

Procedure:

1. On a client computer, if you are using Kerberos as the ClientAuthentication mechanism, log on with a user

account that is a member of your **ClientSecurityGroup**.

2. Open Windows PowerShell, type the following commands to add a credential to Network Controller, and then press ENTER. Ensure that you add values for each parameter that are appropriate for your deployment.

```
$cred=New-Object Microsoft.Windows.NetworkController.CredentialProperties  
$cred.type="usernamepassword"  
$cred.username="admin"  
$cred.value="abcd"  
  
New-NetworkControllerCredential -ConnectionUri https://networkcontroller -Properties $cred -ResourceId  
cred1
```

3. To retrieve the credential that you added to Network Controller, type the following command, and then press ENTER. Ensure that you add values for each parameter that are appropriate for your deployment.

```
Get-NetworkControllerCredential -ConnectionUri https://networkcontroller -ResourceId cred1
```

4. Review the command output, which should be similar to the following example output.

```
Tags :  
ResourceRef : /credentials/cred1  
CreatedTime : 1/1/0001 12:00:00 AM  
InstanceId : e16ffe62-a701-4d31-915e-7234d4bc5a18  
Etag : W/"1ec59631-607f-4d3e-ac78-94b0822f3a9d"  
ResourceMetadata :  
ResourceId : cred1  
Properties : Microsoft.Windows.NetworkController.CredentialProperties
```

NOTE

When you run the **Get-NetworkControllerCredential** command, you can assign the output of the command to a variable by using the dot operator to list the properties of the credentials. For example, \$cred.Properties.

Additional Windows PowerShell commands for Network Controller

After you deploy Network Controller, you can use Windows PowerShell commands to manage and modify your deployment. Following are some of the changes that you can make to your deployment.

- Modify Network Controller node, cluster, and application settings
- Remove the Network Controller cluster and application
- Manage Network Controller cluster nodes, including adding, removing, enabling, and disabling nodes.

The following table provides the syntax for Windows PowerShell commands that you can use to accomplish these tasks.

| Task | Command | Syntax |
|--|------------------------------|--|
| Modify Network Controller cluster settings | Set-NetworkControllerCluster | <pre>Set-NetworkControllerCluster [-ManagementSecurityGroup <string>] [-Credential <PSCredential>] [-computerName <string>][-CertificateThumbprint <String>] [-UseSSL]</pre> |

| Task | Command | Syntax |
|--|------------------------------------|---|
| Modify Network Controller application settings | Set-NetworkController | <pre>Set-NetworkController [-ClientAuthentication <ClientAuthentication>] [-Credential <PSCredential>] [-ClientCertificateThumbprint <string[]>] [-ClientSecurityGroup <string>] [-ServerCertificate <X509Certificate2>] [-RestIPAddress <String>] [-ComputerName <String>][-CertificateThumbprint <String>] [-UseSSL]</pre> |
| Modify Network Controller node settings | Set-NetworkControllerNode | <pre>Set-NetworkControllerNode -Name <string> > [-RestInterface <string>] [-NodeCertificate <X509Certificate2>] [-Credential <PSCredential>] [-ComputerName <string>][-CertificateThumbprint <String>] [-UseSSL]</pre> |
| Modify Network Controller diagnostic settings | Set-NetworkControllerDiagnostic | <pre>Set-NetworkControllerDiagnostic [-LogScope <string>] [-DiagnosticLogLocation <string>] [-LogLocationCredential <PSCredential>] [-UseLocalLogLocation] >] [-LogLevel <loglevel>][-LogSizeLimitInMBs <uint32>] [-LogTimeLimitInDays <uint32>] [-Credential <PSCredential>] [-ComputerName <string>][-CertificateThumbprint <String>] [-UseSSL]</pre> |
| Remove the Network Controller application | Uninstall-NetworkController | <pre>Uninstall-NetworkController [-Credential <PSCredential>][-ComputerName <string>] [-CertificateThumbprint <String>] [-UseSSL]</pre> |
| Remove the Network Controller cluster | Uninstall-NetworkControllerCluster | <pre>Uninstall-NetworkControllerCluster [-Credential <PSCredential>][-ComputerName <string>][-CertificateThumbprint <String>] [-UseSSL]</pre> |
| Add a node to the Network Controller cluster | Add-NetworkControllerNode | <pre>Add-NetworkControllerNode -FaultDomain <String> -Name <String> -RestInterface <String> -Server <String> [-CertificateThumbprint <String>] [-ComputerName <String>] [-Credential <PSCredential>] [-Force] [-NodeCertificate <X509Certificate2>] [-PassThru] [-UseSsl]</pre> |
| Disable a Network Controller cluster node | Disable-NetworkControllerNode | <pre>Disable-NetworkControllerNode -Name <String> [-CertificateThumbprint <String>] [-ComputerName <String>] [-Credential <PSCredential>] [-PassThru] [-UseSsl]</pre> |
| Enable a Network Controller cluster node | Enable-NetworkControllerNode | <pre>Enable-NetworkControllerNode -Name <String> [-CertificateThumbprint <String>] [-ComputerName <String>] [-Credential <PSCredential>] [-PassThru] [-UseSsl]</pre> |

| Task | Command | Syntax |
|---|------------------------------|---|
| Remove a Network Controller node from a cluster | Remove-NetworkControllerNode | <pre>Remove-NetworkControllerNode [-CertificateThumbprint <String>] [-ComputerName <String>] [-Credential <PSCredential>] [-Force] [-Name <String>] [-PassThru] [-UseSsl]</pre> |

NOTE

Windows PowerShell commands for Network Controller are in the TechNet Library at [Network Controller Cmdlets](#).

Sample Network Controller configuration script

The following sample configuration script shows how to create a multi-node Network Controller cluster and install the Network Controller application. In addition, the \$cert variable selects a certificate from the local computer certificates store that matches the subject name string "networkController.contoso.com".

```
$a = New-NetworkControllerNodeObject -Name Node1 -Server NCNode1.contoso.com -FaultDomain fd:/rack1/host1 -RestInterface Internal
$b = New-NetworkControllerNodeObject -Name Node2 -Server NCNode2.contoso.com -FaultDomain fd:/rack1/host2 -RestInterface Internal
$c = New-NetworkControllerNodeObject -Name Node3 -Server NCNode3.contoso.com -FaultDomain fd:/rack1/host3 -RestInterface Internal

$cert= get-item Cert:\LocalMachine\My | get-ChildItem | where {$_.Subject -imatch "networkController.contoso.com" }

Install-NetworkControllerCluster -Node @($a,$b,$c) -ClusterAuthentication Kerberos -DiagnosticLogLocation \\share\Diagnostics -ManagementSecurityGroup Contoso\NCManagementAdmins -CredentialEncryptionCertificate $cert
Install-NetworkController -Node @($a,$b,$c) -ClientAuthentication Kerberos -ClientSecurityGroup Contoso\NCRESTClients -ServerCertificate $cert -RestIpAddress 10.0.0.1/24
```

Post-deployment steps For non-Kerberos deployments

If you are not using Kerberos with your Network Controller deployment, you must deploy certificates.

For more information, see [Post-Deployment Steps for Network Controller](#).

Manage SDN

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the topics in this section to manage Software Defined Networking, including tenant workloads and virtual networks.

NOTE

For additional Software Defined Networking documentation, you can use the following library sections.

- [SDN Technologies](#)
- [Plan SDN](#)
- [Deploy SDN](#)
- [Security for SDN](#)
- [Troubleshoot SDN](#)

This section contains the following topics.

- [Manage Tenant Virtual Networks](#)
- [Manage Tenant Workloads](#)
- [Update, Backup, and Restore Software Defined Networking Infrastructure](#)

Manage Tenant Virtual Networks

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the topics in this section to manage Tenant Hyper-V Network Virtualization Virtual Networks after you have deployed Software Defined Networking by using the topic [Deploy a Software Defined Network infrastructure using scripts](#).

This section contains the following topics.

- [Understanding Usage of Virtual Networks and VLANs](#)
- [Use Access Control Lists \(ACLs\) to Manage Datacenter Network Traffic Flow](#)
- [Create, Delete, or Update Tenant Virtual Networks](#)
- [Add a Virtual Gateway to a Tenant Virtual Network](#)
- [Connect container endpoints to a tenant virtual network](#)

Understand the usage of virtual networks and VLANs

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

In this topic, you learn about Hyper-V network virtualization virtual networks and how they differ from virtual local area networks (VLANs). With Hyper-V network virtualization, you create overlay virtual networks, also called virtual networks.

Software Defined Networking (SDN) in Windows Server 2016 is based on programming policy for overlay virtual networks within a Hyper-V Virtual Switch. You can create overlay virtual networks, also called Virtual Networks, with Hyper-V Network Virtualization.

When you deploy Hyper-V Network Virtualization, overlay networks are created by encapsulating the original tenant virtual machine's Layer-2 Ethernet frame with an overlay - or tunnel - header (for example, VXLAN or NVGRE) and Layer-3 IP and Layer-2 Ethernet headers from the underlay (or physical) network. The overlay virtual networks are identified by a 24-bit Virtual Network Identifier (VNI) to maintain tenant traffic isolation and to allow overlapping IP addresses. The VNI is composed of a virtual subnet ID (VSID), logical switch ID, and tunnel ID.

Additionally, each tenant is assigned a routing domain (similar to virtual routing and forwarding - VRF) so that multiple virtual subnet prefixes (each represented by a VNI) can be directly routed to each other. Cross-tenant (or cross routing domain) routing is not supported without going through a gateway.

The physical network on which each tenant's encapsulated traffic is tunneled is represented by a logical network called the provider logical network. This provider logical network consists of one or more subnets, each represented by an IP Prefix and, optionally, a VLAN 802.1q tag.

You can create additional logical networks and subnets for infrastructure purposes to carry management traffic, storage traffic, live migration traffic, etc.

Microsoft SDN does not support the isolation of tenant networks by using VLANs. Tenant isolation is accomplished solely by using Hyper-V Network Virtualization overlay Virtual Networks and encapsulation.

Use access control lists (ACLs) to manage datacenter network traffic flow

9/21/2018 • 4 minutes to read • [Edit Online](#)

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

In this topic, you learn how to configure access control lists (ACLs) to manage data traffic flow using Datacenter Firewall and ACLs on virtual subnets. You enable and configure Datacenter Firewall by creating ACLs that get applied to a virtual subnet or a network interface.

The following examples in this topic demonstrate how to use Windows PowerShell to create these ACLs.

Configure Datacenter Firewall to allow all traffic

Once you deploy SDN, you should test for basic network connectivity in your new environment. To accomplish this, create a rule for Datacenter Firewall that allows all network traffic, without restriction.

Use the entries in the following table to create a set of rules that allow all inbound and outbound network traffic.

| SOURCE IP | DESTINATION IP | PROTOCOL | SOURCE PORT | DESTINATION PORT | DIRECTION | ACTION | PRIORITY |
|-----------|----------------|----------|-------------|------------------|-----------|--------|----------|
| * | * | All | * | * | Inbound | Allow | 100 |
| * | * | All | * | * | Outbound | Allow | 110 |

Example: Create an ACL

In this example, you create an ACL with two rules:

1. **AllowAll_Inbound** - allows all network traffic to pass into the network interface where this ACL is configured.
2. **AllowAllOutbound** - allows all traffic to pass out of the network interface. This ACL, identified by the resource id "AllowAll-1" is now ready to be used in virtual subnets and network interfaces.

The following example script uses Windows PowerShell commands exported from the **NetworkController** module to create this ACL.

```

$ruleproperties = new-object Microsoft.Windows.NetworkController.AclRuleProperties
$ruleproperties.Protocol = "All"
$ruleproperties.SourcePortRange = "0-65535"
$ruleproperties.DestinationPortRange = "0-65535"
$ruleproperties.Action = "Allow"
$ruleproperties.SourceAddressPrefix = "*"
$ruleproperties.DestinationAddressPrefix = "*"
$ruleproperties.Priority = "100"
$ruleproperties.Type = "Inbound"
$ruleproperties.Logging = "Enabled"
$aclrule1 = new-object Microsoft.Windows.NetworkController.AclRule
$aclrule1.Properties = $ruleproperties
$aclrule1.ResourceId = "AllowAll_Inbound"
$ruleproperties = new-object Microsoft.Windows.NetworkController.AclRuleProperties
$ruleproperties.Protocol = "All"
$ruleproperties.SourcePortRange = "0-65535"
$ruleproperties.DestinationPortRange = "0-65535"
$ruleproperties.Action = "Allow"
$ruleproperties.SourceAddressPrefix = "*"
$ruleproperties.DestinationAddressPrefix = "*"
$ruleproperties.Priority = "110"
$ruleproperties.Type = "Outbound"
$ruleproperties.Logging = "Enabled"
$aclrule2 = new-object Microsoft.Windows.NetworkController.AclRule
$aclrule2.Properties = $ruleproperties
$aclrule2.ResourceId = "AllowAll_Outbound"
$acclistproperties = new-object Microsoft.Windows.NetworkController.AccessControlListProperties
$acclistproperties.AclRules = @($aclrule1, $aclrule2)
New-NetworkControllerAccessControlList -ResourceId "AllowAll" -Properties $acclistproperties -ConnectionUri
<NC REST FQDN>

```

NOTE

The Windows PowerShell command reference for Network Controller is located in the topic [Network Controller Cmdlets](#).

Use ACLs to limit traffic on a subnet

In this example, you create an ACL that prevents VMs within the 192.168.0.0/24 subnet from communicating with each other. This type of ACL is useful for limiting the ability of an attacker to spread laterally within the subnet, while still allowing the VMs to receive requests from outside of the subnet, as well as to communicate with other services on other subnets.

| SOURCE IP | DESTINATIO
N IP | PROTOCOL | SOURCE
PORT | DESTINATIO
N PORT | DIRECTION | ACTION | PRIORITY |
|--------------------|--------------------|----------|----------------|----------------------|-----------|--------|----------|
| 192.168.0.1 | * | All | * | * | Inbound | Allow | 100 |
| * | 192.168.0.1 | All | * | * | Outbound | Allow | 101 |
| 192.168.0.0
/24 | * | All | * | * | Inbound | Block | 102 |
| * | 192.168.0.0
/24 | All | * | * | Outbound | Block | 103 |
| * | * | All | * | * | Inbound | Allow | 104 |
| * | * | All | * | * | Outbound | Allow | 105 |

The ACL created by the example script below, identified by the resource id **Subnet-192-168-0-0**, can now be applied to a virtual network subnet that uses the "192.168.0.0/24" subnet address. Any network interface that is attached to that virtual network subnet automatically gets the above ACL rules applied.

The following is an example script using Windows Powershell commands to create this ACL using the Network Controller REST API:

```
import-module networkcontroller
$ncURI = "https://mync.contoso.local"
$aclrules = @()

$ruleproperties = new-object Microsoft.Windows.NetworkController.AclRuleProperties
$ruleproperties.Protocol = "All"
$ruleproperties.SourcePortRange = "0-65535"
$ruleproperties.DestinationPortRange = "0-65535"
$ruleproperties.Action = "Allow"
$ruleproperties.SourceAddressPrefix = "192.168.0.1"
$ruleproperties.DestinationAddressPrefix = "*"
$ruleproperties.Priority = "100"
$ruleproperties.Type = "Inbound"
$ruleproperties.Logging = "Enabled"

$aclrule = new-object Microsoft.Windows.NetworkController.AclRule
$aclrule.Properties = $ruleproperties
$aclrule.ResourceId = "AllowRouter_Inbound"
$aclrules += $aclrule

$ruleproperties = new-object Microsoft.Windows.NetworkController.AclRuleProperties
$ruleproperties.Protocol = "All"
$ruleproperties.SourcePortRange = "0-65535"
$ruleproperties.DestinationPortRange = "0-65535"
$ruleproperties.Action = "Allow"
$ruleproperties.SourceAddressPrefix = "*"
$ruleproperties.DestinationAddressPrefix = "192.168.0.1"
$ruleproperties.Priority = "101"
$ruleproperties.Type = "Outbound"
$ruleproperties.Logging = "Enabled"

$aclrule = new-object Microsoft.Windows.NetworkController.AclRule
$aclrule.Properties = $ruleproperties
$aclrule.ResourceId = "AllowRouter_Outbound"
$aclrules += $aclrule

$ruleproperties = new-object Microsoft.Windows.NetworkController.AclRuleProperties
$ruleproperties.Protocol = "All"
$ruleproperties.SourcePortRange = "0-65535"
$ruleproperties.DestinationPortRange = "0-65535"
$ruleproperties.Action = "Deny"
$ruleproperties.SourceAddressPrefix = "192.168.0.0/24"
$ruleproperties.DestinationAddressPrefix = "*"
$ruleproperties.Priority = "102"
$ruleproperties.Type = "Inbound"
$ruleproperties.Logging = "Enabled"

$aclrule = new-object Microsoft.Windows.NetworkController.AclRule
$aclrule.Properties = $ruleproperties
$aclrule.ResourceId = "DenySubnet_Inbound"
$aclrules += $aclrule

$ruleproperties = new-object Microsoft.Windows.NetworkController.AclRuleProperties
$ruleproperties.Protocol = "All"
$ruleproperties.SourcePortRange = "0-65535"
$ruleproperties.DestinationPortRange = "0-65535"
$ruleproperties.Action = "Deny"
$ruleproperties.SourceAddressPrefix = "*"
$ruleproperties.DestinationAddressPrefix = "192.168.0.0/24"
```

```

$ruleproperties.Priority = "103"
$ruleproperties.Type = "Outbound"
$ruleproperties.Logging = "Enabled"

$aclrule = new-object Microsoft.Windows.NetworkController.AclRule
$aclrule.Properties = $ruleproperties
$aclrule.ResourceId = "DenySubnet_Outbound"

$ruleproperties = new-object Microsoft.Windows.NetworkController.AclRuleProperties
$ruleproperties.Protocol = "All"
$ruleproperties.SourcePortRange = "0-65535"
$ruleproperties.DestinationPortRange = "0-65535"
$ruleproperties.Action = "Allow"
$ruleproperties.SourceAddressPrefix = "*"
$ruleproperties.DestinationAddressPrefix = "*"
$ruleproperties.Priority = "104"
$ruleproperties.Type = "Inbound"
$ruleproperties.Logging = "Enabled"

$aclrule = new-object Microsoft.Windows.NetworkController.AclRule
$aclrule.Properties = $ruleproperties
$aclrule.ResourceId = "AllowAll_Inbound"
$aclrules += $aclrule

$ruleproperties = new-object Microsoft.Windows.NetworkController.AclRuleProperties
$ruleproperties.Protocol = "All"
$ruleproperties.SourcePortRange = "0-65535"
$ruleproperties.DestinationPortRange = "0-65535"
$ruleproperties.Action = "Allow"
$ruleproperties.SourceAddressPrefix = "*"
$ruleproperties.DestinationAddressPrefix = "*"
$ruleproperties.Priority = "105"
$ruleproperties.Type = "Outbound"
$ruleproperties.Logging = "Enabled"

$aclrule = new-object Microsoft.Windows.NetworkController.AclRule
$aclrule.Properties = $ruleproperties
$aclrule.ResourceId = "AllowAll_Outbound"
$aclrules += $aclrule

$acclistproperties = new-object Microsoft.Windows.NetworkController.AccessControlListProperties
$acclistproperties.AclRules = $aclrules

New-NetworkControllerAccessControlList -ResourceId "Subnet-192-168-0-0" -Properties $acclistproperties -
ConnectionUri $ncURI

```

Create, delete, or update tenant virtual networks

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

In this topic, you learn how to create, delete, and update Hyper-V Network Virtualization Virtual Networks after you deploy Software Defined Networking (SDN). Hyper-V Network Virtualization helps you isolate tenant networks so that each tenant network is a separate entity. Each entity has no cross-connection possibility unless you configure public access workloads.

Create a new virtual network

Creating a virtual network for a tenant places it within a unique routing domain on the Hyper-V host. Beneath every virtual network, there is at least one virtual subnet. Virtual Subnets get defined by an IP prefix and reference a previously defined ACL.

The steps to create a new virtual network are:

1. Identify the IP address prefixes from which you want to create the virtual subnets.
2. Identify the logical provider network upon which the tenant traffic is tunneled.
3. Create at least one virtual subnet for each IP prefix that you identified in step 1.
4. (Optional) Add the previously created ACLs to the virtual subnets or add gateway connectivity for tenants.

The following table includes example subnet IDs and prefixes for two fictional tenants. The tenant Fabrikam has two virtual subnets, while the Contoso tenant has three virtual subnets.

| TENANT NAME | VIRTUAL SUBNET ID | VIRTUAL SUBNET PREFIX |
|-------------|-------------------|-----------------------|
| Fabrikam | 5001 | 24.30.1.0/24 |
| Fabrikam | 5002 | 24.30.2.0/20 |
| Contoso | 6001 | 24.30.1.0/24 |
| Contoso | 6002 | 24.30.2.0/24 |
| Contoso | 6003 | 24.30.3.0/24 |

The following example script uses Windows PowerShell commands exported from the **NetworkController** module to create Contoso's virtual network and one subnet:

```

import-module networkcontroller
$URI = "https://ncrest.contoso.local"

#Find the HNV Provider Logical Network

$logicalnetworks = Get-NetworkControllerLogicalNetwork -ConnectionUri $uri
foreach ($ln in $logicalnetworks) {
    if ($ln.Properties.NetworkVirtualizationEnabled -eq "True") {
        $HNVProviderLogicalNetwork = $ln
    }
}

#Find the Access Control List to user per virtual subnet

$acclist = Get-NetworkControllerAccessControlList -ConnectionUri $uri -ResourceId "AllowAll"

#create the Virtual Subnet

$vsubnet = new-object Microsoft.Windows.NetworkController.VirtualSubnet
$vsubnet.ResourceId = "Contoso_WebTier"
$vsubnet.Properties = new-object Microsoft.Windows.NetworkController.VirtualSubnetProperties
$vsubnet.Properties.AccessControlList = $acclist
$vsubnet.Properties.AddressPrefix = "24.30.1.0/24"

#create the Virtual Network

$vnetproperties = new-object Microsoft.Windows.NetworkController.VirtualNetworkProperties
$vnetproperties.AddressSpace = new-object Microsoft.Windows.NetworkController.AddressSpace
$vnetproperties.AddressSpace.AddressPrefixes = @("24.30.1.0/24")
$vnetproperties.LogicalNetwork = $HNVProviderLogicalNetwork
$vnetproperties.Subnets = @($vsubnet)
New-NetworkControllerVirtualNetwork -ResourceId "Contoso_VNet1" -ConnectionUri $uri -Properties $vnetproperties

```

Modify an existing Virtual Network

You can use Windows PowerShell to update an existing Virtual subnet or network.

When you run the following example script, the updated resources are simply PUT to Network Controller with the same resource ID. If your tenant Contoso wants to add a new virtual subnet (24.30.2.0/24) to their virtual network, either you or the Contoso Administrator can use the following script.

```

$acclist = Get-NetworkControllerAccessControlList -ConnectionUri $uri -ResourceId "AllowAll"

$vnet = Get-NetworkControllerVirtualNetwork -ResourceId "Contoso_VNet1" -ConnectionUri $uri

$vnet.properties.AddressSpace.AddressPrefixes += "24.30.2.0/24"

$vsubnet = new-object Microsoft.Windows.NetworkController.VirtualSubnet
$vsubnet.ResourceId = "Contoso_DBTier"
$vsubnet.Properties = new-object Microsoft.Windows.NetworkController.VirtualSubnetProperties
$vsubnet.Properties.AccessControlList = $acclist
$vsubnet.Properties.AddressPrefix = "24.30.2.0/24"

$vnet.properties.Subnets += $vsubnet

New-NetworkControllerVirtualNetwork -ResourceId "Contoso_VNet1" -ConnectionUri $uri -Properties
$vnet.properties

```

Delete a Virtual Network

You can use Windows PowerShell to delete a Virtual Network.

The following Windows PowerShell example deletes a tenant Virtual Network by issuing an HTTP delete to the URI of the Resource ID.

```
Remove-NetworkControllerVirtualNetwork -ResourceId "Contoso_Vnet1" -ConnectionUri $uri
```

Add a virtual gateway to a tenant virtual network

9/21/2018 • 9 minutes to read • [Edit Online](#)

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

Learn how to use Windows PowerShell cmdlets and scripts to provide site-to-site connectivity for your tenant's virtual networks. In this topic, you add tenant virtual gateways to instances of RAS gateway that are members of gateways pools, using Network Controller. RAS gateway supports up to one hundred tenants, depending on the bandwidth used by each tenant. Network Controller automatically determines the best RAS Gateway to use when you deploy a new virtual gateway for your tenants.

Each virtual gateway corresponds to a particular tenant and consists of one or more network connections (site-to-site VPN tunnels) and, optionally, Border Gateway Protocol (BGP) connections. When you provide site-to-site connectivity, your customers can connect their tenant virtual network to an external network, such as a tenant enterprise network, a service provider network, or the Internet.

When you deploy a Tenant Virtual Gateway, you have the following configuration options:

| NETWORK CONNECTION OPTIONS | BGP CONFIGURATION OPTIONS |
|---|--|
| <ul style="list-style-type: none">• IPSec site-to-site virtual private network (VPN)• Generic Routing Encapsulation (GRE)• Layer 3 forwarding | <ul style="list-style-type: none">• BGP router configuration• BGP peer configuration• BGP routing policies configuration |

The Windows PowerShell example scripts and commands in this topic demonstrate how to deploy a tenant virtual gateway on a RAS Gateway with each of these options.

IMPORTANT

Before you run any of the example Windows PowerShell commands and scripts provided, you must change all variable values so that the values are appropriate for your deployment.

1. Verify that the gateway pool object exists in network controller.

```
$uri = "https://ncrest.contoso.com"

# Retrieve the Gateway Pool configuration
$gwPool = Get-NetworkControllerGatewayPool -ConnectionUri $uri

# Display in JSON format
$gwPool | ConvertTo-Json -Depth 2
```

2. Verify that the subnet used for routing packets out of the tenant's virtual network exists in Network Controller. You also retrieve the virtual subnet used for routing between the tenant gateway and virtual network.

```

$uri = "https://ncrest.contoso.com"

# Retrieve the Tenant Virtual Network configuration
$Vnet = Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId "Contoso_Vnet1"

# Display in JSON format
$Vnet | ConvertTo-Json -Depth 4

# Retrieve the Tenant Virtual Subnet configuration
$RoutingSubnet = Get-NetworkControllerVirtualSubnet -ConnectionUri $uri -ResourceId "Contoso_WebTier" -
VirtualNetworkID $vnet.ResourceId

# Display in JSON format
$RoutingSubnet | ConvertTo-Json -Depth 4

```

3. Create a new object for the tenant virtual gateway and then update the gateway pool reference. You also specify the virtual subnet used for routing between the gateway and virtual network. After specifying the virtual subnet you update the rest of the virtual gateway object properties and then add the new virtual gateway for the tenant.

```

# Create a new object for Tenant Virtual Gateway
$VirtualGWProperties = New-Object Microsoft.Windows.NetworkController.VirtualGatewayProperties

# Update Gateway Pool reference
$VirtualGWProperties.GatewayPools = @()
$VirtualGWProperties.GatewayPools += $gwPool

# Specify the Virtual Subnet that is to be used for routing between the gateway and Virtual Network
$VirtualGWProperties.GatewaySubnets = @()
$VirtualGWProperties.GatewaySubnets += $RoutingSubnet

# Update the rest of the Virtual Gateway object properties
$VirtualGWProperties.RoutingType = "Dynamic"
$VirtualGWProperties.NetworkConnections = @()
$VirtualGWProperties.BgpRouters = @()

# Add the new Virtual Gateway for tenant
$virtualGW = New-NetworkControllerVirtualGateway -ConnectionUri $uri -ResourceId "Contoso_VirtualGW" -
Properties $VirtualGWProperties -Force

```

4. Create a site-to-site VPN connection with IPsec, GRE, or Layer 3 (L3) forwarding.

TIP

Optionally, you can combine all the previous steps and configure a tenant virtual gateway with all three connection options. For more details, see [Configure a gateway with all three connection types \(IPsec, GRE, L3\) and BGP](#).

IPsec VPN site-to-site network connection

```

# Create a new object for Tenant Network Connection
$nwConnectionProperties = New-Object Microsoft.Windows.NetworkController.NetworkConnectionProperties

# Update the common object properties
$nwConnectionProperties.ConnectionType = "IPSec"
$nwConnectionProperties.OutboundKiloBitsPerSecond = 10000
$nwConnectionProperties.InboundKiloBitsPerSecond = 10000

# Update specific properties depending on the Connection Type
$nwConnectionProperties.IpSecConfiguration = New-Object
Microsoft.Windows.NetworkController.IpSecConfiguration
$nwConnectionProperties.IpSecConfiguration.AuthenticationMethod = "PSK"
$nwConnectionProperties.IpSecConfiguration.SharedSecret = "P@ssw0rd"

$nwConnectionProperties.IpSecConfiguration.QuickMode = New-Object
Microsoft.Windows.NetworkController.QuickMode
$nwConnectionProperties.IpSecConfiguration.QuickMode.PerfectForwardSecrecy = "PFS2048"
$nwConnectionProperties.IpSecConfiguration.QuickMode.AuthenticationTransformationConstant = "SHA256128"
$nwConnectionProperties.IpSecConfiguration.QuickMode.CipherTransformationConstant = "DES3"
$nwConnectionProperties.IpSecConfiguration.QuickMode.SALifeTimeSeconds = 1233
$nwConnectionProperties.IpSecConfiguration.QuickMode.IdleDisconnectSeconds = 500
$nwConnectionProperties.IpSecConfiguration.QuickMode.SALifeTimeKiloBytes = 2000

$nwConnectionProperties.IpSecConfiguration.MainMode = New-Object
Microsoft.Windows.NetworkController.MainMode
$nwConnectionProperties.IpSecConfiguration.MainMode.DiffieHellmanGroup = "Group2"
$nwConnectionProperties.IpSecConfiguration.MainMode.IntegrityAlgorithm = "SHA256"
$nwConnectionProperties.IpSecConfiguration.MainMode.EncryptionAlgorithm = "AES256"
$nwConnectionProperties.IpSecConfiguration.MainMode.SALifeTimeSeconds = 1234
$nwConnectionProperties.IpSecConfiguration.MainMode.SALifeTimeKiloBytes = 2000

# L3 specific configuration (leave blank for IPSec)
$nwConnectionProperties.IPAddresses = @()
$nwConnectionProperties.PeerIPAddresses = @()

# Update the IPv4 Routes that are reachable over the site-to-site VPN Tunnel
$nwConnectionProperties.Routes = @()
$ipv4Route = New-Object Microsoft.Windows.NetworkController.RouteInfo
$ipv4Route.DestinationPrefix = "14.1.10.1/32"
$ipv4Route.metric = 10
$nwConnectionProperties.Routes += $ipv4Route

# Tunnel Destination (Remote Endpoint) Address
$nwConnectionProperties.DestinationIPAddress = "10.127.134.121"

# Add the new Network Connection for the tenant
New-NetworkControllerVirtualGatewayNetworkConnection -ConnectionUri $uri -VirtualGatewayId
$virtualGW.ResourceId -ResourceId "Contoso_IPSecGW" -Properties $nwConnectionProperties -Force

```

GRE VPN site-to-site network connection

```

# Create a new object for the Tenant Network Connection
$nwConnectionProperties = New-Object Microsoft.Windows.NetworkController.NetworkConnectionProperties

# Update the common object properties
$nwConnectionProperties.ConnectionType = "GRE"
$nwConnectionProperties.OutboundKiloBitsPerSecond = 10000
$nwConnectionProperties.InboundKiloBitsPerSecond = 10000

# Update specific properties depending on the Connection Type
$nwConnectionProperties.GreConfiguration = New-Object
Microsoft.Windows.NetworkController.GreConfiguration
$nwConnectionProperties.GreConfiguration.GreKey = 1234

# Update the IPv4 Routes that are reachable over the site-to-site VPN Tunnel
$nwConnectionProperties.Routes = @()
$ipv4Route = New-Object Microsoft.Windows.NetworkController.RouteInfo
$ipv4Route.DestinationPrefix = "14.2.20.1/32"
$ipv4Route.metric = 10
$nwConnectionProperties.Routes += $ipv4Route

# Tunnel Destination (Remote Endpoint) Address
$nwConnectionProperties.DestinationIPAddress = "10.127.134.122"

# L3 specific configuration (leave blank for GRE)
$nwConnectionProperties.L3Configuration = New-Object Microsoft.Windows.NetworkController.L3Configuration
$nwConnectionProperties.IPAddresses = @()
$nwConnectionProperties.PeerIPAddresses = @()

# Add the new Network Connection for the tenant
New-NetworkControllerVirtualGatewayNetworkConnection -ConnectionUri $uri -VirtualGatewayId
$virtualGW.ResourceId -ResourceId "Contoso_GreGW" -Properties $nwConnectionProperties -Force

```

L3 forwarding network connection

For a L3 forwarding network connection to work properly, you must configure a corresponding logical network.

- Configure a logical network for the L3 forwarding Network Connection.

```

# Create a new object for the Logical Network to be used for L3 Forwarding
$lnProperties = New-Object Microsoft.Windows.NetworkController.LogicalNetworkProperties

$lnProperties.NetworkVirtualizationEnabled = $false
$lnProperties.Subnets = @()

# Create a new object for the Logical Subnet to be used for L3 Forwarding and update properties
$logicalsubnet = New-Object Microsoft.Windows.NetworkController.LogicalSubnet
$logicalsubnet.ResourceId = "Contoso_L3_Subnet"
$logicalsubnet.Properties = New-Object Microsoft.Windows.NetworkController.LogicalSubnetProperties
$logicalsubnet.Properties.VlanID = 1001
$logicalsubnet.Properties.AddressPrefix = "10.127.134.0/25"
$logicalsubnet.Properties.DefaultGateways = "10.127.134.1"

$lnProperties.Subnets += $logicalsubnet

# Add the new Logical Network to Network Controller
$vlanNetwork = New-NetworkControllerLogicalNetwork -ConnectionUri $uri -ResourceId
"Contoso_L3_Network" -Properties $lnProperties -Force

```

- Create a Network Connection JSON Object and add it to Network Controller.

```

# Create a new object for the Tenant Network Connection
$nwConnectionProperties = New-Object
Microsoft.Windows.NetworkController.NetworkConnectionProperties

# Update the common object properties
$nwConnectionProperties.ConnectionType = "L3"
$nwConnectionProperties.OutboundKiloBitsPerSecond = 10000
$nwConnectionProperties.InboundKiloBitsPerSecond = 10000

# GRE specific configuration (leave blank for L3)
$nwConnectionProperties.GreConfiguration = New-Object
Microsoft.Windows.NetworkController.GreConfiguration

# Update specific properties depending on the Connection Type
$nwConnectionProperties.L3Configuration = New-Object
Microsoft.Windows.NetworkController.L3Configuration
$nwConnectionProperties.L3Configuration.VlanSubnet = $vlanNetwork.properties.Subnets[0]

$nwConnectionProperties.IPAddresses = @()
$localIPAddress = New-Object Microsoft.Windows.NetworkController.CidrIPAddress
$localIPAddress.IPAddress = "10.127.134.55"
$localIPAddress.PrefixLength = 25
$nwConnectionProperties.IPAddresses += $localIPAddress

$nwConnectionProperties.PeerIPAddresses = @("10.127.134.65")

# Update the IPv4 Routes that are reachable over the site-to-site VPN Tunnel
$nwConnectionProperties.Routes = @()
$ipv4Route = New-Object Microsoft.Windows.NetworkController.RouteInfo
$ipv4Route.DestinationPrefix = "14.2.20.1/32"
$ipv4Route.metric = 10
$nwConnectionProperties.Routes += $ipv4Route

# Add the new Network Connection for the tenant
New-NetworkControllerVirtualGatewayNetworkConnection -ConnectionUri $uri -VirtualGatewayId
$virtualGW.ResourceId -ResourceId "Contoso_L3GW" -Properties $nwConnectionProperties -Force

```

5. Configure the gateway as a BGP router and add it to Network Controller.

a. Add a BGP router for the tenant.

```

# Create a new object for the Tenant BGP Router
$bgpRouterProperties = New-Object Microsoft.Windows.NetworkController.VGwBgpRouterProperties

# Update the BGP Router properties
$bgpRouterProperties.ExtAsNumber = "0.64512"
$bgpRouterProperties.RouterId = "192.168.0.2"
$bgpRouterProperties.RouterIP = @("192.168.0.2")

# Add the new BGP Router for the tenant
$bgpRouter = New-NetworkControllerVirtualGatewayBgpRouter -ConnectionUri $uri -VirtualGatewayId
$virtualGW.ResourceId -ResourceId "Contoso_BgpRouter1" -Properties $bgpRouterProperties -Force

```

b. Add a BGP Peer for this tenant, corresponding to the site-to-site VPN Network Connection added above.

```

# Create a new object for Tenant BGP Peer
$bgpPeerProperties = New-Object Microsoft.Windows.NetworkController.VGwBgpPeerProperties

# Update the BGP Peer properties
$bgpPeerProperties.PeerIpAddress = "14.1.10.1"
$bgpPeerProperties.AsNumber = 64521
$bgpPeerProperties.ExtAsNumber = "0.64521"

# Add the new BGP Peer for tenant
New-NetworkControllerVirtualGatewayBgpPeer -ConnectionUri $uri -VirtualGatewayId
$virtualGW.ResourceId -BgpRouterName $bgpRouter.ResourceId -ResourceId "Contoso_IPSec_Peer" -
Properties $bgpPeerProperties -Force

```

(Optional step) Configure a gateway with all three connection types (IPsec, GRE, L3) and BGP

Optionally, you can combine all previous steps and configure a tenant virtual gateway with all three connection options:

```

# Create a new Virtual Gateway Properties type object
$VirtualGWProperties = New-Object Microsoft.Windows.NetworkController.VirtualGatewayProperties

# Update GatewayPool reference
$VirtualGWProperties.GatewayPools = @()
$VirtualGWProperties.GatewayPools += $gwPool

# Specify the Virtual Subnet that is to be used for routing between GW and VNET
$VirtualGWProperties.GatewaySubnets = @()
$VirtualGWProperties.GatewaySubnets += $RoutingSubnet

# Update some basic properties
$VirtualGWProperties.RoutingType = "Dynamic"

# Update Network Connection object(s)
$VirtualGWProperties.NetworkConnections = @()

# IPsec Connection configuration
$ipSecConnection = New-Object Microsoft.Windows.NetworkController.NetworkConnection
$ipSecConnection.ResourceId = "Contoso_IPSecGW"
$ipSecConnection.Properties = New-Object Microsoft.Windows.NetworkController.NetworkConnectionProperties
$ipSecConnection.Properties.ConnectionType = "IPSec"
$ipSecConnection.Properties.OutboundKiloBitsPerSecond = 10000
$ipSecConnection.Properties.InboundKiloBitsPerSecond = 10000

$ipSecConnection.Properties.IpSecConfiguration = New-Object
Microsoft.Windows.NetworkController.IpSecConfiguration

$ipSecConnection.Properties.IpSecConfiguration.AuthenticationMethod = "PSK"
$ipSecConnection.Properties.IpSecConfiguration.SharedSecret = "P@ssw0rd"

$ipSecConnection.Properties.IpSecConfiguration.QuickMode = New-Object
Microsoft.Windows.NetworkController.QuickMode

$ipSecConnection.Properties.IpSecConfiguration.QuickMode.PerfectForwardSecrecy = "PFS2048"
$ipSecConnection.Properties.IpSecConfiguration.QuickMode.AuthenticationTransformationConstant = "SHA256128"
$ipSecConnection.Properties.IpSecConfiguration.QuickMode.CipherTransformationConstant = "DES3"
$ipSecConnection.Properties.IpSecConfiguration.QuickMode.SALifeTimeSeconds = 1233
$ipSecConnection.Properties.IpSecConfiguration.QuickMode.IdleDisconnectSeconds = 500
$ipSecConnection.Properties.IpSecConfiguration.QuickMode.SALifeTimeKiloBytes = 2000

$ipSecConnection.Properties.IpSecConfiguration.MainMode = New-Object
Microsoft.Windows.NetworkController.MainMode

$ipSecConnection.Properties.IpSecConfiguration.MainMode.DiffieHellmanGroup = "Group2"

```

```

$ipSecConnection.Properties.IpSecConfiguration.MainMode.IntegrityAlgorithm = "SHA256"
$ipSecConnection.Properties.IpSecConfiguration.MainMode.EncryptionAlgorithm = "AES256"
$ipSecConnection.Properties.IpSecConfiguration.MainMode.SALifeTimeSeconds = 1234
$ipSecConnection.Properties.IpSecConfiguration.MainMode.SALifeTimeKiloBytes = 2000

$ipSecConnection.Properties.IPAddresses = @()
$ipSecConnection.Properties.PeerIPAddresses = @()

$ipSecConnection.Properties.Routes = @()

$ipv4Route = New-Object Microsoft.Windows.NetworkController.RouteInfo
$ipv4Route.DestinationPrefix = "14.1.10.1/32"
$ipv4Route.metric = 10
$ipSecConnection.Properties.Routes += $ipv4Route

$ipSecConnection.Properties.DestinationIPAddress = "10.127.134.121"

# GRE Connection configuration
$greConnection = New-Object Microsoft.Windows.NetworkController.NetworkConnection
$greConnection.ResourceId = "Contoso_GreGW"

$greConnection.Properties = New-Object Microsoft.Windows.NetworkController.NetworkConnectionProperties
$greConnection.Properties.ConnectionType = "GRE"
$greConnection.Properties.OutboundKiloBitsPerSecond = 10000
$greConnection.Properties.InboundKiloBitsPerSecond = 10000

$greConnection.Properties.GreConfiguration = New-Object Microsoft.Windows.NetworkController.GreConfiguration
$greConnection.Properties.GreConfiguration.GreKey = 1234

$greConnection.Properties.IPAddresses = @()
$greConnection.Properties.PeerIPAddresses = @()

$greConnection.Properties.Routes = @()

$ipv4Route = New-Object Microsoft.Windows.NetworkController.RouteInfo
$ipv4Route.DestinationPrefix = "14.2.20.1/32"
$ipv4Route.metric = 10
$greConnection.Properties.Routes += $ipv4Route

$greConnection.Properties.DestinationIPAddress = "10.127.134.122"

$greConnection.Properties.L3Configuration = New-Object Microsoft.Windows.NetworkController.L3Configuration

# L3 Forwarding connection configuration
$l3Connection = New-Object Microsoft.Windows.NetworkController.NetworkConnection
$l3Connection.ResourceId = "Contoso_L3GW"

$l3Connection.Properties = New-Object Microsoft.Windows.NetworkController.NetworkConnectionProperties
$l3Connection.Properties.ConnectionType = "L3"
$l3Connection.Properties.OutboundKiloBitsPerSecond = 10000
$l3Connection.Properties.InboundKiloBitsPerSecond = 10000

$l3Connection.Properties.GreConfiguration = New-Object Microsoft.Windows.NetworkController.GreConfiguration
$l3Connection.Properties.L3Configuration = New-Object Microsoft.Windows.NetworkController.L3Configuration
$l3Connection.Properties.L3Configuration.VlanSubnet = $vlanNetwork.properties.Subnets[0]

$l3Connection.Properties.IPAddresses = @()
$localIPAddress = New-Object Microsoft.Windows.NetworkController.CidrIPAddress
$localIPAddress.IPAddress = "10.127.134.55"
$localIPAddress.PrefixLength = 25
$l3Connection.Properties.IPAddresses += $localIPAddress

$l3Connection.Properties.PeerIPAddresses = @("10.127.134.65")

$l3Connection.Properties.Routes = @()
$ipv4Route = New-Object Microsoft.Windows.NetworkController.RouteInfo
$ipv4Route.DestinationPrefix = "14.2.20.1/32"
$ipv4Route.metric = 10
$l3Connection.Properties.Routes += $ipv4Route

```

```

# Update BGP Router Object
$VirtualGWProperties.BgpRouters = @()

$bgpRouter = New-Object Microsoft.Windows.NetworkController.VGwBgpRouter
$bgpRouter.ResourceId = "Contoso_BgpRouter1"
$bgpRouter.Properties = New-Object Microsoft.Windows.NetworkController.VGwBgpRouterProperties

$bgpRouter.Properties.ExtAsNumber = "0.64512"
$bgpRouter.Properties.RouterId = "192.168.0.2"
$bgpRouter.Properties.RouterIP = @("192.168.0.2")

$bgpRouter.Properties.BgpPeers = @()

# Create BGP Peer Object(s)
# BGP Peer for IPSec Connection
$bgpPeer_IPSec = New-Object Microsoft.Windows.NetworkController.VGwBgpPeer
$bgpPeer_IPSec.ResourceId = "Contoso_IPSec_Peer"

$bgpPeer_IPSec.Properties = New-Object Microsoft.Windows.NetworkController.VGwBgpPeerProperties
$bgpPeer_IPSec.Properties.PeerIpAddress = "14.1.10.1"
$bgpPeer_IPSec.Properties.AsNumber = 64521
$bgpPeer_IPSec.Properties.ExtAsNumber = "0.64521"

$bgpRouter.Properties.BgpPeers += $bgpPeer_IPSec

# BGP Peer for GRE Connection
$bgpPeer_Gre = New-Object Microsoft.Windows.NetworkController.VGwBgpPeer
$bgpPeer_Gre.ResourceId = "Contoso_Gre_Peer"

$bgpPeer_Gre.Properties = New-Object Microsoft.Windows.NetworkController.VGwBgpPeerProperties
$bgpPeer_Gre.Properties.PeerIpAddress = "14.2.20.1"
$bgpPeer_Gre.Properties.AsNumber = 64522
$bgpPeer_Gre.Properties.ExtAsNumber = "0.64522"

$bgpRouter.Properties.BgpPeers += $bgpPeer_Gre

# BGP Peer for L3 Connection
$bgpPeer_L3 = New-Object Microsoft.Windows.NetworkController.VGwBgpPeer
$bgpPeer_L3.ResourceId = "Contoso_L3_Peer"

$bgpPeer_L3.Properties = New-Object Microsoft.Windows.NetworkController.VGwBgpPeerProperties
$bgpPeer_L3.Properties.PeerIpAddress = "14.3.30.1"
$bgpPeer_L3.Properties.AsNumber = 64523
$bgpPeer_L3.Properties.ExtAsNumber = "0.64523"

$bgpRouter.Properties.BgpPeers += $bgpPeer_L3

$VirtualGWProperties.BgpRouters += $bgpRouter

# Finally Add the new Virtual Gateway for tenant
New-NetworkControllerVirtualGateway -ConnectionUri $uri -ResourceId "Contoso_VirtualGW" -Properties
$VirtualGWProperties -Force

```

Modify a gateway for a virtual network

Retrieve the configuration for the component and store it in a variable

```
$nwConnection = Get-NetworkControllerVirtualGatewayNetworkConnection -ConnectionUri $uri -VirtualGatewayId
"Contoso_VirtualGW" -ResourceId "Contoso_IPSecGW"
```

Navigate the variable structure to reach the required property and set it to the updates value

```
$nwConnection.properties.IpSecConfiguration.SharedSecret = "C0mplexP@ssW0rd"
```

Add the modified configuration to replace the older configuration on Network Controller

```
New-NetworkControllerVirtualGatewayNetworkConnection -ConnectionUri $uri -VirtualGatewayId "Contoso_VirtualGW" -ResourceId $nwConnection.ResourceId -Properties $nwConnection.Properties -Force
```

Remove a gateway from a virtual network

You can use the following Windows PowerShell commands to remove either individual gateway features or the entire gateway.

Remove a network connection

```
Remove-NetworkControllerVirtualGatewayNetworkConnection -ConnectionUri $uri -VirtualGatewayId "Contoso_VirtualGW" -ResourceId "Contoso_IPSecGW" -Force
```

Remove a BGP peer

```
Remove-NetworkControllerVirtualGatewayBgpPeer -ConnectionUri $uri -VirtualGatewayId "Contoso_VirtualGW" -BgpRouterName "Contoso_BgpRouter1" -ResourceId "Contoso_IPSec_Peer" -Force
```

Remove a BGP router

```
Remove-NetworkControllerVirtualGatewayBgpRouter -ConnectionUri $uri -VirtualGatewayId "Contoso_VirtualGW" -ResourceId "Contoso_BgpRouter1" -Force
```

Remove a gateway

```
Remove-NetworkControllerVirtualGateway -ConnectionUri $uri -ResourceId "Contoso_VirtualGW" -Force
```

Connect container endpoints to a tenant virtual network

9/1/2018 • 4 minutes to read • [Edit Online](#)

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

In this topic, we show you how to connect container endpoints to an existing tenant virtual network created through SDN. You use the *l2bridge* (and optionally *l2tunnel*) network driver available with the Windows libnetwork plugin for Docker to create a container network on the tenant VM.

In the [Container network drivers](#) topic, we discussed the multiple network drivers are available through Docker on Windows. For SDN, use the *l2bridge* and *l2tunnel* drivers. For both drivers, each container endpoint is in the same virtual subnet as the container host (tenant) virtual machine.

The Host Networking Service (HNS), through the private cloud plugin, dynamically assigns the IP addresses for container endpoints. The container endpoints have unique IP addresses but share the same MAC address of the container host (tenant) virtual machine due to Layer-2 address translation.

Network policy (ACLs, encapsulation, and QoS) for these container endpoints are enforced in the physical Hyper-V host as received by the Network Controller and defined in upper-layer management systems.

The difference between the *l2bridge* and *l2tunnel* drivers are:

| L2BRIDGE | L2TUNNEL |
|---|---|
| <p>Container endpoints that reside on:</p> <ul style="list-style-type: none">• The same container host virtual machine and on the same subnet have all network traffic bridged within the Hyper-V virtual switch.• Different container host VMs or on different subnets have their traffic forwarded to the physical Hyper-V host. <p>Network policy does not get enforced since network traffic between containers on the same host and in the same subnet do not flow to the physical host. Network policy applies only to cross-host or cross-subnet container network traffic.</p> | <p><i>ALL</i> network traffic between two container endpoints is forwarded to the physical Hyper-V host regardless of host or subnet. Network policy applies to both cross-subnet and cross-host network traffic.</p> |

NOTE

These networking modes do not work for connecting windows container endpoints to a tenant virtual network in Azure public cloud.

Prerequisites

- A deployed SDN infrastructure with the Network Controller.
- A tenant virtual network has been created.
- A deployed tenant virtual machine with the Windows Container feature enabled, Docker installed, and Hyper-V feature enabled. The Hyper-V feature is required to install several binaries for l2bridge and l2tunnel networks.

```
# To install HyperV feature without checks for nested virtualization  
dism /Online /Enable-Feature /FeatureName:Microsoft-Hyper-V /All
```

NOTE

Nested virtualization and exposing virtualization extensions is not required unless using Hyper-V Containers.

Workflow

1. Add multiple IP configurations to an existing VM NIC resource through Network Controller (Hyper-V Host)
2. Enable the network proxy on the host to allocate CA IP Addresses for container endpoints (Hyper-V Host)
3. Install the private cloud plug-in to assign CA IP addresses to container endpoints (Container Host VM)
4. Create an *l2bridge* or *l2tunnel* network using docker (Container Host VM)

NOTE

Multiple IP configurations is not supported on VM NIC resources created through System Center Virtual Machine Manager. It is recommended for these deployment types that you create the VM NIC resource out of band using Network Controller PowerShell.

1. Add Multiple IP Configurations

In this step, we assume the VM NIC of the tenant virtual machine has one IP configuration with IP address of 192.168.1.9 and is attached to a VNet Resource ID of 'VNet1' and VM Subnet Resource of 'Subnet1' in the 192.168.1.0/24 IP subnet. We add 10 IP addresses for containers from 192.168.1.101 - 192.168.1.110.

```

Import-Module NetworkController

# Specify Network Controller REST IP or FQDN
$uri = "<NC REST IP or FQDN>"
$vnetResourceId = "VNet1"
$vsubnetResourceId = "Subnet1"

$vmnic= Get-NetworkControllerNetworkInterface -ConnectionUri $uri | where
{$_.properties.IpConfigurations.Properties.PrivateIPAddress -eq "192.168.1.9" }
$vmsubnet = Get-NetworkControllerVirtualSubnet -VirtualNetworkId $vnetResourceId -ResourceId $vsubnetResourceId
-ConnectionUri $uri

# For this demo, we will assume an ACL has already been defined; any ACL can be applied here
$allowallacl = Get-NetworkControllerAccessControlList -ConnectionUri $uri -ResourceId "AllowAll"

foreach ($i in 1..10)
{
    $newipconfig = new-object Microsoft.Windows.NetworkController.NetworkInterfaceIpConfiguration
    $props = new-object Microsoft.Windows.NetworkController.NetworkInterfaceIpConfigurationProperties

    $resourceid = "IP_192_168_1_1"
    if ($i -eq 10)
    {
        $resourceid += "10"
        $ipstr = "192.168.1.110"
    }
    else
    {
        $resourceid += "0$i"
        $ipstr = "192.168.1.10$i"
    }

    $newipconfig.ResourceId = $resourceid
    $props.PrivateIPAddress = $ipstr

    $props.PrivateIPAllocationMethod = "Static"
    $props.Subnet = new-object Microsoft.Windows.NetworkController.Subnet
    $props.Subnet.ResourceRef = $vmsubnet.ResourceRef
    $props.AccessControlList = new-object Microsoft.Windows.NetworkController.AccessControlList
    $props.AccessControlList.ResourceRef = $allowallacl.ResourceRef

    $newipconfig.Properties = $props
    $vmnic.Properties.IpConfigurations += $newipconfig
}

New-NetworkControllerNetworkInterface -ResourceId $vmnic.ResourceId -Properties $vmnic.Properties -
ConnectionUri $uri

```

2. Enable the network proxy

In this step, you enable the network proxy to allocate multiple IP addresses for the container host virtual machine.

To enable the network proxy, run the [ConfigureMCNP.ps1](#) script on the **Hyper-V Host** hosting the container host (tenant) virtual machine.

```
PS C:\> ConfigureMCNP.ps1
```

3. Install the Private Cloud plug-in

In this step, you install a plug-in to allow the HNS to communicate with the network proxy on the Hyper-V Host.

To install the plug-in, run the [InstallPrivateCloudPlugin.ps1](#) script inside the **container host (tenant) virtual machine**.

```
PS C:\> InstallPrivateCloudPlugin.ps1
```

4. Create an *l2bridge* Container Network

In this step, you use the `docker network create` command on the **container host (tenant) virtual machine** to create an *l2bridge* network.

```
# Create the container network
C:\> docker network create -d l2bridge --subnet="192.168.1.0/24" --gateway="192.168.1.1"
MyContainerOverlayNetwork

# Attach a container to the MyContainerOverlayNetwork
C:\> docker run -it --network=MyContainerOverlayNetwork <image> <cmd>
```

NOTE

Static IP assignment is not supported with *l2bridge* or *l2tunnel* container networks when used with the Microsoft SDN Stack.

More information

For more details about deploying an SDN infrastructure, see [Deploy a Software Defined Network Infrastructure](#).

Configure Encryption for a Virtual Subnet

9/1/2018 • 4 minutes to read • [Edit Online](#)

Applies to: Windows Server

Virtual network encryption allows for encryption of virtual network traffic between VMs that communicate with each other within subnets marked as 'Encryption Enabled.' It also utilizes Datagram Transport Layer Security (DTLS) on the virtual subnet to encrypt packets. DTLS protects against eavesdropping, tampering, and forgery by anyone with access to the physical network.

Virtual network encryption requires:

- Encryption certificates installed on each of the SDN-enabled Hyper-V hosts.
- A credential object in the Network Controller referencing the thumbprint of that certificate.
- Configuration on each of the Virtual Networks contain subnets that require encryption.

Once you enable encryption on a subnet, all network traffic within that subnet is encrypted automatically, in addition to any application-level encryption that may also take place. Traffic that crosses between subnets, even if marked as encrypted, is sent unencrypted automatically. Any traffic that crosses the virtual network boundary also gets sent unencrypted.

NOTE

When communicating with another VM on the same subnet, whether its currently connected or connected at a later time, the traffic gets encrypted automatically.

TIP

If you must restrict applications to only communicate on the encrypted subnet, you can use Access Control Lists (ACLs) only to allow communication within the current subnet. For more information, see [Use Access Control Lists \(ACLs\) to Manage Datacenter Network Traffic Flow](#).

Step 1. Create the Encryption Certificate

Each host must have an encryption certificate installed. You can use the same certificate for all tenants or generate a unique one for each tenant.

1. Generate the certificate

```

$subjectName = "EncryptedVirtualNetworks"
$cryptographicProviderName = "Microsoft Base Cryptographic Provider v1.0";
[int] $privateKeyLength = 1024;
$sslServerOidString = "1.3.6.1.5.5.7.3.1";
$sslClientOidString = "1.3.6.1.5.5.7.3.2";
[int] $validityPeriodInYear = 5;

$name = new-object -com "X509Enrollment.CX500DistinguishedName.1"
$name.Encode("CN=" + $subjectName, 0)

#Generate Key
$key = new-object -com "X509Enrollment.CX509PrivateKey.1"
$key.ProviderName = $cryptographicProviderName
$keyKeySpec = 1 #X509KeySpec.XCN_AT_KEYEXCHANGE
$key.Length = $privateKeyLength
$key.MachineContext = 1
$key.ExportPolicy = 0x2 #X509PrivateKeyExportFlags.XCN_NCRYPT_ALLOW_EXPORT_FLAG
$key.Create()

#Configure Eku
$serverauthoid = new-object -com "X509Enrollment.CObjectID.1"
$serverauthoid.InitializeFromValue($sslServerOidString)
$clientauthoid = new-object -com "X509Enrollment.CObjectID.1"
$clientauthoid.InitializeFromValue($sslClientOidString)
$ekuoids = new-object -com "X509Enrollment.CObjectIDs.1"
$ekuoids.add($serverauthoid)
$ekuoids.add($clientauthoid)
$ekuext = new-object -com "X509Enrollment.CX509ExtensionEnhancedKeyUsage.1"
$ekuext.InitializeEncode($ekuoids)

# Set the hash algorithm to sha512 instead of the default sha1
$hashAlgorithmObject = New-Object -ComObject X509Enrollment.CObjectID
$hashAlgorithmObject.InitializeAlgorithmName( $objectIDGroupId.XCN_CRYPT_HASH_ALG_OID_GROUP_ID,
$objectIDPublicKeyFlags.XCN_CRYPT_OID_INFO_PUBKEY_ANY, $algorithmFlags.AlgorithmFlagsNone, "SHA512")

#Request Certificate
$cert = new-object -com "X509Enrollment.CX509CertificateRequestCertificate.1"

$cert.InitializeFromPrivateKey(2, $key, "")
$cert.Subject = $name
$cert.Issuer = $cert.Subject
$cert.NotBefore = (get-date).ToUniversalTime()
$cert.NotAfter = $cert.NotBefore.AddYears($validityPeriodInYear);
$cert.X509Extensions.Add($ekuext)
$cert.HashAlgorithm = $hashAlgorithmObject
$cert.Encode()

$enrollment = new-object -com "X509Enrollment.CX509Enrollment.1"
$enrollment.InitializeFromRequest($cert)
$certdata = $enrollment.CreateRequest(0)
$enrollment.InstallResponse(2, $certdata, 0, "")

```

After running the script, a new certificate appears in the My store:

```

PS D:\> dir cert:\localmachine\my

PSParentPath: Microsoft.PowerShell.Security\Certificate::localmachine\my

Thumbprint          Subject
-----            -----
84857CBBE7A1C851A80AE22391EB2C39BF820CE7 CN=MyNetwork
5EFF2CE51EACA82408572A56AE1A9BCC7E0843C6 CN=EncryptedVirtualNetworks

```

1. Export the certificate to a file.

You need two copies of the certificate, one with the private key and one without.

```
$subjectName = "EncryptedVirtualNetworks" $cert = Get-ChildItem cert:\localmachine\my | ? {$_.Subject -eq "CN=$subjectName"} [System.io.file]::WriteAllBytes("c:$subjectName.pfx", $cert.Export("PFX", "secret"))  
Export-Certificate -Type CERT -FilePath "c:$subjectName.cer" -cert $cert
```

2. Install the certificates on each of your hyper-v hosts

```
PS C:> dir c:$subjectname.*
```

```
Directory: C:\  
  
Mode                LastWriteTime         Length Name  
----                -----          ---- -  
-a----    9/22/2017   4:54 PM           543 EncryptedVirtualNetworks.cer  
-a----    9/22/2017   4:54 PM        1706 EncryptedVirtualNetworks.pfx
```

1. Installing on a Hyper-V host

```
$server = "Server01"  
  
$subjectname = "EncryptedVirtualNetworks" copy c:$SubjectName.* \$server\c$ invoke-command -  
computername $server -ArgumentList $subjectname,"secret" { param ( [string] $SubjectName, [string]  
$Secret ) $certFullPath = "c:$SubjectName.cer"
```

```
# create a representation of the certificate file  
$certificate = new-object System.Security.Cryptography.X509Certificates.X509Certificate2  
$certificate.import($certFullPath)  
  
# import into the store  
$store = new-object System.Security.Cryptography.X509Certificates.X509Store("Root", "LocalMachine")  
$store.open("MaxAllowed")  
$store.add($certificate)  
$store.close()  
  
$certFullPath = "c:\$SubjectName.pfx"  
$certificate = new-object System.Security.Cryptography.X509Certificates.X509Certificate2  
$certificate.import($certFullPath, $Secret, "MachineKeySet,PersistKeySet")  
  
# import into the store  
$store = new-object System.Security.Cryptography.X509Certificates.X509Store("My", "LocalMachine")  
$store.open("MaxAllowed")  
$store.add($certificate)  
$store.close()  
  
# Important: Remove the certificate files when finished  
remove-item C:\$SubjectName.cer  
remove-item C:\$SubjectName.pfx
```

```
}
```

2. Repeat for each server in your environment.

After repeating for each server, you should have a certificate installed in the root and my store of each Hyper-V host.

3. Verify the installation of the certificate.

Verify the certificates by checking the contents of the My and Root certificate stores:

```
PS C:> enter-pssession Server1
```

```
[Server1]: PS C:\> get-childitem cert://localmachine/my,cert://localmachine/root | ? {$_['Subject -eq "CN=EncryptedVirtualNetworks"}
```

```
PSParentPath: Microsoft.PowerShell.Security\Certificate::localmachine\my
```

| Thumbprint | Subject |
|--|-----------------------------|
| ----- | ----- |
| 5EFF2CE51EACA82408572A56AE1A9BCC7E0843C6 | CN=EncryptedVirtualNetworks |

```
PSParentPath: Microsoft.PowerShell.Security\Certificate::localmachine\root
```

| Thumbprint | Subject |
|--|-----------------------------|
| ----- | ----- |
| 5EFF2CE51EACA82408572A56AE1A9BCC7E0843C6 | CN=EncryptedVirtualNetworks |

1. Make note of the Thumbprint.

You must make a note of the thumbprint because you need it to create the certificate credential object in the network controller.

Step 2. Create the Certificate Credential

After you install the certificate on each of the Hyper-V hosts connected to the network controller, you must now configure the network controller to use it. To do this, you must create a credential object containing the certificate thumbprint from the machine with the Network Controller PowerShell modules installed.

```
# Replace with thumbprint from your certificate
$thumbprint = "5EFF2CE51EACA82408572A56AE1A9BCC7E0843C6"

# Replace with your Network Controller URI
$uri = "https://nc.contoso.com"

Import-module networkcontroller

$credproperties = new-object Microsoft.Windows.NetworkController.CredentialProperties
$credproperties.Type = "X509Certificate"
$credproperties.Value = $thumbprint
New-networkcontrollercredential -connectionuri $uri -resourceid "EncryptedNetworkCertificate" -properties
$credproperties -force
```

TIP

You can reuse this credential for each encrypted virtual network, or you can deploy and use a unique certificate for each tenant.

Step 3. Configuring a Virtual Network for Encryption

This step assumes you have already created a virtual network name "My Network" and it contains at least one virtual subnet. For information on creating virtual networks, see [Create, Delete, or Update Tenant Virtual Networks](#).

NOTE

When communicating with another VM on the same subnet, whether its currently connected or connected at a later time, the traffic gets encrypted automatically.

1. Retrieve the Virtual Network and Credential objects from the network controller

```
$vnet = Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId "MyNetwork" $certcred =  
Get-NetworkControllerCredential -ConnectionUri $uri -ResourceId "EncryptedNetworkCertificate"
```

2. Add a reference to the certificate credential and enable encryption on individual subnets

```
$vnet.properties.EncryptionCredential = $certcred
```

Replace the Subnets index with the value corresponding to the subnet you want encrypted.

Repeat for each subnet where encryption is needed

```
$vnet.properties.Subnets[0].properties.EncryptionEnabled = $true
```

3. Put the updated Virtual Network object into the network controller

```
New-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId $vnet.ResourceId -Properties  
$vnet.Properties -force
```

Congratulations! You're done once you complete these steps.

Next steps

Manage tenant workloads

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

This topic contains links to documentation that allows you to manage tenant workloads by adding tenant virtual machines (VMs), using network virtual appliances, configuring software load balancing, and more.

This section includes the following topics.

- [Create a VM and Connect to a Tenant Virtual Network or VLAN](#)
- [Configure Quality of Service \(QoS\) for a Tenant VM Network Adapter](#)
- [Configure Datacenter Firewall Access Control Lists \(ACLs\)](#)
- [Configure the Software Load Balancer for Load Balancing and Network Address Translation \(NAT\)](#)
- [Use Network Virtual Appliances on a Virtual Network](#)
- [Guest Clustering in a Virtual Network](#)

Create a VM and connect to a tenant virtual network or VLAN

9/21/2018 • 4 minutes to read • [Edit Online](#)

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

In this topic, you create a tenant VM and connect it to either a virtual network that you created with Hyper-V Network Virtualization or to a virtual Local Area Network (VLAN). You can use Windows PowerShell Network Controller cmdlets to connect to a virtual network or NetworkControllerRESTWrappers to connect to a VLAN.

Use the processes described in this topic to deploy virtual appliances. With a few additional steps, you can configure appliances to process or inspect data packets that flow to or from other VMs on the Virtual Network.

The sections in this topic include example Windows PowerShell commands that contain example values for many parameters. Ensure that you replace example values in these commands with values that are appropriate for your deployment before you run these commands.

Prerequisites

1. VM network adapters created with static MAC addresses for the lifetime of the VM.

If the MAC address changes during the VM lifetime, Network Controller cannot configure the necessary policy for the network adapter. Not configuring the policy for the network prevents the network adapter from processing network traffic, and all communication with the network fails.

2. If the VM requires network access on startup, do not start the VM until after setting the interface ID on the VM network adapter port. If you start the VM before setting the interface ID, and the network interface does not exist, the VM cannot communicate on the network in the Network Controller, and all policies applied.
3. If you require custom ACLs for this network interface, then create the ACL now by using instructions in the topic [Use Access Control Lists \(ACLs\) to Manage Datacenter Network Traffic Flow](#)

Ensure that you have already created a Virtual Network before using this example command. For more information, see [Create, Delete, or Update Tenant Virtual Networks](#).

Create a VM and connect to a Virtual Network by using the Windows PowerShell Network Controller cmdlets

1. Create a VM with a VM network adapter that has a static MAC address.

```
New-VM -Generation 2 -Name "MyVM" -Path "C:\VMs\MyVM" -MemoryStartupBytes 4GB -VHDPath  
"c:\VMs\MyVM\Virtual Hard Disks\WindowsServer2016.vhdx" -SwitchName "SDNvSwitch"  
  
Set-VM -Name "MyVM" -ProcessorCount 4  
  
Set-VMNetworkAdapter -VMName "MyVM" -StaticMacAddress "00-11-22-33-44-55"
```

2. Get the virtual network that contains the subnet to which you want to connect the network adapter.

```
$vnet = get-networkcontrolervirtualnetwork -connectionuri $uri -ResourceId "Contoso_WebTier"
```

3. Create a network interface object in Network Controller.

TIP

In this step, you use the custom ACL.

```
$vmnicproperties = new-object Microsoft.Windows.NetworkController.NetworkInterfaceProperties
$vmnicproperties.PrivateMacAddress = "001122334455"
$vmnicproperties.PrivateMacAllocationMethod = "Static"
$vmnicproperties.IsPrimary = $true

$vmnicproperties.DnsSettings = new-object
Microsoft.Windows.NetworkController.NetworkInterfaceDnsSettings
$vmnicproperties.DnsSettings.DnsServers = @("24.30.1.11", "24.30.1.12")

$ipconfiguration = new-object Microsoft.Windows.NetworkController.NetworkInterfaceIpConfiguration
$ipconfiguration.resourceid = "MyVM_IP1"
$ipconfiguration.properties = new-object
Microsoft.Windows.NetworkController.NetworkInterfaceIpConfigurationProperties
$ipconfiguration.properties.PrivateIPAddress = "24.30.1.101"
$ipconfiguration.properties.PrivateIPAllocationMethod = "Static"

$ipconfiguration.properties.Subnet = new-object Microsoft.Windows.NetworkController.Subnet
$ipconfiguration.properties.subnet.ResourceRef = $vnet.Properties.Subnets[0].ResourceRef

$vmnicproperties.IpConfigurations = @($ipconfiguration)
New-NetworkControllerNetworkInterface -ResourceId "MyVM_Ethernet1" -Properties $vmnicproperties -
ConnectionUri $uri
```

4. Get the InstanceId for the network interface from Network Controller.

```
$nic = Get-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId "MyVM-Ethernet1"
```

5. Set the Interface ID on the Hyper-V VM network adapter port.

NOTE

You must run these commands on the Hyper-V host where the VM is installed.

```

#Do not change the hardcoded IDs in this section, because they are fixed values and must not change.

$FeatureId = "9940cd46-8b06-43bb-b9d5-93d50381fd56"

$vmNics = Get-VMNetworkAdapter -VMName "MyVM"

$CurrentFeature = Get-VMSwitchExtensionPortFeature -FeatureId $FeatureId -VMNetworkAdapter $vmNics

if ($CurrentFeature -eq $null)
{
    $Feature = Get-VMSystemSwitchExtensionPortFeature -FeatureId $FeatureId

    $Feature.SettingData.ProfileId = "{$($nic.InstanceId)}"
    $Feature.SettingData.NetCfgInstanceId = "{56785678-a0e5-4a26-bc9b-c0cba27311a3}"
    $Feature.SettingData.CdnLabelString = "TestCdn"
    $Feature.SettingData.CdnLabelId = 1111
    $Feature.SettingData.ProfileName = "Testprofile"
    $Feature.SettingData.VendorId = "{1FA41B39-B444-4E43-B35A-E1F7985FD548}"
    $Feature.SettingData.VendorName = "NetworkController"
    $Feature.SettingData.ProfileData = 1

    Add-VMSwitchExtensionPortFeature -VMSwitchExtensionFeature $Feature -VMNetworkAdapter $vmNics
}
else
{
    $CurrentFeature.SettingData.ProfileId = "{$($nic.InstanceId)}"
    $CurrentFeature.SettingData.ProfileData = 1

    Set-VMSwitchExtensionPortFeature -VMSwitchExtensionFeature $CurrentFeature -VMNetworkAdapter $vmNic
}

```

6. Start the VM.

```
Get-VM -Name "MyVM" | Start-VM
```

You have successfully created a VM, connected the VM to a tenant Virtual Network, and started the VM so that it can process tenant workloads.

Create a VM and connect to a VLAN by using NetworkControllerRESTWrappers

1. Create the VM and assign a static MAC address to the VM.

```

New-VM -Generation 2 -Name "MyVM" -Path "C:\VMs\MyVM" -MemoryStartupBytes 4GB -VHDPath
"c:\VMs\MyVM\Virtual Hard Disks\WindowsServer2016.vhdx" -SwitchName "SDNvSwitch"

Set-VM -Name "MyVM" -ProcessorCount 4

Set-VMNetworkAdapter -VMName "MyVM" -StaticMacAddress "00-11-22-33-44-55"

```

2. Set the VLAN ID on the VM network adapter.

```
Set-VMNetworkAdapterIsolation -VMName "MyVM" -AllowUntaggedTraffic $true -IsolationMode VLAN -
DefaultIsolationId 123
```

3. Get the logical network subnet and create the network interface.

```

$logicalnet = get-networkcontrollerLogicalNetwork -connectionuri $uri -ResourceId "00000000-2222-1111-9999-000000000002"

$vmnicproperties = new-object Microsoft.Windows.NetworkController.NetworkInterfaceProperties
$vmnicproperties.PrivateMacAddress = "00-1D-C8-B7-01-02"
$vmnicproperties.PrivateMacAllocationMethod = "Static"
$vmnicproperties.IsPrimary = $true

$vmnicproperties.DnsSettings = new-object
Microsoft.Windows.NetworkController.NetworkInterfaceDnsSettings
$vmnicproperties.DnsSettings.DnsServers = $logicalnet.Properties.Subnets[0].DNSServers

$ipconfiguration = new-object Microsoft.Windows.NetworkController.NetworkInterfaceIpConfiguration
$ipconfiguration.resourceid = "MyVM_Ip1"
$ipconfiguration.properties = new-object
Microsoft.Windows.NetworkController.NetworkInterfaceIpConfigurationProperties
$ipconfiguration.properties.PrivateIPAddress = "10.127.132.177"
$ipconfiguration.properties.PrivateIPAllocationMethod = "Static"

$ipconfiguration.properties.Subnet = new-object Microsoft.Windows.NetworkController.Subnet
$ipconfiguration.properties.subnet.ResourceRef = $logicalnet.Properties.Subnets[0].ResourceRef

$vmnicproperties.IpConfigurations = @($ipconfiguration)
$vnic = New-NetworkControllerNetworkInterface -ResourceID "MyVM_Ethernet1" -Properties $vmnicproperties
-ConnectionUri $uri

$vnic.InstanceId

```

4. Set the InstanceId on the Hyper-V port.

```

#The hardcoded IDs in this section are fixed values and must not change.
$FeatureId = "9940cd46-8b06-43bb-b9d5-93d50381fd56"

$vmNics = Get-VMNetworkAdapter -VMName "MyVM"

$CurrentFeature = Get-VMSwitchExtensionPortFeature -FeatureId $FeatureId -VMNetworkAdapter $vmNic

if ($CurrentFeature -eq $null)
{
    $Feature = Get-VMSystemSwitchExtensionFeature -FeatureId $FeatureId

    $Feature.SettingsData.ProfileId = "{$InstanceId}"
    $Feature.SettingsData.NetCfgInstanceId = "{56785678-a0e5-4a26-bc9b-c0cba27311a3}"
    $Feature.SettingsData.CdnLabelString = "TestCdn"
    $Feature.SettingsData.CdnLabelId = 1111
    $Feature.SettingsData.ProfileName = "Testprofile"
    $Feature.SettingsData.VendorId = "{1FA41B39-B444-4E43-B35A-E1F7985FD548}"
    $Feature.SettingsData.VendorName = "NetworkController"
    $Feature.SettingsData.ProfileData = 1

    Add-VMSwitchExtensionFeature -VMSwitchExtensionFeature $Feature -VMNetworkAdapter $vmNic
}
else
{
    $CurrentFeature.SettingsData.ProfileId = "{$InstanceId}"
    $CurrentFeature.SettingsData.ProfileData = 1

    Set-VMSwitchExtensionPortFeature -VMSwitchExtensionFeature $CurrentFeature -VMNetworkAdapter $vmNic
}

```

5. Start the VM.

```
Get-VM -Name "MyVM" | Start-VM
```

You have successfully created a VM, connected the VM to a VLAN, and started the VM so that it can process tenant workloads.

Configure Quality of Service (QoS) for a tenant VM network adapter

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

When you configure QoS for a tenant VM network adapter, you have a choice between Data Center Bridging (DCB) or Software Defined Networking (SDN) QoS.

1. **DCB.** You can configure DCB by using the Windows PowerShell NetQoS cmdlets. For an example, see the section “Enable Data Center Bridging” in the topic [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#).
2. **SDN QoS.** You can enable SDN QoS by using Network Controller, which can be set to limit bandwidth on a virtual interface to prevent a high-traffic VM from blocking other users. You can also configure SDN QoS to reserve a specific amount of bandwidth for a VM to ensure that the VM is accessible regardless of the amount of network traffic.

Apply all SDN QoS settings through the Port settings of the Network Interface properties. Refer to the table below for more details.

| ELEMENT NAME | DESCRIPTION |
|--------------|---|
| macSpoofing | <p>Allows VMs to change the source media access control (MAC) address in outgoing packets to a MAC address not assigned to the VM.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• Enabled – Use a different MAC address.• Disabled – Use only the MAC address assigned to it. |
| arpGuard | <p>Allows ARP guard only addresses specified in ArpFilter to pass through the port.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• Enabled - Allowed• Disabled – Not allowed |
| dhcpGuard | <p>Allows or drops any DHCP messages from a VM that claims to be a DHCP server.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• Enabled – Drops DHCP messages because the virtualized DHCP server is considered untrusted.• Disabled – Allows the message to be received because the virtualized DHCP server is considered trustworthy. |

| ELEMENT NAME | DESCRIPTION |
|------------------------|---|
| stormLimit | The number of packets (broadcast, multicast, and unknown unicast) per second a VM is allowed to send through the virtual network adapter. Packets beyond the limit during that one-second interval get dropped. A value of zero (0) means there is no limit.. |
| portFlowLimit | The maximum number of flows allowed to execute for the port. A value of blank or zero (0) means there is no limit. |
| vmqWeight | <p>The relative weight describes the affinity of the virtual network adapter to use virtual machine queue (VMQ). The range of value is 0 through 100.</p> <p>Allowed values:</p> <ul style="list-style-type: none"> • 0 – Disables the VMQ on the virtual network adapter. • 1-100 – Enables the VMQ on the virtual network adapter. |
| iovWeight | <p>The relative weight sets the affinity of the virtual network adapter to the assigned single-root I/O virtualization (SR-IOV) virtual function.</p> <p>Allowed values:</p> <ul style="list-style-type: none"> • 0 – Disables SR-IOV on the virtual network adapter. • 1-100 – Enables SR-IOV on the virtual network adapter. |
| iovInterruptModeration | <p>Allowed values:</p> <ul style="list-style-type: none"> • default – The physical network adapter vendor's setting determines the value. • adaptive - • off • low • medium • high <p>If you choose default, the physical network adapter vendor's setting determines the value. If you choose, adaptive, the runtime traffic pattern determines the interrupt moderation rate.</p> |
| iovQueuePairsRequested | <p>The number of hardware queue pairs allocated to an SR-IOV virtual function. If receive-side scaling (RSS) is required, and if the physical network adapter that binds to the virtual switch supports RSS on SR-IOV virtual functions, then more than one queue pair is required.</p> <p>Allowed values: 1 to 4294967295.</p> |

| ELEMENT NAME | DESCRIPTION |
|--------------|---|
| QosSettings | <p>Configure the following Qos settings, all of which are optional:</p> <ul style="list-style-type: none"> • outboundReservedValue - If outboundReservedMode is "absolute" then the value indicates the bandwidth, in Mbps, guaranteed to the virtual port for transmission (egress). If outboundReservedMode is "weight" then the value indicates the weighted portion of the bandwidth guaranteed. • outboundMaximumMbps - Indicates the maximum permitted send-side bandwidth, in Mbps, for the virtual port (egress). • InboundMaximumMbps - Indicates the maximum permitted receive-side bandwidth for the virtual port (ingress) in Mbps. |

Configure datacenter firewall Access Control Lists (ACLs)

9/21/2018 • 2 minutes to read • [Edit Online](#)

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

Once you've created an ACL and assigned it to a virtual subnet, you might want to override that default ACL on the virtual subnet with a specific ACL for an individual network interface. In this case, you apply specific ACLs directly to network interfaces attached to VLANs, instead of the virtual network. If you have ACLs set on the virtual subnet connected to the network interface, both ACLs are applied and prioritizes the network interface ACLs above the virtual subnet ACLs.

IMPORTANT

If you have not created an ACL and assigned it to a virtual network, see [Use Access Control Lists \(ACLs\) to Manage Datacenter Network Traffic Flow](#) to create an ACL and assign it to a virtual subnet.

In this topic, we show you how to add an ACL to a network interface. We also show you how to remove an ACL from a network interface using Windows PowerShell and the Network Controller REST API.

- [Example: Add an ACL to a network interface](#)
- [Example: Remove an ACL from a network interface by using Windows Powershell and the Network Controller REST API](#)

Example: Add an ACL to a network interface

In this example, we demonstrate how to add an ACL to a virtual network.

TIP

It is also possible to add an ACL at the same time that you create the network interface.

1. Get or create the network interface to which you will add the ACL.

```
$nic = get-networkcontrollernetworkinterface -ConnectionUri $uri -ResourceId "MyVM_Ethernet1"
```

2. Get or create the ACL you will add to the network interface.

```
$acl = get-networkcontrolleraccesscontrollist -ConnectionUri $uri -resourceid "AllowAllACL"
```

3. Assign the ACL to the AccessControlList property of the network interface

```
$nic.properties.ipconfigurations[0].properties.AccessControlList = $acl
```

4. Add the network interface in Network Controller

```
new-networkcontrolernetworkinterface -ConnectionUri $uri -Properties $nic.properties -ResourceId  
$nic.resourceid
```

Example: Remove an ACL from a network interface by using Windows Powershell and the Network Controller REST API

In this example, we show you how to remove an ACL. Removing an ACL applies the default set of rules to the network interface. The default set of rules allows all outbound traffic but blocks all inbound traffic.

NOTE

If you want to allow all inbound traffic, you must follow the previous [example](#) to add an ACL that allows all inbound and all outbound traffic.

1. Get the network interface from which you will remove the ACL.

```
$nic = get-networkcontrolernetworkinterface -ConnectionUri $uri -ResourceId "MyVM_Ethernet1"
```

2. Assign \$NULL to the AccessControlList property of the ipConfiguration.

```
$nic.properties.ipconfigurations[0].properties.AccessControlList = $null
```

3. Add the network interface object in Network Controller.

```
new-networkcontrolernetworkinterface -ConnectionUri $uri -Properties $nic.properties -ResourceId  
$nic.resourceid
```

Configure the Software Load Balancer for Load Balancing and Network Address Translation (NAT)

9/21/2018 • 6 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to use the Software Defined Networking (SDN) software load balancer (SLB) to provide outbound network address translation NAT, inbound NAT, or load balancing between multiple instances of an application.

This topic contains the following sections.

- [Software Load Balancer Overview](#)
- [Example: Create a public VIP for load balancing a pool of two VMs on a virtual network](#)
- [Example: Use SLB for outbound NAT](#)
- [Example: Add network interfaces to the back-end pool](#)
- [Example: Use the Software Load Balancer for forwarding traffic](#)

Software Load Balancer overview

The SDN Software Load Balancer (SLB) delivers high availability and network performance to your applications. It is a Layer 4 (TCP, UDP) load balancer that distributes incoming traffic among healthy service instances in cloud services or virtual machines defined in a load balancer set.

Configure SLB to do the following:

- Load balance incoming traffic external to a virtual network to virtual machines (VMs), also called public VIP load balancing.
- Load balance incoming traffic between VMs in a virtual network, between VMs in cloud services, or between on-premises computers and VMs in a cross-premises virtual network.
- Forward VM network traffic from the virtual network to external destinations using network address translation (NAT), also called outbound NAT.
- Forward external traffic to a specific VM, also called inbound NAT.

IMPORTANT

A known issue prevents the Load Balancer objects in the NetworkController Windows PowerShell module from working correctly in Windows Server 2016 5. Use dynamic hash tables and Invoke-WebRequest instead., shown in the following examples in this topic.

Example: Create a public VIP for load balancing a pool of two VMs on a virtual network

In this example, you create a load balancer object with a public VIP and two VMs as pool members to serve requests to the VIP. This example code also adds an HTTP health probe to detect whether one of the pool members becomes non-responsive.

1. Prepare the load balancer object.

```

$lbresourceId = "LB2"

$lbproperties = @{}
$lbproperties.frontendIpConfigurations = @()
$lbproperties.backendAddressPools = @()
$lbproperties.probes = @()
$lbproperties.loadBalancingRules = @()
$lbproperties.outboundNatRules = @()

```

2. Assign a front-end IP address, commonly referred to as a Virtual IP (VIP).

The VIP must be from an unused IP in one of the logical network IP pools given to the load balancer manager.

```

$vipip = "10.127.132.5"
$vipln = get-networkcontrollerlogicalnetwork -ConnectionUri $uri -resourceid "f8f67956-3906-4303-94c5-
09cf91e7e311"

$fe = @{}
$fe.resourceId = "FE1"
$fe.resourceRef = "/loadBalancers/$lbresourceId/frontendIPConfigurations/ $($fe.resourceId)"
$fe.properties = @{}
$fe.properties.subnet = @{}
$fe.properties.subnet.ResourceRef = $vipln.properties.Subnets[0].ResourceRef
$fe.properties.privateIPAddress = $vipip
$fe.properties.privateIPAllocationMethod = "Static"
$lbproperties.frontendIpConfigurations += $fe

```

3. Allocate a back-end address pool, which contains the Dynamic IPs (DIPs) that make up the members of the load-balanced set of VMs.

```

$backend = @{}
$backend.resourceId = "BE1"
$backend.resourceRef = "/loadBalancers/$lbresourceId/backendAddressPools/ $($backend.resourceId)"
$lbproperties.backendAddressPools += $backend

```

4. Define a health probe that the load balancer uses to determine the health state of the backend pool members.

In this example, you define an HTTP probe that queries to the RequestPath of "/health.htm." The query runs every 5 seconds, as specified by the IntervalInSeconds property.

The health probe must receive an HTTP response code of 200 for 11 consecutive queries for the probe to consider the back-end IP to be healthy. If the back-end IP is not healthy, it does not receive traffic from the load balancer.

IMPORTANT

Do not block traffic to or from the first IP in the subnet for any Access Control Lists (ACLs) that you apply to the back-end IP because that is the origination point for the probes.

Use the following example to define a health probe.

```

$lbprobe = @{}
$lbprobe.ResourceId = "Probe1"
$lbprobe.resourceRef = "/loadBalancers/$lbresourceId/Probes/$( $lbprobe.resourceId )"
$lbprobe.properties = @{}
$lbprobe.properties.protocol = "HTTP"
$lbprobe.properties.port = "80"
$lbprobe.properties.RequestPath = "/health.htm"
$lbprobe.properties.IntervalInSeconds = 5
$lbprobe.properties.NumberOfProbes = 11
$lbproperties.probes += $lbprobe

```

5. Define a load balancing rule to send traffic that arrives at the front-end IP to the back-end IP. In this example, the back-end pool receives TCP traffic to port 80.

Use the following example to define a load balancing rule:

```

$lbrule = @{}
$lbrule.ResourceId = "webserver1"
$lbrule.properties = @{}
$lbrule.properties.FrontEndIPConfigurations = @()
$lbrule.properties.FrontEndIPConfigurations += $fe
$lbrule.properties.backendaddresspool = $backend
$lbrule.properties.protocol = "TCP"
$lbrule.properties.frontendPort = 80
$lbrule.properties.Probe = $lbprobe
$lbproperties.loadbalancingRules += $lbrule

```

6. Add the load balancer configuration to Network Controller.

Use the following example to add the load balancer configuration to Network Controller:

```

$lb = @{}
$lb.ResourceId = $lbresourceid
$lb.properties = $lbproperties

$body = convertto-json $lb -Depth 100

Invoke-WebRequest -Headers @{"Accept"="application/json"} -ContentType "application/json; charset=UTF-8"
-Method "Put" -Uri "$uri/Networking/v1/loadbalancers/$lbresourceid" -Body $body -DisableKeepAlive -
UseBasicParsing

```

7. Follow the next example to add the network interfaces to this back-end pool.

Example: Use SLB for outbound NAT

In this example, you configure SLB with a back-end pool for providing outbound NAT capability for a VM on a virtual network's private address space to reach outbound to the internet.

1. Create the load balancer properties, front-end IP, and back-end pool.

```

$lbresourceId = "OutboundNATMembers"
$vipip = "10.127.132.7"

$vipln = get-networkcontrollerlogicalnetwork -ConnectionUri $uri -resourceid "f8f67956-3906-4303-94c5-09cf91e7e311"

$lbproperties = @{}
$lbproperties.frontendipconfigurations = @()
$lbproperties.backendAddressPools = @()
$lbproperties.probes = @()
$lbproperties.loadbalancingRules = @()
$lbproperties.OutboundNatRules = @()

$fe = @{}
$fe.resourceId = "FE1"
$fe.resourceRef = "/loadBalancers/$lbresourceId/frontendIPConfigurations/$($fe.resourceId)"
$fe.properties = @{}
$fe.properties.subnet = @{}
$fe.properties.subnet.ResourceRef = $vipln.properties.Subnets[0].ResourceRef
$fe.properties.privateIPAddress = $vipip
$fe.properties.privateIPAllocationMethod = "Static"
$lbproperties.frontendipconfigurations += $fe

$backend = @{}
$backend.resourceId = "BE1"
$backend.resourceRef = "/loadBalancers/$lbresourceId/backendAddressPools/$($backend.resourceId)"
$lbproperties.backendAddressPools += $backend

```

2. Define the outbound NAT rule.

```

$onat = @{}
$onat.ResourceId = "onat1"
$onat.properties = @{}
$onat.properties.frontendipconfigurations = @()
$onat.properties.frontendipconfigurations += $fe
$onat.properties.backendaddresspool = $backend
$onat.properties.protocol = "ALL"
$lbproperties.OutboundNatRules += $onat

```

3. Add the load balancer object in Network Controller.

```

$lb = @{}
$lb.ResourceId = $lbresourceid
$lb.properties = $lbproperties

$body = convertto-json $lb -Depth 100

Invoke-WebRequest -Headers @{"Accept"="application/json"} -ContentType "application/json; charset=UTF-8"
-Method "Put" -Uri "$uri/Networking/v1/loadbalancers/$lbresourceid" -Body $body -DisableKeepAlive -
UseBasicParsing

```

4. Follow the next example to add the network interfaces to which you want to provide internet access.

Example: Add network interfaces to the back-end pool

In this example, you add network interfaces to the back-end pool. You must repeat this step for each network interface that can process requests made to the VIP.

You can also repeat this process on a single network interface to add it to multiple load balancer objects. For example, if you have a load balancer object for a web server VIP and a separate load balancer object to provide outbound NAT.

1. Get the load balancer object containing the back-end pool to add a network interface.

```
$lbresourceid = "LB2"
$lb = (Invoke-WebRequest -Headers @{"Accept"="application/json"} -ContentType "application/json;
charset=UTF-8" -Method "Get" -Uri "$uri/Networking/v1/loadbalancers/$lbresourceid" -DisableKeepAlive -
UseBasicParsing).content | convertfrom-json
```

2. Get the network interface and add the backendaddress pool to the loadbalancerbackendaddresspools array.

```
$nic = get-networkcontrolernetworkinterface -connectionuri $uri -resourceid 6daca142-7d94-0000-1111-
c38c0141be06
$nic.properties.IpConfigurations[0].properties.LoadBalancerBackendAddressPools +=
$lb.properties.backendaddresspools[0]
```

3. Put the network interface to apply the change.

```
new-networkcontrolernetworkinterface -connectionuri $uri -resourceid 6daca142-7d94-0000-1111-
c38c0141be06 -properties $nic.properties -force
```

Example: Use the Software Load Balancer for forwarding traffic

If you need to map a Virtual IP to a single network interface on a virtual network without defining individual ports, you can create an L3 forwarding rule. This rule forwards all traffic to and from the VM via the assigned VIP contained in a PublicIPAddress object.

If you defined the VIP and DIP as the same subnet, then this is equivalent to performing L3 forwarding without NAT.

NOTE

This process does not require you to create a load balancer object. Assigning the PublicIPAddress to the network interface is enough information for the Software Load Balancer to perform its configuration.

1. Create a public IP object to contain the VIP.

```
$publicIPProperties = new-object Microsoft.Windows.NetworkController.PublicIpAddressProperties
$publicIPProperties.ipaddress = "10.127.132.6"
$publicIPProperties.PublicIPAllocationMethod = "static"
$publicIPProperties.IdleTimeoutInMinutes = 4
$publicIP = New-NetworkControllerPublicIpAddress -ResourceId "MyPIP" -Properties $publicIPProperties -
ConnectionUri $uri
```

2. Assign the PublicIPAddress to a network interface.

```
$nic = get-networkcontrolernetworkinterface -connectionuri $uri -resourceid 6daca142-7d94-0000-1111-
c38c0141be06
$nic.properties.IpConfigurations[0].Properties.PublicIPAddress = $publicIP
New-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId $nic.ResourceId -Properties
$nic.properties
```

Use network virtual appliances on a virtual network

9/21/2018 • 4 minutes to read • [Edit Online](#)

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

In this topic, you learn how to deploy network virtual appliances on tenant virtual networks. You can add network virtual appliances to networks that perform user-defined routing and port mirroring functions.

Types of network virtual appliances

You can use one of the two types of virtual appliances:

1. **User defined routing** - replaces distributed routers on the virtual network with the routing capabilities of the virtual appliance. With user-defined routing, the virtual appliance gets used as a router between the virtual subnets on the virtual network.
2. **Port mirroring** - all network traffic that is entering or leaving the monitored port is duplicated and sent to a virtual appliance for analysis.

Deploying a network virtual appliance

To deploy a network virtual appliance, you must first create a VM that contains the appliance, and then connect the VM to the appropriate virtual network subnets. For more details, see [Create a Tenant VM and Connect to a Tenant Virtual Network or VLAN](#).

Some appliances require multiple virtual network adapters. Usually, one network adapter dedicated to the appliance management while additional adapters process traffic. If your appliance requires multiple network adapters, you must create each network interface in Network Controller. You must also assign an interface ID on each host for each of the additional adapters that are on different virtual subnets.

Once you've deployed the network virtual appliance, you can use the appliance for defined routing, porting mirroring, or both.

Example: User-defined routing

For most environments, you only need the system routes already defined by the virtual network's distributed router. However, you might need to create a routing table and add one or more routes in specific cases, such as:

- Force tunneling to the Internet via your on-premises network.
- Use of virtual appliances in your environment.

For these scenarios, you must create a routing table and add user-defined routes to the table. You can have multiple routing tables, and you can associate the same routing table to one or more subnets. You can only associate each subnet to a single routing table. All VMs in a subnet use the routing table associated to the subnet.

Subnets rely on system routes until a routing table gets associated to the subnet. After an association exists, routing is done based on Longest Prefix Match (LPM) among both user-defined routes and system routes. If there is more than one route with the same LPM match, then the user defined route is selected first - before the system route.

Procedure:

1. Create the route table properties, which contains all of the user defined routes.

System routes still apply according to the rules defined above.

```
$routetableproperties = new-object Microsoft.Windows.NetworkController.RouteTableProperties
```

2. Add a route to the routing table properties.

Any route destined for 12.0.0.0/8 subnet gets routed to the virtual appliance at 192.168.1.10. The appliance must have a virtual network adapter attached to the virtual network with that IP assigned to a network interface.

```
$route = new-object Microsoft.Windows.NetworkController.Route
$route.ResourceID = "0_0_0_0_0"
$route.properties = new-object Microsoft.Windows.NetworkController.RouteProperties
$route.properties.AddressPrefix = "0.0.0.0/0"
$route.properties.nextHopType = "VirtualAppliance"
$route.properties.nextHopIpAddress = "192.168.1.10"
$routetableproperties.routes += $route
```

TIP

If you want to add more routes, repeat this step for each route you want to define.

3. Add the routing table to Network Controller.

```
$routetable = New-NetworkControllerRouteTable -ConnectionUri $uri -ResourceId "Route1" -Properties
$routetableproperties
```

4. Apply the routing table to the virtual subnet.

When you apply the route table to the virtual subnet, the first virtual subnet in the Tenant1_Vnet1 network uses the route table. You can assign the route table to as many of the subnets in the virtual network as you want.

```
$vnet = Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId "Tenant1_VNet1"
$vnet.properties.subnets[0].properties.RouteTable = $routetable
New-NetworkControllerVirtualNetwork -ConnectionUri $uri -Properties $vnet.properties -ResourceId
$vnet.ResourceId
```

As soon as you apply the routing table to the virtual network, traffic gets forwarded to the virtual appliance. You must configure the routing table in the virtual appliance to forward the traffic, in a manner that is appropriate for your environment.

Example: Port mirroring

In this example, you configure the traffic for MyVM_Ethernet1 to mirror Appliance_Ethernet1. We assume that you've deployed two VMs, one as the appliance and the other as the VM to monitor with mirroring.

The appliance must have a second network interface for management. After you enable mirroring as a destination on Appliance_Ethernet1, it no longer receives traffic destined for the IP interface configured there.

Procedure:

1. Get the virtual network on which your VMs are located.

```
$vnet = Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId "Tenant1_VNet1"
```

2. Get the Network Controller network interfaces for the mirroring source and destination.

```
$dstNic = get-networkcontrollernetworkinterface -ConnectionUri $uri -ResourceId "Appliance_Ethernet1"  
$srcNic = get-networkcontrollernetworkinterface -ConnectionUri $uri -ResourceId "MyVM_Ethernet1"
```

3. Create a serviceinsertionproperties object to contain the port mirroring rules and the element which represents the destination interface.

```
$portmirror = [Microsoft.Windows.NetworkController.ServiceInsertionProperties]::new()  
$portMirror.Priority = 1
```

4. Create a serviceinsertionrules object to contain the rules that must be matched in order for the traffic to be sent to the appliance.

The rules defined below match all traffic, both inbound and outbound, which represents a traditional mirror. You can adjust these rules if you are interested in mirroring a specific port, or specific source/destinations.

```
$portmirror.ServiceInsertionRules = [Microsoft.Windows.NetworkController.ServiceInsertionRule[]]::new(1)  
  
$portmirror.ServiceInsertionRules[0] = [Microsoft.Windows.NetworkController.ServiceInsertionRule]::new()  
$portmirror.ServiceInsertionRules[0].ResourceId = "Rule1"  
$portmirror.ServiceInsertionRules[0].Properties =  
[Microsoft.Windows.NetworkController.ServiceInsertionRuleProperties]::new()  
  
$portmirror.ServiceInsertionRules[0].Properties.Description = "Port Mirror Rule"  
$portmirror.ServiceInsertionRules[0].Properties.Protocol = "All"  
$portmirror.ServiceInsertionRules[0].Properties.SourcePortRangeStart = "0"  
$portmirror.ServiceInsertionRules[0].Properties.SourcePortRangeEnd = "65535"  
$portmirror.ServiceInsertionRules[0].Properties.DestinationPortRangeStart = "0"  
$portmirror.ServiceInsertionRules[0].Properties.DestinationPortRangeEnd = "65535"  
$portmirror.ServiceInsertionRules[0].Properties.SourceSubnets = "*"  
$portmirror.ServiceInsertionRules[0].Properties.DestinationSubnets = "*"
```

5. Create a serviceinsertionelements object to contain the network interface of the mirrored appliance.

```
$portmirror.ServiceInsertionElements =  
[Microsoft.Windows.NetworkController.ServiceInsertionElement[]]::new(1)  
  
$portmirror.ServiceInsertionElements[0] =  
[Microsoft.Windows.NetworkController.ServiceInsertionElement]::new()  
$portmirror.ServiceInsertionElements[0].ResourceId = "Element1"  
$portmirror.ServiceInsertionElements[0].Properties =  
[Microsoft.Windows.NetworkController.ServiceInsertionElementProperties]::new()  
  
$portmirror.ServiceInsertionElements[0].Properties.Description = "Port Mirror Element"  
$portmirror.ServiceInsertionElements[0].Properties.NetworkInterface = $dstNic  
$portmirror.ServiceInsertionElements[0].Properties.Order = 1
```

6. Add the service insertion object in Network Controller.

When you issue this command, all traffic to the appliance network interface specified in the previous step stops.

```
$portMirror = New-NetworkControllerServiceInsertion -ConnectionUri $uri -Properties $portmirror -  
ResourceId "MirrorAll"
```

7. Update the network interface of the source to be mirrored.

```
$srcNic.Properties.IpConfigurations[0].Properties.ServiceInsertion = $portMirror  
$srcNic = New-NetworkControllerNetworkInterface -ConnectionUri $uri -Properties $srcNic.Properties -  
ResourceId $srcNic.ResourceId
```

After completing these steps, the Appliance_Ethernet1 interface mirrors the traffic from the MyVM_Ethernet1 interface.

Guest clustering in a virtual network

9/21/2018 • 3 minutes to read • [Edit Online](#)

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

Virtual machines connected to a virtual network are only permitted to use the IP addresses that Network Controller has assigned to communicate on the network. Clustering technologies that require a floating IP address, such as Microsoft Failover Clustering, require some extra steps to function correctly.

The method for making the floating IP reachable is to use a Software Load Balancer (SLB) virtual IP (VIP). The software load balancer must be configured with a health probe on a port on that IP so that SLB directs traffic to the machine that currently has that IP.

Example: Load balancer configuration

This example assumes that you've already created the VMs which will become cluster nodes, and attached them to a Virtual Network. For guidance, refer to [Create a VM and Connect to a Tenant Virtual Network or VLAN](#).

In this example you will create a virtual IP address (192.168.2.100) to represent the floating IP address of the cluster, and configure a health probe to monitor TCP port 59999 to determine which node is the active one.

1. Select the VIP.

Prepare by assigning a VIP IP address, which can be any unused or reserved address in the same subnet as the cluster nodes. The VIP must match the floating address of the cluster.

```
$VIP = "192.168.2.100"
$subnet = "Subnet2"
$VirtualNetwork = "MyNetwork"
$ResourceId = "MyNetwork_InternalVIP"
```

2. Create the load balancer properties object.

```
$LoadBalancerProperties = new-object Microsoft.Windows.NetworkController.LoadBalancerProperties
```

3. Create a front-end IP address.

```
$LoadBalancerProperties.frontendipconfigurations += $FrontEnd = new-object
Microsoft.Windows.NetworkController.LoadBalancerFrontendIpConfiguration
$FrontEnd.properties = new-object
Microsoft.Windows.NetworkController.LoadBalancerFrontendIpConfigurationProperties
$FrontEnd.resourceId = "Frontend1"
$FrontEnd.resourceRef = "/loadBalancers/$ResourceId/frontendIPConfigurations/($FrontEnd.resourceId)"
$FrontEnd.properties.subnet = new-object Microsoft.Windows.NetworkController.Subnet
$FrontEnd.properties.subnet.ResourceRef = "/VirtualNetworks/MyNetwork/Subnets/Subnet2"
$FrontEnd.properties.privateIPAddress = $VIP
$FrontEnd.properties.privateIPAllocationMethod = "Static"
```

4. Create a back-end pool to contain the cluster nodes.

```

$BackEnd = new-object Microsoft.Windows.NetworkController.LoadBalancerBackendAddressPool
$BackEnd.properties = new-object Microsoft.Windows.NetworkController.LoadBalancerBackendAddressPoolProperties
$BackEnd.resourceId = "Backend1"
$BackEnd.resourceRef = "/loadBalancers/$ResourceId/backendAddressPools/$( $BackEnd.resourceId )"
$LoadBalancerProperties.backendAddressPools += $BackEnd

```

5. Add a probe to detect which cluster node the floating address is currently active on.

NOTE

The probe query against the VM's permanent address at the port defined below. The port must only respond on the active node.

```

$LoadBalancerProperties.probes += $lbprobe = new-object
Microsoft.Windows.NetworkController.LoadBalancerProbe
$lbprobe.properties = new-object Microsoft.Windows.NetworkController.LoadBalancerProbeProperties

$lbprobe.ResourceId = "Probe1"
$lbprobe.resourceRef = "/loadBalancers/$ResourceId/Probes/$( $lbprobe.resourceId )"
$lbprobe.properties.protocol = "TCP"
$lbprobe.properties.port = "59999"
$lbprobe.properties.IntervalInSeconds = 5
$lbprobe.properties.NumberOfProbes = 11

```

6. Add the load balancing rules for TCP port 1433.

You can modify the protocol and port as needed. You can also repeat this step multiple times for additional ports and protocols on this VIP. It is important that `EnableFloatingIP` is set to `$true` because this tells the load balancer to send the packet to the node with the original VIP in place.

```

$LoadBalancerProperties.loadbalancingRules += $lbrule = new-object
Microsoft.Windows.NetworkController.LoadBalancingRule
$lbrule.properties = new-object Microsoft.Windows.NetworkController.LoadBalancingRuleProperties
$lbrule.ResourceId = "Rules1"

$lbrule.properties.frontendipconfigurations += $FrontEnd
$lbrule.properties.backendaddresspool = $BackEnd
$lbrule.properties.protocol = "TCP"
$lbrule.properties.frontendPort = $lbrule.properties.backendPort = 1433
$lbrule.properties.IdleTimeoutInMinutes = 4
$lbrule.properties.EnableFloatingIP = $true
$lbrule.properties.Probe = $lbprobe

```

7. Create the load balancer in Network Controller.

```

$lb = New-NetworkControllerLoadBalancer -ConnectionUri $URI -ResourceId $ResourceId -Properties
$LoadBalancerProperties -Force

```

8. Add the cluster nodes to the backend pool.

You can add as many nodes to the pool as you require for the cluster.

```

# Cluster Node 1

$nic = get-networkcontrollernetworkinterface -connectionuri $uri -resourceid "ClusterNode1_Network-
Adapter"
$nic.properties.IpConfigurations[0].properties.LoadBalancerBackendAddressPools +=
$lb.properties.backendaddresspools[0]
$nic = new-networkcontrollernetworkinterface -connectionuri $uri -resourceid $nic.resourceid -
properties $nic.properties -force

# Cluster Node 2

$nic = get-networkcontrollernetworkinterface -connectionuri $uri -resourceid "ClusterNode2_Network-
Adapter"
$nic.properties.IpConfigurations[0].properties.LoadBalancerBackendAddressPools +=
$lb.properties.backendaddresspools[0]
$nic = new-networkcontrollernetworkinterface -connectionuri $uri -resourceid $nic.resourceid -
properties $nic.properties -force

```

Once you've created the load balancer and added the network interfaces to the backend pool, you are ready to configure the cluster.

9. (Optional) If you are using a Microsoft Failover Cluster, continue with the next example.

Example 2: Configuring a Microsoft failover cluster

You can use the following steps to configure a failover cluster.

1. Install and configure properties for a failover cluster.

```

add-windowsfeature failover-clustering -IncludeManagementTools
Import-module failoverclusters

$ClusterName = "MyCluster"

$ClusterNetworkName = "Cluster Network 1"
$IPResourceName =
$ILBIP = "192.168.2.100"

$nodes = @("DB1", "DB2")

```

2. Create the cluster on one node.

```
New-Cluster -Name $ClusterName -NoStorage -Node $nodes[0]
```

3. Stop the cluster resource.

```
Stop-ClusterResource "Cluster Name"
```

4. Set the cluster IP and probe port.

The IP address must match the front-end ip address used in the previous example, and the probe port must match the probe port in the previous example.

```

Get-ClusterResource "Cluster IP Address" | Set-ClusterParameter -Multiple
@{"Address"="$ILBIP"; "ProbePort"="59999"; "SubnetMask"="255.255.255.255"; "Network"="$ClusterNetworkName";
"EnableDhcp"=0}

```

5. Start the cluster resources.

```
Start-ClusterResource "Cluster IP Address" -Wait 60  
Start-ClusterResource "Cluster Name" -Wait 60
```

6. Add the remaining nodes.

```
Add-ClusterNode $nodes[1]
```

Your cluster is active. Traffic going to the VIP on the specified port is directed at the active node.

Upgrade, backup, and restore SDN infrastructure

9/21/2018 • 7 minutes to read • [Edit Online](#)

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

In this topic, you learn how to update, backup and restore an SDN infrastructure.

Upgrade the SDN infrastructure

SDN infrastructure can be upgraded from Windows Server 2016 to Windows Server 2019. For upgrade ordering, follow the same sequence of steps as mentioned in the section "Update the SDN infrastructure". Before upgrade, it is recommended to take a backup of the Network Controller database.

For Network Controller machines, use the `Get-NetworkControllerNode` cmdlet to check the status of the node after the upgrade has been completed. Ensure that the node comes back to "Up" status before upgrading the other nodes. Once you have upgraded all of the Network Controller nodes, the Network Controller updates the microservices running within the Network Controller cluster within an hour. You can trigger an immediate update using the `update-networkcontroller` cmdlet.

Install the same Windows updates on all of the operating system components of the Software Defined Networking (SDN) system, which includes:

- SDN enabled Hyper-V hosts
- Network Controller VMs
- Software Load Balancer Mux VMs
- RAS Gateway VMs

IMPORTANT

If you use System Center Virtual Manager, you must update it with the latest update rollups.

When you update each component, you can use any of the standard methods for installing Windows updates. However, to ensure minimal downtime for workloads and the integrity of the Network Controller database, follow these steps:

1. Update the management consoles.

Install the updates on each of the computers where you use the Network Controller Powershell module. Including anywhere that you have the RSAT-NetworkController role installed by itself. Excluding the Network Controller VMs themselves; you update them in the next step.

2. On the first Network Controller VM, install all updates and restart.

3. Before proceeding to the next Network Controller VM, use the `get-networkcontrollernode` cmdlet to check the status of the node that you updated and restarted.

4. During the reboot cycle, wait for the Network Controller node to go down and then come back up again.

After rebooting the VM, it can take several minutes before it goes back into the **Up** status. For an example of the output, see

5. Install updates on each SLB Mux VM one at a time to ensure continuous availability of the load balancer

infrastructure.

6. Update Hyper-V hosts and RAS gateways, starting with the hosts that contain the RAS gateways that are in **Standby** mode.

RAS gateway VMs can't be migrated live without losing tenant connections. During the update cycle, you must be careful to minimize the number of times tenant connections failover to a new RAS gateway. By coordinating the update of hosts and RAS gateways, each tenant fails over once, at most.

- a. Evacuate the host of VMs that are capable of live migration.

RAS gateway VMs should remain on the host.

- b. Install updates on each Gateway VM on this host.

- c. If the update requires the gateway VM to reboot then reboot the VM.

- d. Install updates on the host containing the gateway VM that was just Updated.

- e. Reboot the host if required by the updates.

- f. Repeat for each additional host containing a standby gateway.

If no standby gateways remain, then follow these same steps for all remaining hosts.

Example: Use the `get-networkcontrollernode` cmdlet

In this example, you see the output for the `get-networkcontrollernode` cmdlet run from within one of the Network Controller VMs.

The status of the nodes that you see in the example output is:

- NCNode1.contoso.com = Down
- NCNode2.contoso.com = Up
- NCNode3.contoso.com = Up

IMPORTANT

You must wait several minutes until the status for the node changes to **Up** before you update any additional nodes, one at a time.

Once you have updated all of the Network Controller nodes, the Network Controller updates the microservices running within the Network Controller cluster within an hour.

TIP

You can trigger an immediate update using the `update-networkcontroller` cmdlet.

```
PS C:\> get-networkcontrolernode
Name      : NCNode1.contoso.com
Server    : NCNode1.Contoso.com
FaultDomain : fd:/NCNode1.Contoso.com
RestInterface : Ethernet
NodeCertificate :
Status     : Down

Name      : NCNode2.Contoso.com
Server    : NCNode2.contoso.com
FaultDomain : fd:/ NCNode2.Contoso.com
RestInterface : Ethernet
NodeCertificate :
Status     : Up

Name      : NCNode3.Contoso.com
Server    : NCNode3.Contoso.com
FaultDomain : fd:/ NCNode3.Contoso.com
RestInterface : Ethernet
NodeCertificate :
Status     : Up
```

Example: Use the update-networkcontroller cmdlet

In this example, you see the output for the `update-networkcontroller` cmdlet to force Network Controller to update.

IMPORTANT

Run this cmdlet when you have no more updates to install.

```
PS C:\> update-networkcontroller
NetworkControllerClusterVersion NetworkControllerVersion
-----
10.1.1           10.1.15
```

Backup the SDN infrastructure

Regular backups of the Network Controller database ensures business continuity in the event of a disaster or data loss. Backing up the Network Controller VMs is not sufficient because it does not ensure that the session continues across the multiple Network Controller nodes.

Requirements:

- An SMB share and credentials with Read/Write permissions to the share and file system.
- You can optionally use a Group Managed Service Account (GMSA) if the Network Controller was installed using a GMSA as well.

Procedure:

1. Use the VM backup method of your choice, or use Hyper-V to export a copy of each Network Controller VM.

Backing up the Network Controller VM ensures that the necessary certificates for decrypting the database are present.

2. If using System Center Virtual Machine Manager (SCVMM), stop the SCVMM service and back it up via SQL Server.

The goal here is to ensure that no updates get made to SCVMM during this time, which could create an

inconsistency between the Network Controller backup and SCVMM.

IMPORTANT

Do not re-start the SCVMM service until the Network Controller backup is complete.

3. Backup the Network Controller database with the `new-networkcontrollerbackup` cmdlet.
4. Check the completion and success of the backup with the `get-networkcontrollerbackup` cmdlet.
5. If using SCVMM, start SCVMM service.

Example: Backing up the Network Controller database

```
$URI = "https://NC.contoso.com"
$Credential = Get-Credential

# Get or Create Credential object for File share user

$ShareUserResourceId = "BackupUser"

$ShareCredential = Get-NetworkControllerCredential -ConnectionURI $URI -Credential $Credential | Where
{$_._ResourceId -eq $ShareUserResourceId }
If ($ShareCredential -eq $null) {
    $CredentialProperties = New-Object Microsoft.Windows.NetworkController.CredentialProperties
    $CredentialProperties.Type = "usernamePassword"
    $CredentialProperties.UserName = "contoso\alyoung"
    $CredentialProperties.Value = "<Password>"

    $ShareCredential = New-NetworkControllerCredential -ConnectionURI $URI -Credential $Credential -Properties
    $CredentialProperties -ResourceId $ShareUserResourceId -Force
}

# Create backup

$BackupTime = (get-date).ToString("s").Replace(":", " ")

$BackupProperties = New-Object Microsoft.Windows.NetworkController.NetworkControllerBackupProperties
$BackupProperties.BackupPath = "\\fileshare\backups\NetworkController\$BackupTime"
$BackupProperties.Credential = $ShareCredential

$Backup = New-NetworkControllerBackup -ConnectionURI $URI -Credential $Credential -Properties $BackupProperties
-ResourceId $BackupTime -Force
```

Example: Checking the status of a Network Controller backup operation

```
PS C:\ > Get-NetworkControllerBackup -ConnectionUri $URI -Credential $Credential -ResourceId $Backup.ResourceId
| ConvertTo-JSON -Depth 10
{
    "Tags": null,
    "ResourceRef": "/networkControllerBackup/2017-04-25T16_53_13",
    "InstanceId": "c3ea75ae-2892-4e10-b26c-a2243b755dc8",
    "Etag": "W/\"0dafea6c-39db-401b-bda5-d2885ded470e\"",
    "ResourceMetadata": null,
    "ResourceId": "2017-04-25T16_53_13",
    "Properties": {
        "BackupPath": "\\\\[fileshare]\backups\NetworkController\\2017-04-25T16_53_13",
        "ErrorMessage": "",
        "FailedResourcesList": [
            ],
        "SuccessfulResourcesList": [
            "/networking/v1/credentials/11ebfc10-438c-4a96-a1ee-
8a048ce675be",
```

"/networking/v1/credentials/41229069-85d4-4352-be85-
034d0c5f4658",
"/networking/v1/credentials/b2a82c93-2583-4a1f-91f8-
232b801e11bb",
"/networking/v1/credentials/BackupUser",
"/networking/v1/credentials/fd5b1b96-b302-4395-b6cd-
ed9703435dd1",
"/networking/v1/virtualNetworkManager/configuration",
"/networking/v1/virtualSwitchManager/configuration",
"/networking/v1/accessControlLists/f8b97a4c-4419-481d-b757-
a58483512640",
"/networking/v1/logicalnetworks/24fa1af9-88d6-4cdc-aba0-
66e38c1a7bb8",
"/networking/v1/logicalnetworks/48610528-f40b-4718-938e-
99c2be76f1e0",
"/networking/v1/logicalnetworks/89035b49-1ee3-438a-8d7a-
f93cbae40619",
"/networking/v1/logicalnetworks/a9c8eaa0-519c-4988-acd6-
11723e9efae5",
"/networking/v1/logicalnetworks/d4ea002c-c926-4c57-a178-
461d5768c31f",
"/networking/v1/macPools/11111111-1111-1111-1111-
111111111111",
"/networking/v1/loadBalancerManager/config",
"/networking/v1/publicIPAddresses/2c502b2d-b39a-4be1-a85a-
55ef6a3a9a1d",
"/networking/v1/GatewayPools/Default",
"/networking/v1/servers/4c4c4544-0058-5810-8056-
b4c04f395931",
"/networking/v1/servers/4c4c4544-0058-5810-8057-
b4c04f395931",
"/networking/v1/servers/4c4c4544-0058-5910-8056-",
"/networking/v1/networkInterfaces/058430d3-af43-4328-a440-
56540f41da50",
"/networking/v1/networkInterfaces/08756090-6d55-4dec-98d5-
80c4c5a47db8",
"/networking/v1/networkInterfaces/2175d74a-aacd-44e2-80d3-
03f39ea3bc5d",
"/networking/v1/networkInterfaces/2400c2c3-2291-4b0b-929c-
9bb8da55851a",
"/networking/v1/networkInterfaces/4c695570-6faa-4e4d-a552-
0b36ed3e0962",
"/networking/v1/networkInterfaces/7e317638-2914-42a8-a2dd-
3a6d966028d6",
"/networking/v1/networkInterfaces/834e3937-f43b-4d3c-88be-
d79b04e63bce",
"/networking/v1/networkInterfaces/9d668fe6-b1c6-48fc-b8b1-
b3f98f47d508",
"/networking/v1/networkInterfaces/ac4650ac-c3ef-4366-96e7-
d9488fb661ba",
"/networking/v1/networkInterfaces/b9f23e35-d79e-495f-a1c9-
fa626b85ae13",
"/networking/v1/networkInterfaces/fdd929f1-f64f-4463-949a-
77b67fe6d048",
"/networking/v1/virtualServers/15a891ee-7509-4e1d-878d-
de0cb4fa35fd",
"/networking/v1/virtualServers/57416993-b410-44fd-9675-
727cd4e98930",
"/networking/v1/virtualServers/5f8aebdc-ee5b-488f-ac44-
dd6b57bd316a",
"/networking/v1/virtualServers/6c812217-5931-43dc-92a8-
1da3238da893",
"/networking/v1/virtualServers/d78b7fa3-812d-4011-9997-
aeb5ded2b431",
"/networking/v1/virtualServers/d90820a5-635b-4016-9d6f-
bf3f1e18971d",
"/networking/v1/loadBalancerMuxes/5f8aebdc-ee5b-488f-ac44-
dd6b57bd316a_suffix",

```

        "/networking/v1/loadBalancerMuxes/d78b7fa3-812d-4011-9997-
aeb5ded2b431_suffix",
        "/networking/v1/loadBalancerMuxes/d90820a5-635b-4016-9d6f-
bf3f1e18971d_suffix",
        "/networking/v1/Gateways/15a891ee-7509-4e1d-878d-
de0cb4fa35fd_suffix",
        "/networking/v1/Gateways/57416993-b410-44fd-9675-
727cd4e98930_suffix",
        "/networking/v1/Gateways/6c812217-5931-43dc-92a8-
1da3238da893_suffix",
        "/networking/v1/virtualNetworks/b3dbafb9-2655-433d-b47d-
a0e0bbac867a",
        "/networking/v1/virtualNetworks/d705968e-2dc2-48f2-a263-
76c7892fb143",
        "/networking/v1/loadBalancers/24fa1af9-88d6-4cdc-aba0-
66e38c1a7bb8_10.127.132.2",
        "/networking/v1/loadBalancers/24fa1af9-88d6-4cdc-aba0-
66e38c1a7bb8_10.127.132.3",
        "/networking/v1/loadBalancers/24fa1af9-88d6-4cdc-aba0-
66e38c1a7bb8_10.127.132.4"
    ],
    "InProgressResourcesList": [
        ],
    "ProvisioningState": "Succeeded",
    "Credential": {
        "Tags": null,
        "ResourceRef": "/credentials/BackupUser",
        "InstanceId": "00000000-0000-0000-0000-000000000000",
        "Etag": null,
        "ResourceMetadata": null,
        "ResourceId": null,
        "Properties": null
    }
}
}

```

Restore the SDN infrastructure from a backup

When you restore all the necessary components from backup, the SDN environment returns to an operational state.

IMPORTANT

The steps vary depending on the number of components restored.

1. If necessary, redeploy Hyper-V hosts and the necessary storage.
2. If necessary, restore the Network Controller VMs, RAS gateway VMs and Mux VMs from backup.
3. Stop NC host agent and SLB host agent on all Hyper-V hosts:

```

stop-service slbhostagent

stop-service nchostagent

```

4. Stop RAS Gateway VMs.
5. Stop SLB Mux VMs.
6. Restore the Network Controller with the `new-networkcontrollerrestore` cmdlet.

7. Check the restore **ProvisioningState** to know when the restore had completed successfully.
8. If using SCVMM, restore the SCVMM database using the backup that was created at the same time as the Network Controller backup.
9. If you want to restore workload VMs from backup, do that now.
10. Check the health of your system with the debug-networkcontrollerconfigurationstate cmdlet.

```
$cred = Get-Credential
Debug-NetworkControllerConfigurationState -NetworkController "https://NC.contoso.com" -Credential $cred

Fetching ResourceType: accessControlLists
Fetching ResourceType: servers
Fetching ResourceType: virtualNetworks
Fetching ResourceType: networkInterfaces
Fetching ResourceType: virtualGateways
Fetching ResourceType: loadbalancerMuxes
Fetching ResourceType: Gateways
```

Example: Restoring a Network Controller database

```
$URI = "https://NC.contoso.com"
$Credential = Get-Credential

$ShareUserResourceId = "BackupUser"
$ShareCredential = Get-NetworkControllerCredential -ConnectionURI $URI -Credential $Credential | Where
{$_.ResourceId -eq $ShareUserResourceId }

$RestoreProperties = New-Object Microsoft.Windows.NetworkController.NetworkControllerRestoreProperties
$RestoreProperties.RestorePath = "\\fileshare\backups\NetworkController\2017-04-25T16_53_13"
$RestoreProperties.Credential = $ShareCredential

$RestoreTime = (Get-Date).ToString("s").Replace(":", "_")
New-NetworkControllerRestore -ConnectionURI $URI -Credential $Credential -Properties $RestoreProperties -
ResourceId $RestoreTime -Force
```

Example: Checking the status of a Network Controller database restore

```
PS C:\ > get-networkcontrollerrestore -connectionuri $uri -credential $cred -ResourceId $restoreTime |
convertto-json -depth 10
{
  "Tags": null,
  "ResourceRef": "/networkControllerRestore/2017-04-26T15_04_44",
  "InstanceId": "22edecc8-a613-48ce-a74f-0418789f04f6",
  "Etag": "W/\\"f14f6b84-80a7-4b73-93b5-59a9c4b5d98e\\\"",
  "ResourceMetadata": null,
  "ResourceId": "2017-04-26T15_04_44",
  "Properties": {
    "RestorePath": "\\\\sa18fs\\\\sa18n22\\\\NetworkController\\\\2017-04-25T16_53_13",
    "ErrorMessage": null,
    "FailedResourcesList": null,
    "SuccessfulResourcesList": null,
    "ProvisioningState": "Succeeded",
    "Credential": null
  }
}
```

For information on configuration state messages that may appear, see [Troubleshoot the Windows Server 2016 Software Defined Networking Stack](#).

Security for SDN

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

You can use the topics in this section to learn about security in Software Defined Networking (SDN).

NOTE

For additional Software Defined Networking documentation, you can use the following library sections.

- [SDN Technologies](#)
- [Plan SDN](#)
- [Deploy SDN](#)
- [Manage SDN](#)
- [Troubleshoot SDN](#)

This section contains the following topics.

- [Network Controller Security](#)
- [Manage Certificates for Software Defined Networking](#)

Secure the Network Controller

9/21/2018 • 8 minutes to read • [Edit Online](#)

In this topic, you learn how to configure security for all communication between [Network Controller](#) and other software and devices.

The communication paths that you can secure include Northbound communication on the management plane, cluster communication between Network Controller virtual machines (VMs) in a cluster, and Southbound communication on the data plane.

1. **Northbound Communication.** Network Controller communicates on the management plane with SDN-capable management software like Windows PowerShell and System Center Virtual Machine Manager (SCVMM). These management tools provide you with the ability to define network policy and to create a goal state for the network, against which you can compare the actual network configuration to bring the actual configuration into parity with the goal state.
2. **Network Controller Cluster Communication.** When you configure three or more VMs as Network Controller cluster nodes, these nodes communicate with each other. This communication might be related to synchronizing and replication of data across nodes, or specific communication between Network Controller services.
3. **Southbound Communication.** Network Controller communicates on the data plane with SDN infrastructure and other devices like software load balancers, gateways, and host machines. You can use Network Controller to configure and manage these southbound devices so that they maintain the goal state that you have configured for the network.

Northbound Communication

Network Controller supports authentication, authorization, and encryption for Northbound communication. The following sections provide information on how to configure these security settings.

Authentication

When you configure authentication for Network Controller Northbound communication, you allow Network Controller cluster nodes and management clients to verify the identity of the device with which they are communicating.

Network Controller supports the following three modes of authentication between management clients and Network Controller nodes.

NOTE

If you are deploying Network Controller with System Center Virtual Machine Manager, only **Kerberos** mode is supported.

1. **Kerberos.** Use Kerberos authentication when joining both the management client and all Network Controller cluster nodes to an Active Directory domain. The Active Directory domain must have domain accounts used for authentication.
2. **X509.** Use X509 for certificate-based authentication for management clients not joined to an Active Directory domain. You must enroll certificates to all Network Controller cluster nodes and management clients. Also, all nodes and management clients must trust each others' certificates.
3. **None.** Use None for testing purposes in a test environment and, therefore, not recommended for use in a

production environment. When you choose this mode, there is no authentication performed between nodes and management clients.

You can configure the Authentication mode for Northbound communication by using the Windows PowerShell command **Install-NetworkController** with the *ClientAuthentication* parameter.

Authorization

When you configure authorization for Network Controller Northbound communication, you allow Network Controller cluster nodes and management clients to verify that the device with which they are communicating is trusted and has permission to participate in the communication.

Use the following authorization methods for each of the authentication modes supported by Network Controller.

1. **Kerberos.** When you are using the Kerberos authentication method, you define the users and computers authorized to communicate with Network Controller by creating a security group in Active Directory, and then adding the authorized users and computers to the group. You can configure Network Controller to use the security group for authorization by using the *ClientSecurityGroup* parameter of the **Install-NetworkController** Windows PowerShell command. After installing the Network Controller, you can change the security group by using the **Set-NetworkController** command with the parameter - *ClientSecurityGroup*. If using SCVMM, you must provide the security group as a parameter during deployment.
2. **X509.** When you are using the X509 authentication method, Network Controller only accepts requests from management clients whose certificate thumbprints are known to Network Controller. You can configure these thumbprints by using the *ClientCertificateThumbprint* parameter of the **Install-NetworkController** Windows PowerShell command. You can add other client thumbprints at any time by using the **Set-NetworkController** command.
3. **None.** When you choose this mode, there is no authentication performed between nodes and management clients. Use None for testing purposes in a test environment and, therefore, not recommended for use in a production environment.

Encryption

Northbound communication uses Secure Sockets Layer (SSL) to create an encrypted channel between management clients and Network Controller nodes. SSL encryption for Northbound communication includes the following requirements:

- All Network Controller nodes must have an identical certificate that includes the Server Authentication and Client Authentication purposes in Enhanced Key Usage (EKU) extensions.
- The URI used by management clients to communicate with Network Controller must be the certificate subject name. The certificate subject name must contain either the Fully Qualified Domain Name (FQDN) or the IP address of the Network Controller REST Endpoint.
- If Network Controller nodes are on different subnets, the subject name of their certificates must be the same as the value used for the *RestName* parameter in the **Install-NetworkController** Windows PowerShell command.
- All of the management clients must trust the SSL certificate.

SSL Certificate Enrollment and Configuration

You must manually enroll the SSL certificate on Network Controller nodes.

After the certificate is enrolled, you can configure Network Controller to use the certificate with the - **ServerCertificate** parameter of the **Install-NetworkController** Windows PowerShell command. If you have already installed Network Controller, you can update the configuration at any time by using the **Set-NetworkController** command.

NOTE

If you are using SCVMM, you must add the certificate as a library resource. For more information, see [Set up an SDN network controller in the VMM fabric](#).

Network Controller Cluster Communication

Network Controller supports authentication, authorization, and encryption for communication between Network Controller nodes. The communication is over [Windows Communication Foundation](#) (WCF) and TCP.

You can configure this mode with the **ClusterAuthentication** parameter of the **Install-NetworkControllerCluster** Windows PowerShell command.

For more information, see [Install-NetworkControllerCluster](#).

Authentication

When you configure authentication for Network Controller Cluster communication, you allow Network Controller cluster nodes to verify the identity of the other nodes with which they are communicating.

Network Controller supports the following three modes of authentication between Network Controller nodes.

NOTE

If you deploy Network Controller by using SCVMM, only **Kerberos** mode is supported.

1. **Kerberos**. You can use Kerberos authentication when all Network Controller cluster nodes are joined to an Active Directory domain, with domain accounts used for authentication.
2. **X509**. X509 is certificate-based authentication. You can use X509 authentication when Network Controller cluster nodes are not joined to an Active Directory domain. To use X509, you must enroll certificates to all Network Controller cluster nodes, and all nodes must trust the certificates. In addition, the subject name of the certificate that is enrolled on each node must be the same as the DNS name of the node.
3. **None**. When you choose this mode, there is no authentication performed between Network Controller nodes. This mode is provided only for testing purposes, and is not recommended for use in a production environment.

Authorization

When you configure authorization for Network Controller Cluster communication, you allow Network Controller cluster nodes to verify that the nodes with which they are communicating are trusted and have permission to participate in the communication.

For each of the authentication modes supported by Network Controller, the following authorization methods are used.

1. **Kerberos**. Network Controller nodes accept communication requests only from other Network Controller machine accounts. You can configure these accounts when you deploy Network Controller by using the **Name** parameter of the [New-NetworkControllerNodeObject](#) Windows PowerShell command.
2. **X509**. Network Controller nodes accept communication requests only from other Network Controller machine accounts. You can configure these accounts when you deploy Network Controller by using the **Name** parameter of the [New-NetworkControllerNodeObject](#) Windows PowerShell command.
3. **None**. When you choose this mode, there is no authorization performed between Network Controller nodes. This mode is provided only for testing purposes, and is not recommended for use in a production environment.

Encryption

Communication between Network Controller nodes is encrypted using WCF Transport level encryption. This form of encryption is used when the authentication and authorization methods are either Kerberos or X509 certificates. For more information, see the following topics.

- [How to: Secure a Service with Windows Credentials](#)
- [How to: Secure a Service with X.509 Certificates](#).

Southbound Communication

Network Controller interacts with different types of devices for Southbound communication. These interactions use different protocols. Because of this, there are different requirements for authentication, authorization, and encryption depending on the type of device and protocol used by Network Controller to communicate with the device.

The following table provides information about Network Controller interaction with different southbound devices.

| SOUTHBOUND DEVICE/SERVICE | PROTOCOL | AUTHENTICATION USED |
|---------------------------|-----------------------|------------------------|
| Software Load Balancer | WCF (MUX), TCP (Host) | Certificates |
| Firewall | OVSDB | Certificates |
| Gateway | WinRM | Kerberos, Certificates |
| Virtual Networking | OVSDB, WCF | Certificates |
| User defined routing | OVSDB | Certificates |

For each of these protocols, the communication mechanism is described in the following section.

Authentication

For Southbound communication, the following protocols and authentication methods are used.

1. **WCF/TCP/OVSDB.** For these protocols, authentication is performed by using X509 certificates. Both Network Controller and the peer Software Load Balancing (SLB) Multiplexer (MUX)/host machines present their certificates to each other for mutual authentication. Each certificate must be trusted by the remote peer.

For southbound authentication, you can use the same SSL certificate that is configured for encrypting the communication with the Northbound clients. You must also configure a certificate on the SLB MUX and host devices. The certificate subject name must be same as the DNS name of the device.

2. **WinRM.** For this protocol, authentication is performed by using Kerberos (for domain joined machines) and by using certificates (for non-domain joined machines).

Authorization

For Southbound communication, the following protocols and authorization methods are used.

1. **WCF/TCP.** For these protocols, authorization is based on the subject name of the peer entity. Network Controller stores the peer device DNS name, and uses it for authorization. This DNS name must match the subject name of the device in the certificate. Likewise, Network Controller certificate must match the Network Controller DNS name stored on the peer device.
2. **WinRM.** If Kerberos is being used, the WinRM client account must be present in a predefined group in Active Directory or in the Local Administrators group on the server. If certificates are being used, the client

presents a certificate to the server that the server authorizes using the subject name/issuer, and the server uses a mapped user account to perform authentication.

3. **OVSDB**. There is no authorization provided for this protocol.

Encryption

For Southbound communication, the following encryption methods are used for protocols.

1. **WCF/TCP/OVSDB**. For these protocols, encryption is performed using the certificate that is enrolled on the client or server.
2. **WinRM**. WinRM traffic is encrypted by default using Kerberos security support provider (SSP). You can configure Additional encryption, in the form of SSL, on the WinRM server.

Manage certificates for Software Defined Networking

9/21/2018 • 10 minutes to read • [Edit Online](#)

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

You can use this topic to learn how to manage certificates for Network Controller Northbound and Southbound communications when you deploy Software Defined Networking (SDN) in Windows Server 2016 Datacenter and you are using System Center Virtual Machine Manager (SCVMM) as your SDN management client.

NOTE

For overview information about Network Controller, see [Network Controller](#).

If you are not using Kerberos for securing the Network Controller communication, you can use X.509 certificates for authentication, authorization, and encryption.

SDN in Windows Server 2016 Datacenter supports both self-signed and Certification Authority (CA)-signed X.509 certificates. This topic provides step-by-step instructions for creating these certificates and applying them to secure Network Controller Northbound communication channels with management clients and Southbound communications with network devices, such as the Software Load Balancer (SLB). When you are using certificate-based authentication, you must enroll one certificate on Network Controller nodes that is used in the following ways.

1. Encrypting Northbound Communication with Secure Sockets Layer (SSL) between Network Controller nodes and management clients, such as System Center Virtual Machine Manager.
2. Authentication between Network Controller nodes and Southbound devices and services, such as Hyper-V hosts and Software Load Balancers (SLBs).

Creating and Enrolling an X.509 Certificate

You can create and enroll either a self-signed certificate or a certificate that is issued by a CA.

NOTE

When you are using SCVMM to deploy Network Controller, you must specify the X.509 certificate that is used to encrypt Northbound communications during the configuration of the Network Controller Service Template.

The certificate configuration must include the following values.

- The value for the **RestEndPoint** text box must either be the Network Controller Fully Qualified Domain Name (FQDN) or IP address.
- The **RestEndPoint** value must match the subject name (Common Name, CN) of the X.509 certificate.

Creating a Self-Signed X.509 Certificate

You can create a self-signed X.509 certificate and export it with the private key (protected with a password) by following these steps for single-node and multiple-node deployments of Network Controller.

When you create self-signed certificates, you can use the following guidelines.

- You can use the IP address of the Network Controller REST Endpoint for the DnsName parameter - but this is

not recommended because it requires that the Network Controller nodes are all located within a single management subnet (e.g. on a single rack)

- For multiple node NC deployments, the DNS name that you specify will become the FQDN of the Network Controller Cluster (DNS Host A records are automatically created.)
- For single node Network Controller deployments, the DNS name can be the Network Controller's host name followed by the full domain name.

Multiple node

You can use the [New-SelfSignedCertificate](#) Windows PowerShell command to create a self-signed certificate.

Syntax

```
New-SelfSignedCertificate -KeyUsageProperty All -Provider "Microsoft Strong Cryptographic Provider" -  
FriendlyName "<YourNCComputerName>" -DnsName @("<NCRESTName>")
```

Example usage

```
New-SelfSignedCertificate -KeyUsageProperty All -Provider "Microsoft Strong Cryptographic Provider" -  
FriendlyName "MultiNodeNC" -DnsName @("NCcluster.Contoso.com")
```

Single node

You can use the [New-SelfSignedCertificate](#) Windows PowerShell command to create a self-signed certificate.

Syntax

```
New-SelfSignedCertificate -KeyUsageProperty All -Provider "Microsoft Strong Cryptographic Provider" -  
FriendlyName "<YourNCComputerName>" -DnsName @("<NCFQDN>")
```

Example usage

```
New-SelfSignedCertificate -KeyUsageProperty All -Provider "Microsoft Strong Cryptographic Provider" -  
FriendlyName "SingleNodeNC" -DnsName @("SingleNodeNC.Contoso.com")
```

Creating a CA-Signed X.509 Certificate

To create a certificate by using a CA, you must have already deployed a Public Key Infrastructure (PKI) with Active Directory Certificate Services (AD CS).

NOTE

You can use third party CAs or tools, such as openssl, to create a certificate for use with Network Controller, however the instructions in this topic are specific to AD CS. To learn how to use a third party CA or tool, see the documentation for the software you are using.

Creating a certificate with a CA includes the following steps.

1. You or your organization's Domain or Security Administrator configures the certificate template
2. You or your organization's Network Controller Administrator or SCVMM Administrator requests a new certificate from the CA.

Certificate configuration requirements

While you are configuring a certificate template in the next step, ensure that the template you configure includes the following required elements.

1. The certificate subject name must be the FQDN of the Hyper-V host

2. The certificate must be placed in the local machine personal store (My – cert:\localmachine\my)
3. The certificate must have both Server Authentication (EKU: 1.3.6.1.5.5.7.3.1) and Client Authentication (EKU: 1.3.6.1.5.5.7.3.2) Application policies.

NOTE

If the Personal (My – cert:\localmachine\my) certificate store on the Hyper-V host has more than one X.509 certificate with Subject Name (CN) as the host Fully Qualified Domain Name (FQDN), ensure that the certificate that will be used by SDN has an additional custom Enhanced Key Usage property with the OID 1.3.6.1.4.1.311.95.1.1.1. Otherwise, the communication between Network Controller and the host might not work.

To configure the certificate template

NOTE

Before you perform this procedure, you should review the certificate requirements and the available certificate templates in the Certificate Templates console. You can either modify an existing template or create a duplicate of an existing template and then modify your copy of the template. Creating a copy of an existing template is recommended.

1. On the server where AD CS is installed, in Server Manager, click **Tools**, and then click **Certification Authority**. The Certification Authority Microsoft Management Console (MMC) opens.
2. In the MMC, double-click the CA name, right-click **Certificate Templates**, and then click **Manage**.
3. The Certificate Templates console opens. All of the certificate templates are displayed in the details pane.
4. In the details pane, click the template that you want to duplicate.
5. Click the **Action** menu, and then click **Duplicate Template**. The template **Properties** dialog box opens.
6. In the template **Properties** dialog box, on the **Subject Name** tab, click **Supply in the request**. (This setting is required for Network Controller SSL certificates.)
7. In the template **Properties** dialog box, on the **Request Handling** tab, ensure that **Allow private key to be exported** is selected. Also ensure that the **Signature and encryption** purpose is selected.
8. In the template **Properties** dialog box, on the **Extensions** tab, select **Key Usage**, and then click **Edit**.
9. In **Signature**, ensure that **Digital Signature** is selected.
10. In the template **Properties** dialog box, on the **Extensions** tab, select **Application Policies**, and then click **Edit**.
11. In **Application Policies**, ensure that **Client Authentication** and **Server Authentication** are listed.
12. Save the copy of the certificate template with a unique name, such as **Network Controller template**.

To request a certificate from the CA

You can use the Certificates snap-in to request certificates. You can request any type of certificate that has been preconfigured and made available by an administrator of the CA that processes the certificate request.

Users or local **Administrators** is the minimum group membership required to complete this procedure.

1. Open the Certificates snap-in for a computer.
2. In the console tree, click **Certificates (Local Computer)**. Select the **Personal** certificate store.
3. On the **Action** menu, point to** All Tasks, and then click **Request New Certificate to start the Certificate Enrollment wizard. Click **Next**.
4. Select the **Configured by your administrator** Certificate Enrollment Policy and click **Next**.
5. Select the **Active Directory Enrollment Policy** (based on the CA template that you configured in the previous section).
6. Expand the **Details** section and configure the following items.
 - a. Ensure that **Key usage** includes both **Digital Signature** **and **Key encipherment**.
 - b. Ensure that **Application policies** includes both **Server Authentication** (1.3.6.1.5.5.7.3.1) and **Client**

Authentication (1.3.6.1.5.5.7.3.2).

7. Click **Properties**.
8. On the **Subject** tab, in **Subject name**, in **Type**, select **Common name**. In Value, specify **Network Controller REST Endpoint**.
9. Click **Apply**, and then click **OK**.
10. Click **Enroll**.

In the Certificates MMC, click on the Personal store to view the certificate you have enrolled from the CA.

Exporting and Copying the Certificate to the SCVMM Library

After creating either a self-signed or CA-signed certificate, you must export the certificate with the private key (in .pfx format) and without the private key (in Base-64 .cer format) from the Certificates snap-in.

You must then copy the two exported files to the **ServerCertificate.cr** and **NCCertificate.cr** folders that you specified at the time when you imported the NC Service Template.

1. Open the Certificates snap-in (certlm.msc) and locate the certificate in the Personal certificate store for the local computer.
2. Right-click the certificate, click **All Tasks**, and then click **Export**. The Certificate Export Wizard opens. Click **Next**.
3. Select **Yes**, export the private key option, click **Next**.
4. Choose **Personal Information Exchange - PKCS #12 (.PFX)** and accept the default to **Include all certificates in the certification path if possible**.
5. Assign the Users/Groups and a password for the certificate you are exporting, click **Next**.
6. On the File to export page, browse the location where you want to place the exported file, and give it a name.
7. Similarly, export the certificate in .CER format. Note: To export to .CER format, uncheck the Yes, export the private key option.
8. Copy the .PFX to the ServerCertificate.cr folder.
9. Copy the .CER file to the NCCertificate.cr folder.

When you are done, refresh these folders in the SCVMM Library and ensure that you have these certificates copied. Continue with the Network Controller Service Template Configuration and Deployment.

Authenticating Southbound devices and services

Network Controller communication with hosts and SLB MUX devices uses certificates for authentication.

Communication with the hosts is over OVSDB protocol while communication with the SLB MUX devices is over the WCF protocol.

Hyper-V Host Communication with Network Controller

For communication with the Hyper-V hosts over OVSDB, Network Controller needs to present a certificate to the host machines. By default, SCVMM picks up the SSL certificate configured on the Network Controller and uses it for southbound communication with the hosts.

That is the reason why the SSL certificate must have the Client Authentication EKU configured. This certificate is configured on the "Servers" REST resource (Hyper-V hosts are represented in Network Controller as a Server resource), and can be viewed by running the Windows PowerShell command **Get-NetworkControllerServer**.

Following is a partial example of the server REST resource.

```

"resourceId": "host31.fabrikam.com",
"properties": {
    "connections": [
        {
            "managementAddresses": [
                "host31.fabrikam.com"
            ],
            "credential": {
                "resourceRef": "/credentials/a738762f-f727-43b5-9c50-cf82a70221fa"
            },
            "credentialType": "X509Certificate"
        }
    ],
},

```

For mutual authentication, the Hyper-V host must also have a certificate to communicate with Network Controller.

You can enroll the certificate from a Certification Authority (CA). If a CA based certificate is not found on the host machine, SCVMM creates a self-signed certificate and provisions it on the host machine.

Network Controller and the Hyper-V host certificates must be trusted by each other. The Hyper-V host certificate's root certificate must be present in the Network Controller Trusted Root Certification Authorities store for the Local Computer, and vice versa.

When you're using self-signed certificates, SCVMM ensures that the required certificates are present in the Trusted Root Certification Authorities store for the Local Computer.

If you are using CA based certificates for the Hyper-V hosts, you need to ensure that the CA root certificate is present on the Network Controller's Trusted Root Certification Authorities store for the Local Computer.

Software Load Balancer MUX Communication with Network Controller

The Software Load Balancer Multiplexor (MUX) and Network Controller communicate over the WCF protocol, using certificates for authentication.

By default, SCVMM picks up the SSL certificate configured on the Network Controller and uses it for southbound communication with the Mux devices. This certificate is configured on the "NetworkControllerLoadBalancerMux" REST resource and can be viewed by executing the Powershell cmdlet **Get-NetworkControllerLoadBalancerMux**.

Example of MUX REST resource (partial):

```

"resourceId": "slbmux1.fabrikam.com",
"properties": {
    "connections": [
        {
            "managementAddresses": [
                "slbmux1.fabrikam.com"
            ],
            "credential": {
                "resourceRef": "/credentials/a738762f-f727-43b5-9c50-cf82a70221fa"
            },
            "credentialType": "X509Certificate"
        }
    ],
},

```

For mutual authentication, you must also have a certificate on the SLB MUX devices. This certificate is automatically configured by SCVMM when you deploy software load balancer using SCVMM.

IMPORTANT

On the host and SLB nodes, it is critical that the Trusted Root Certification Authorities certificate store does not include any certificate where "Issued to" is not the same as "Issued by". If this occurs, communication between Network Controller and the southbound device fails.

Network Controller and the SLB MUX certificates must be trusted by each other (the SLB MUX certificate's root certificate must be present in the Network Controller machine Trusted Root Certification Authorities store and vice versa). When you're using self-signed certificates, SCVMM ensures that the required certificates are present in the Trusted Root Certification Authorities store for the Local Computer.

Kerberos with Service Principal Name (SPN)

9/21/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server 2019

Network Controller supports multiple authentication methods for communication with management clients. You can use Kerberos based authentication, X509 certificate-based authentication. You also have the option to use no authentication for test deployments.

System Center Virtual Machine Manager uses Kerberos-based authentication. If you are using Kerberos-based authentication, you must configure a Service Principal Name (SPN) for Network Controller in Active Directory. The SPN is a unique identifier for the Network Controller service instance, which is used by Kerberos authentication to associate a service instance with a service login account. For more details, see [Service Principal Names](#).

Configure Service Principal Names (SPN)

The Network Controller automatically configures the SPN. All you need to do is to provide permissions for the Network Controller machines to register and modify the SPN.

1. On the Domain Controller machine, start **Active Directory Users and Computers**.
2. Select **View > Advanced**.
3. Under **Computers**, locate one of the Network Controller machine accounts, and then right-click and select **Properties**.
4. Select the **Security** tab and click **Advanced**.
5. In the list, if all the Network Controller machine accounts or a security group having all the Network Controller machine accounts is not listed, click **Add** to add it.
6. For each Network Controller machine account or a single security group containing the Network Controller machine accounts:
 - a. Select the account or group and click **Edit**.
 - b. Under Permissions select **Validate Write servicePrincipalName**.
 - d. Scroll down and under **Properties** select:
 - **Read servicePrincipalName**
 - **Write servicePrincipalName**
- e. Click **OK** twice.
7. Repeat step 3 - 6 for each Network Controller machines.
8. Close **Active Directory Users and Computers**.

Failure to provide permissions for SPN registration/modification

On a **NEW** Windows Server 2019 deployment, if you chose Kerberos for REST client authentication and don't grant permission for Network Controller nodes to register or modify the SPN, REST operations on Network Controller fails preventing you from managing the SDN.

For an upgrade from Windows Server 2016 to Windows Server 2019, and you chose Kerberos for REST client authentication, REST operations do not get blocked, ensuring transparency for existing production deployments.

If SPN is not registered, REST client authentication uses NTLM, which is less secure. You also get a critical event in the Admin channel of **NetworkController-Framework** event channel asking you to provide permissions to the Network Controller nodes to register SPN. Once you provide permission, Network Controller registers the SPN automatically, and all client operations use Kerberos.

TIP

Typically, you can configure Network Controller to use an IP address or DNS name for REST-based operations. However, when you configure Kerberos, you cannot use an IP address for REST queries to Network Controller. For example, you can use <<https://networkcontroller.consotso.com>>, but you cannot use <<https://192.34.21.3>>. Service Principal Names cannot function if IP addresses are used.

If you were using IP address for REST operations along with Kerberos authentication in Windows Server 2016, the actual communication would have been over NTLM authentication. In such a deployment, once you upgrade to Windows Server 2019, you continue to use NTLM-based authentication. To move to Kerberos-based authentication, you must use Network Controller DNS name for REST operations and provide permission for Network Controller nodes to register SPN.

SDN firewall auditing

9/21/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server 2019

Firewall auditing is a new capability for the SDN firewall in Windows Server 2019. When you enable SDN firewall, any flow processed by SDN firewall rules (ACLs) that have logging enabled gets recorded. The log files must be in a syntax that is consistent with the [Azure Network Watcher flow logs](#). These logs can be used for diagnostics or archived for later analysis.

We will soon provide some examples of how to process these files using tools such as Power BI.

Try it out and provide us with feedback!

Here is a sample script to enable firewall auditing on the Hyper-V hosts. Update the variables at the beginning and run this on a Windows Server 2019 computer with the RSAT-NetworkController feature installed:

```
$logpath = "C:\test\log1"
$servers = @("sa18n22-2", "sa18n22-3", "sa18n22-4")
$uri = "https://sa18n22sdn.sa18.nttest.microsoft.com"

# Create log directories on the hosts
invoke-command -Computername $servers {
    param(
        $Path
    )
    mkdir $path -force
} -argumentlist $LogPath

# Set firewall auditing settings on Network Controller
$AuditProperties = new-object Microsoft.Windows.NetworkController.AuditingSettingsProperties
$AuditProperties.OutputDirectory = $logpath
set-networkcontrollerauditingsettingsconfiguration -connectionuri $uri -properties $AuditProperties -force | out-null

# Enable logging on each server
$servers = get-networkcontrollerserver -connectionuri $uri
foreach ($s in $servers) {
    $s.properties.AuditingEnabled = @("Firewall")
    new-networkcontrollerserver -connectionuri $uri -resourceid $s.resourceid -properties $s.properties -force | out-null
}
```

Once enabled a new file appears in the specified directory on each host about once per hour. You should periodically process these files and remove them from the hosts. The current file has zero length and is locked until flushed at the next hour mark:

```

PS C:\test\log1> dir

Directory: C:\test\log1

Mode                LastWriteTime         Length Name
----                -----          ----  -
-a----   7/19/2018  6:28 AM           17055 SdnFirewallAuditing.d8b3b697-5355-40e2-84d2-
1bf2f0e0dc4a.20180719TL122803093.json
-a----   7/19/2018  7:28 AM           7880 SdnFirewallAuditing.d8b3b697-5355-40e2-84d2-
1bf2f0e0dc4a.20180719TL132803173.json
-a----   7/19/2018  8:28 AM           7867 SdnFirewallAuditing.d8b3b697-5355-40e2-84d2-
1bf2f0e0dc4a.20180719TL142803264.json
-a----   7/19/2018  9:28 AM           10949 SdnFirewallAuditing.d8b3b697-5355-40e2-84d2-
1bf2f0e0dc4a.20180719TL152803360.json
-a----   7/19/2018  9:28 AM           0 SdnFirewallAuditing.d8b3b697-5355-40e2-84d2-
1bf2f0e0dc4a.20180719TL162803464.json

```

These files contain a sequence of flow events, for example:

```
{
  "records": [
    {
      "properties": {
        "Version": "1.0",
        "flows": [
          {
            "flows": [
              {
                "flowTuples": ["1531963580,192.122.0.22,192.122.255.255,138,138,U,I,A"],
                "portId": "9",
                "portName": "7290436D-0422-498A-8EB8-C6CF5115DACE"
              }
            ],
            "rule": "Allow_Inbound"
          }
        ]
      },
      "operationName": "NetworkSecurityGroupFlowEvents",
      "resourceId": "394f647d-2ed0-4c31-87c5-389b8c0c8132",
      "time": "20180719:L012620622",
      "category": "NetworkSecurityGroupFlowEvent",
      "systemId": "d8b3b697-5355-40e2-84d2-1bf2f0e0dc4a"
    },
  ]
}
```

Note, logging takes place only for rules that have **Logging** set to **Enabled**, for example:

```
{
  "Tags": null,
  "ResourceRef": "/accessControlLists/AllowAll",
  "InstanceId": "4a63e1a5-3264-4986-9a59-4e77a8b107fa",
  "Etag": "W/\"1535a780-0fc8-4bba-a15a-093ecac9b88b\"",
  "ResourceMetadata": null,
  "ResourceId": "AllowAll",
  "Properties": {
    "ConfigurationState": null,
    "ProvisioningState": "Succeeded",
    "AclRules": [
      {
        "ResourceMetadata": null,
        "ResourceRef":
        "/accessControlLists/AllowAll/aclRules/AllowAll_Inbound",
        "InstanceId": "ba8710a8-0f01-422b-9038-d1f2390645d7",
        "Etag": "W/\"1535a780-0fc8-4bba-a15a-093ecac9b88b\"",
        "ResourceId": "AllowAll_Inbound",
        "Name": "AllowAll_Inbound"
      }
    ]
  }
}
```

```

        "Properties": {
            "Protocol": "All",
            "SourcePortRange": "0-65535",
            "DestinationPortRange": "0-65535",
            "Action": "Allow",
            "SourceAddressPrefix": "*",
            "DestinationAddressPrefix": "*",
            "Priority": "101",
            "Description": null,
            "Type": "Inbound",
            "Logging": "Enabled",
            "ProvisioningState": "Succeeded"
        }
    },
    {
        "ResourceMetadata": null,
        "ResourceRef":
        "/accessControlLists/AllowAll/aclRules/AllowAll_Outbound",
        "InstanceId": "068264c6-2186-4dbc-bbe7-f504c6f47fa8",
        "Etag": "W/\"1535a780-0fc8-4bba-a15a-093ecac9b88b\"",
        "ResourceId": "AllowAll_Outbound",
        "Properties": {
            "Protocol": "All",
            "SourcePortRange": "0-65535",
            "DestinationPortRange": "0-65535",
            "Action": "Allow",
            "SourceAddressPrefix": "*",
            "DestinationAddressPrefix": "*",
            "Priority": "110",
            "Description": null,
            "Type": "Outbound",
            "Logging": "Enabled",
            "ProvisioningState": "Succeeded"
        }
    }
],
"IpConfigurations": [
    ],
"Subnets": [
    {
        "ResourceMetadata": null,
        "ResourceRef": "/virtualNetworks/10_0_1_0/subnets/Subnet1",
        "InstanceId": "00000000-0000-0000-0000-000000000000",
        "Etag": null,
        "ResourceId": null,
        "Properties": null
    }
]
}
}

```

Virtual network peering

9/21/2018 • 3 minutes to read • [Edit Online](#)

Applies to: Windows Server

Virtual network peering lets you connect two virtual networks seamlessly. Once peered, for connectivity purposes, the virtual networks appear as one.

The benefits of using virtual network peering include:

- Traffic between virtual machines in the peered virtual networks gets routed through the backbone infrastructure through *private* IP addresses only. The communication between the virtual networks does not require public Internet or gateways.
- A low-latency, high-bandwidth connection between resources in different virtual networks.
- The ability for resources in one virtual network to communicate with resources in a different virtual network.
- No downtime to resources in either virtual network when creating the peering.

Requirements and constraints

Virtual network peering has a few requirements and constraints:

- Peered virtual networks must:
 - Have non-overlapping IP address spaces
 - Be managed by the same Network Controller
- Once you peer a virtual network with another virtual network, you cannot add or delete address ranges in the address space.

TIP

If you need to add address ranges:

1. Remove the peering.
2. Add the address space.
3. Add the peering again.

- Since virtual network peering is between two virtual networks, there is no derived transitive relationship across peerings. For example, if you peer virtualNetworkA with virtualNetworkB and virtualNetworkB with virtualNetworkC, then virtualNetworkA does not get peered with virtualNetworkC.

[image here]

Connectivity

After you peer virtual networks, resources in either virtual network can directly connect with resources in the peered virtual network.

- Network latency between virtual machines in peered virtual networks is the same as the latency within a

single virtual network.

- Network throughput is based on the bandwidth allowed for the virtual machine. There isn't any additional restriction on bandwidth within the peering.
- Traffic between virtual machines in peered virtual networks is routed directly through the backbone infrastructure, not through a gateway or over the public Internet.
- Virtual machines in a virtual network can access the internal load-balancer in the peered virtual network.

You can apply access control lists (ACLs) in either virtual network to block access to other virtual networks or subnets if desired. If you open full connectivity between peered virtual networks (which is the default option), you can apply ACLs to specific subnets or virtual machines to block or deny specific access. To learn more about ACLs, see [Use Access Control Lists \(ACLs\) to Manage Datacenter Network Traffic Flow](#).

Service chaining

You can configure user-defined routes that point to virtual machines in peered virtual networks as the next hop IP address, to enable service chaining. Service chaining enables you to direct traffic from one virtual network to a virtual appliance, in a peered virtual network, through user-defined routes.

You can deploy hub-and-spoke networks, where the hub virtual network can host infrastructure components such as a network virtual appliance. All the spoke virtual networks peer with the hub virtual network. Traffic can flow through network virtual appliances in the hub virtual network.

Virtual network peering enables the next hop in a user-defined route to be the IP address of a virtual machine in the peered virtual network. To learn more about user-defined routes, see [Use Network Virtual Appliances on a Virtual Network](#).

Gateways and on-premises connectivity

Each virtual network, regardless of whether peered with another virtual network, can still have its own gateway to connect to an on-premises network. When you peer virtual networks, you can also configure the gateway in the peered virtual network as a transit point to an on-premises network. In this case, the virtual network that uses a remote gateway cannot have its own gateway. A virtual network can have only one gateway that can be either a local or remote gateway (in the peered virtual network).

Monitor

When you peer two virtual networks, you must configure a peering for each virtual network in the peering.

You can monitor the status of your peering connection, which can be in one of the following states:

- **Initiated:** Shown when you create the peering from the first virtual network to the second virtual network.
- **Connected:** Shown after you've created the peering from the second virtual network to the first virtual network. The peering state for the first virtual network changes from Initiated to Connected. Both virtual network peers must have the state of Connected before establishing a virtual network peering successfully.
- **Disconnected:** Shown if one virtual network disconnects from another virtual network.

[infographic of the states]

Next steps

[Configure the virtual network peering](#): In this procedure, you use Windows PowerShell to find the HNV provider logical network to create two virtual networks, each with one subnet. You also configure the peering between the

two virtual networks.

Configure virtual network peering

9/21/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server

In this procedure, you use Windows PowerShell to create two virtual networks, each with one subnet. Then, you configure peering between the two virtual networks to enable connectivity between them.

- [Step 1. Create the first virtual network](#)
- [Step 2. Create the second virtual network](#)
- [Step 3. Configure peering from the first virtual network to the second virtual network](#)
- [Step 4. Configure peering from the second virtual network to the first virtual network](#)

IMPORTANT

Remember to update the properties for your environment.

Step 1. Create the first virtual network

In this step, you use Windows PowerShell find the HNV provider logical network to create the first virtual network with one subnet. The following example script creates Contoso's virtual network with one subnet.

```
#Find the HNV Provider Logical Network

$logicalnetworks = Get-NetworkControllerLogicalNetwork -ConnectionUri $uri
foreach ($ln in $logicalnetworks) {
    if ($ln.Properties.NetworkVirtualizationEnabled -eq "True") {
        $HNVProviderLogicalNetwork = $ln
    }
}

#Create the Virtual Subnet

$vsubnet = new-object Microsoft.Windows.NetworkController.VirtualSubnet
$vsubnet.ResourceId = "Contoso"
$vsubnet.Properties = new-object Microsoft.Windows.NetworkController.VirtualSubnetProperties
$vsubnet.Properties.AddressPrefix = "24.30.1.0/24"
$uri="https://restserver"

#Create the Virtual Network

$vnetproperties = new-object Microsoft.Windows.NetworkController.VirtualNetworkProperties
$vnetproperties.AddressSpace = new-object Microsoft.Windows.NetworkController.AddressSpace
$vnetproperties.AddressSpace.AddressPrefixes = @("24.30.1.0/24")
$vnetproperties.LogicalNetwork = $HNVProviderLogicalNetwork
$vnetproperties.Subnets = @($vsubnet)
New-NetworkControllerVirtualNetwork -ResourceId "Contoso_VNet1" -ConnectionUri $uri -Properties
$vnetproperties
```

Step 2. Create the second virtual network

In this step, you create a second virtual network with one subnet. The following example script creates

Woodgrove's virtual network with one subnet.

```
#Create the Virtual Subnet

$vsubnet = new-object Microsoft.Windows.NetworkController.VirtualSubnet
$vsubnet.ResourceId = "Woodgrove"
$vsubnet.Properties = new-object Microsoft.Windows.NetworkController.VirtualSubnetProperties
$vsubnet.Properties.AddressPrefix = "24.30.2.0/24"
$uri="https://restserver"

#Create the Virtual Network

$vnetproperties = new-object Microsoft.Windows.NetworkController.VirtualNetworkProperties
$vnetproperties.AddressSpace = new-object Microsoft.Windows.NetworkController.AddressSpace
$vnetproperties.AddressSpace.AddressPrefixes = @("24.30.2.0/24")
$vnetproperties.LogicalNetwork = $HNVProviderLogicalNetwork
$vnetproperties.Subnets = @($vsubnet)
New-NetworkControllerVirtualNetwork -ResourceId "Woodgrove_VNet1" -ConnectionUri $uri -Properties
$vnetproperties
```

Step 3. Configure peering from the first virtual network to the second virtual network

In this step, you configure the peering between the first virtual network and the second virtual network you created in the previous two steps. The following example script establishes virtual network peering from **Contoso_vnet1** to **Woodgrove_vnet1**.

```
$peeringProperties = New-Object Microsoft.Windows.NetworkController.VirtualNetworkPeeringProperties
$vnet2 = Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId "Woodgrove_VNet1"
$peeringProperties.remoteVirtualNetwork = $vnet2

#Indicate whether communication between the two virtual networks
$peeringProperties.allowVirtualnetworkAccess = $true

#Indicates whether forwarded traffic is allowed across the vnets
$peeringProperties.allowForwardedTraffic = $true

#Indicates whether the peer virtual network can access this virtual networks gateway
$peeringProperties.allowGatewayTransit = $false

#Indicates whether this virtual network uses peer virtual networks gateway
$peeringProperties.useRemoteGateways = $false

New-NetworkControllerVirtualNetworkPeering -ConnectionUri $uri -VirtualNetworkId "Contoso_vnet1" -ResourceId
"ContosotoWoodgrove" -Properties $peeringProperties
```

IMPORTANT

After creating this peering, the vnet status shows **Initiated**.

Step 4. Configure peering from the second virtual network to the first virtual network

In this step, you configure the peering between the second virtual network and the first virtual network you created in steps 1 and 2 above. The following example script establishes virtual network peering from **Woodgrove_vnet1** to **Contoso_vnet1**.

```
$peeringProperties = New-Object Microsoft.Windows.NetworkController.VirtualNetworkPeeringProperties
$vnet2=Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId "Contoso_VNet1"
$peeringProperties.remoteVirtualNetwork = $vnet2

# Indicates whether communication between the two virtual networks is allowed
$peeringProperties.allowVirtualnetworkAccess = $true

# Indicates whether forwarded traffic will be allowed across the vnets
$peeringProperties.allowForwardedTraffic = $true

# Indicates whether the peer virtual network can access this virtual network's gateway
$peeringProperties.allowGatewayTransit = $false

# Indicates whether this virtual network will use peer virtual network's gateway
$peeringProperties.useRemoteGateways = $false

New-NetworkControllerVirtualNetworkPeering -ConnectionUri $uri -VirtualNetworkId "Woodgrove_vnet1" -ResourceId "WoodgrovettoContoso" -Properties $peeringProperties
```

After creating this peering, the vnet peering status shows **Connected** for both the peers. Now, virtual machines in one virtual network can communicate with virtual machines in the peered virtual network.

Egress metering in virtual network

9/21/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server

A fundamental aspect of cloud networking monetization is network bandwidth egress. For example - outbound Data transfers In Microsoft Azure business model. Outbound data is charged based on the total amount of data moving out of the Azure datacenters via the Internet in a given billing cycle.

This new feature in Windows Server 2019 enables SDN to offer usage meters for outbound data transfers. With this feature added, Network Controller keeps a whitelist per Virtual Network of all IP ranges used within SDN, and consider any packet bound for a destination that is not included in one of these ranges to be billed outbound data transfers.

Virtual network unbilled address ranges (whitelist of IP ranges)

You can find unbilled address ranges under the **UnbilledAddressRanges** property of an existing virtual Network. By default, there is no address ranges added.

```
PS C:\> Get-NetworkControllerVirtualNetwork -ConnectionUri $u -ResourceId TestVNET01 | convertto-json -depth 10
{
    "Tags": null,
    "ResourceRef": "/virtualNetworks/TestVNET01",
    "InstanceId": "da9c53ca-95e6-40b5-8f8e-00d0fb829ecb",
    "Etag": "W/"f6937910-850a-4bab-954c-dc2db7ee2ab9"",
    "ResourceMetadata": null,
    "ResourceId": "TestVNET01",
    "Properties": {
        "AddressSpace": {
            "AddressPrefixes": [
                "172.16.1.0/24"
            ]
        },
        "DhcpOptions": null,
        "UnbilledAddressRanges": null,
        "ConfigurationState": null,
        "ProvisioningState": "Succeeded",
        "Subnets": [
            {
                "ResourceMetadata": null,
                "ResourceRef": "/virtualNetworks/TestVNET01/subnets/TestSubnet01",
                "InstanceId": "d52cf771-a64f-4fee-a435-f4d94d11cd38",
                "Etag": "W/"f6937910-850a-4bab-954c-dc2db7ee2ab9"",
                "ResourceId": "TestSubnet01",
                "Properties": {
                    "AddressPrefix": "172.16.1.0/24",
                    "AddressSpace": {
                        "AddressPrefixes": [
                            "172.16.1.0/24"
                        ]
                    }
                }
            }
        ]
    }
}
```

Manage the unbilled address ranges of a virtual network

You can manage which data transfer/traffic to the destination IPs defined in the **UnbilledAddressRange** property of a virtual network.

NOTE

These destination IPs do not get metered.

1. Create a **UnbilledAddressRanges** property.

```
$unbilled = (Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId VirtualNetworkResourceID)
$unbilled.Properties.UnbilledAddressRanges = "10.10.2.0/24, 192.168.1.0/24 ... "
```

TIP

If adding multiple IP ranges, use a comma between each of the IP ranges.

2. Update the Virtual Network **UnbilledAddressRanges** property.

```
New-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId VirtualNetworkResourceID -Properties $unbilled.Properties
```

```
PS C:\> New-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId TestVNET01 -Properties $unbilled.Properties
Confirm
Performing the operation 'New-NetworkControllerVirtualNetwork' on entities of type 'Microsoft.Windows.NetworkController.VirtualNetwork'
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): Y

Tags          :
ResourceRef   : /virtualNetworks/TestVNET01
InstanceId    : da9c53ca-95e6-40b5-8f8e-00d0fb829ecb
Etag          : W/"9ad8d735-2fbf-4ecf-84be-b2d3206576f7"
ResourceMetadata :
ResourceId    : TestVNET01
Properties     : Microsoft.Windows.NetworkController.VirtualNetworkProperties
```

3. Check the Virtual Network to see the configured **UnbilledAddressRanges**.

```
Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId VirtualNetworkResourceID
```

```
PS C:\Windows\system32> Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId TESTVNET | con
{
    "Tags": null,
    "ResourceRef": "/virtualNetworks/TESTVNET",
    "InstanceId": "20fbcc6eb-e248-4ba6-a2bc-a697cc7a49cc",
    "Etag": "W/"1facc6ed-f6bc-46b0-9416-7405270c1f03"",
    "ResourceMetadata": null,
    "ResourceId": "TESTVNET",
    "Properties": {
        "AddressSpace": {
            "AddressPrefixes": [
                "10.10.1.0/24"
            ]
        },
        "DhcpOptions": null,
        "UnbilledAddressRanges": "10.10.2.0/24",
        "ConfigurationState": null,
        "ProvisioningState": "Succeeded",
        "Subnets": [
            {
                "ResourceMetadata": null,
                "ResourceRef": "/virtualNetworks/TESTVNET/subnets/Subnet1",
                "InstanceId": "4ea00515-0284-47e9-b188-5d890a4f8e89",
                "Etag": "W/"1facc6ed-f6bc-46b0-9416-7405270c1f03"",
                "ResourceId": "Subnet1",
                "Properties": {
                    "AddressPrefix": "10.10.1.0/24",
                    "UnbilledAddressRanges": "10.10.2.0/24"
                }
            }
        ]
    }
}
```

Check the billed the unbilled egress usage of a virtual network

After you configure the **UnbilledAddressRanges** property, you can check the billed and unbilled egress usage of a virtual network. Egress traffic updates every four minutes with the total bytes of the billed and unbilled ranges.

- **UnbilledEgressBytes** under the *Properties* of *Subnets* shows the number of unbilled bytes sent by virtual machines with network interfaces with IP configurations from this virtual subnet. Unbilled bytes are bytes sent to address ranges that are part of the **UnbilledAddressRanges** property of the parent virtual network.
- **BilledEgressBytes** under the *Properties* of *Subnets* shows Number of billed bytes sent by virtual machines

with network interfaces with IP configurations from this virtual subnet. Billed bytes are bytes sent to address ranges that are not part of the UnbilledAddressRanges property of the parent virtual network.

```
PS C:\Windows\system32> Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId TESTVNET | convertfrom-json
{
    "Tags": null,
    "ResourceRef": "/virtualNetworks/TESTVNET",
    "InstanceId": "20fb6eb-e248-4ba6-a2bc-a697cc7a49cc",
    "Etag": "W/\"1facc6ed-f6bc-46b0-9416-7405270c1f03\"",
    "ResourceMetadata": null,
    "ResourceId": "TESTVNET",
    "Properties": {
        "AddressSpace": {
            "AddressPrefixes": [
                "10.10.1.0/24"
            ]
        },
        "DhcpOptions": null,
        "UnbilledAddressRanges": "10.10.2.0/24",
        "ConfigurationState": null,
        "ProvisioningState": "Succeeded",
        "Subnets": [
            {
                "ResourceMetadata": null,
                "ResourceRef": "/virtualNetworks/TESTVNET/subnets/Subnet1",
                "InstanceId": "4eae0515-0284-47e9-b188-5d890a4f8e89",
                "Etag": "W/\"1facc6ed-f6bc-46b0-9416-7405270c1f03\"",
                "ResourceId": "Subnet1",
                "Properties": {
                    "AddressPrefix": "10.10.1.0/24",
                    "VirtualSubnetId": "4168",
                    "UnbilledEgressBytes": 0,
                    "BilledEgressBytes": 13408408,
                    "EncryptionEnabled": false,
                    "ProvisioningState": "Succeeded",
                    "AccessControlList": null,
                    "RouteTable": null,
                    "DualStackSubnet": null,
                    "ServiceInsertion": null,
                    "IpConfigurations": [
                        {
                            "ResourceMetadata": null
                        }
                    ]
                }
            }
        ]
    }
}
```

Windows Server 2019 Gateway Performance

9/21/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server

In Windows Server 2016, one of the customer concerns was the inability of SDN gateway to meet the throughput requirements of modern networks. The network throughput of IPsec and GRE tunnels had limitations with the single connection throughput for IPsec connectivity being about 300 Mbps and for GRE connectivity being about 2.5 Gbps.

We have improved significantly in Windows Server 2019, with the numbers soaring to 1.8 Gbps and 15 Gbps for IPsec and GRE connections, respectively. All this, with significant reductions in the CPU cycles/per byte, thereby providing ultra-high-performance throughput with much less CPU utilization.

Enable high performance with gateways in Windows Server 2019

For **GRE connections**, once you deploy/upgrade to Windows Server 2019 builds on the gateway VMs, you should automatically see the improved performance. No manual steps are involved.

For **IPsec connections**, by default, when you create the connection for your virtual networks, you get the Windows Server 2016 data path and performance numbers. To enable the Windows Server 2019 data path, do the following:

1. On an SDN gateway VM, go to **Services** console (services.msc).
2. Find the service named **Azure Gateway Service**, and set the startup type to **Automatic**.
3. Restart the gateway VM. The active connections on this gateway failover to a redundant gateway VM.
4. Repeat the previous steps for rest of the gateway VMs.

TIP

For the best performance results, ensure that the cipherTransformationConstant and authenticationTransformConstant in quickMode settings of the IPsec connection uses the **GCMAES256** cipher suite.

For maximum performance, the gateway host hardware must support AES-NI and PCLMULQDQ CPU instruction sets. These are available on any Westmere (32nm) and later Intel CPU except on models where AES-NI has been disabled. You can look at your hardware vendor documentation to see if the CPU supports AES-NI and PCLMULQDQ CPU instruction sets.

Below is a REST sample of IPsec connection with optimal security algorithms:

```

# NOTE: The virtual gateway must be created before creating the IPsec connection. More details here.
# Create a new object for Tenant Network IPsec Connection
$nwConnectionProperties = New-Object Microsoft.Windows.NetworkController.NetworkConnectionProperties

# Update the common object properties
$nwConnectionProperties.ConnectionType = "IPSec"
$nwConnectionProperties.OutboundKiloBitsPerSecond = 2000000
$nwConnectionProperties.InboundKiloBitsPerSecond = 2000000

# Update specific properties depending on the Connection Type
$nwConnectionProperties.IpSecConfiguration = New-Object Microsoft.Windows.NetworkController.IpSecConfiguration
$nwConnectionProperties.IpSecConfiguration.AuthenticationMethod = "PSK"
$nwConnectionProperties.IpSecConfiguration.SharedSecret = "111_aaa"

$nwConnectionProperties.IpSecConfiguration.QuickMode = New-Object Microsoft.Windows.NetworkController.QuickMode
$nwConnectionProperties.IpSecConfiguration.QuickMode.PerfectForwardSecrecy = "PFS2048"
$nwConnectionProperties.IpSecConfiguration.QuickMode.AuthenticationTransformationConstant = "GCMAES256"
$nwConnectionProperties.IpSecConfiguration.QuickMode.CipherTransformationConstant = "GCMAES256"
$nwConnectionProperties.IpSecConfiguration.QuickMode.SALifeTimeSeconds = 3600
$nwConnectionProperties.IpSecConfiguration.QuickMode.IdleDisconnectSeconds = 500
$nwConnectionProperties.IpSecConfiguration.QuickMode.SALifeTimeKiloBytes = 2000

$nwConnectionProperties.IpSecConfiguration.MainMode = New-Object Microsoft.Windows.NetworkController.MainMode
$nwConnectionProperties.IpSecConfiguration.MainMode.DiffieHellmanGroup = "Group2"
$nwConnectionProperties.IpSecConfiguration.MainMode.IntegrityAlgorithm = "SHA256"
$nwConnectionProperties.IpSecConfiguration.MainMode.EncryptionAlgorithm = "AES256"
$nwConnectionProperties.IpSecConfiguration.MainMode.SALifeTimeSeconds = 28800
$nwConnectionProperties.IpSecConfiguration.MainMode.SALifeTimeKiloBytes = 2000

# L3 specific configuration (leave blank for IPSec)
$nwConnectionProperties.IPADressees = @()
$nwConnectionProperties.PeerIPADressees = @()

# Update the IPv4 Routes that are reachable over the site-to-site VPN Tunnel
$nwConnectionProperties.Routes = @()
$ipv4Route = New-Object Microsoft.Windows.NetworkController.RouteInfo
$ipv4Route.DestinationPrefix = "<<On premise subnet that must be reachable over the VPN tunnel. Ex:  
10.0.0.0/24>>"
$ipv4Route.metric = 10
$nwConnectionProperties.Routes += $ipv4Route

# Tunnel Destination (Remote Endpoint) Address
$nwConnectionProperties.DestinationIPAddress = "<<Public IP address of the On-Premise VPN gateway. Ex:  
192.168.3.4>>"

# Add the new Network Connection for the tenant. Note that the virtual gateway must be created before creating
the IPsec connection. $uri is the REST URI of your deployment and must be in the form of "https://<REST URI>"  

New-NetworkControllerVirtualGatewayNetworkConnection -ConnectionUri $uri -VirtualGatewayId  

$virtualGW.ResourceId -ResourceId "Contoso_IPSecGW" -Properties $nwConnectionProperties -Force

```

Testing Results

We have done extensive performance testing for the SDN gateways in our test labs. In the tests, we have compared gateway network performance with Windows Server 2019 in SDN scenarios and non-SDN scenarios. You can find the results and test setup details captured in the blog article [here](#).

Gateway bandwidth allocation

9/21/2018 • 3 minutes to read • [Edit Online](#)

Applies to: Windows Server

In Windows Server 2016, the individual tunnel bandwidth for IPsec, GRE, and L3 was a ratio of the total gateway capacity. Therefore, customers would provide the gateway capacity based on the standard TCP bandwidth expecting this out of the gateway VM.

Also, maximum IPsec tunnel bandwidth on the gateway was limited to $(3/20) * \text{Gateway Capacity}$ provided by the customer. So, for example, if you set the gateway capacity to 100 Mbps, then the IPsec tunnel capacity would be 150 Mbps. The equivalent ratios for GRE and L3 tunnels are 1/5 and 1/2, respectively.

Although this worked for the majority of the deployments, the fixed ratio model was not appropriate for high throughput environments. Even when the data transfer rates were high (say, higher than 40 Gbps), the maximum throughput of SDN gateway tunnels capped due to internal factors.

In Windows Server 2019, for a tunnel type, the maximum throughput is fixed:

- IPsec = 5 Gbps
- GRE = 15 Gbps
- L3 = 5 Gbps

So, even if your gateway host/VM supports NICs with much higher throughput, the maximum available tunnel throughput is fixed. Another issue this takes care of is arbitrarily over-provisioning gateways, which happens when providing a very high number for the gateway capacity.

Gateway capacity calculation

Ideally, you set the gateway throughput capacity to the throughput available to the gateway VM. So, for example, if you have a single gateway VM and the underlying host NIC throughput is 25 Gbps, the gateway throughput can be set to 25 Gbps as well.

If using a gateway only for IPsec connections, the maximum available fixed capacity is 5 Gbps. So, for example, if you provision IPsec connections on the gateway, you can only provision to an aggregate bandwidth (incoming + outgoing) as 5 Gbps.

If using the gateway for both IPsec and GRE connectivity, you can provision maximum 5 Gbps of IPsec throughput or maximum 15 Gbps of GRE throughput. So, for example, if you provision 2 Gbps of IPsec throughput, you have 3 Gbps of IPsec throughput left to provision on the gateway or 9 Gbps of GRE throughput left.

To put this in more mathematical terms:

- Total capacity of the gateway = 25 Gbps
- Total available IPsec capacity = 5 Gbps (fixed)
- Total available GRE capacity = 15 Gbps (fixed)
- IPsec throughput ratio for this gateway = $25/5 = 5$ Gbps
- GRE throughput ratio for this gateway = $25/15 = 5/3$ Gbps

For example, if you allocate 2 Gbps of IPsec throughput to a customer:

Remaining available capacity on the gateway = Total capacity of the gateway – IPsec throughput ratio*IPsec throughput allocated (used capacity)

$$25 - 5 \times 2 = 15 \text{ Gbps}$$

Remaining IPsec throughput that you can allocate on the gateway

$$5 - 2 = 3 \text{ Gbps}$$

Remaining GRE throughput that you can allocate on the gateway = Remaining capacity of gateway/GRE throughput ratio

$$15 \times 3 / 5 = 9 \text{ Gbps}$$

The throughput ratio varies depending on the total capacity of the gateway. One thing to note is that you should set the total capacity to the TCP bandwidth available to the gateway VM. If you have multiple VMs hosted on the gateway, you must adjust the total capacity of the gateway accordingly.

Also, if the gateway capacity is less than the total available tunnel capacity, the total available tunnel capacity is set to the gateway capacity. For example, if you set the gateway capacity to 4 Gbps, the total available capacity for IPsec, L3, and GRE is set to 4 Gbps, leaving the throughput ratio for each tunnel to 1 Gbps.

Windows Server 2016 behavior

The gateway capacity calculation algorithm for Windows Server 2016 remains unchanged. In Windows Server 2016, Maximum IPsec tunnel bandwidth was limited to $(3/20) * \text{gateway capacity}$ on a gateway. The equivalent ratios for GRE and L3 tunnels were 1/5 and 1/2, respectively.

If you are upgrading from Windows Server 2016 to Windows Server 2019:

- GRE and L3 tunnels:** The Windows Server 2019 allocation logic takes effect once Network Controller nodes get updated to Windows Server 2019
- IPSec tunnels:** The Windows Server 2016 gateway allocation logic continues to function until all the gateways in the gateway pool get upgraded to Windows Server 2019. For all gateways in the gateway pool, you must set the Azure gateway service to **Automatic**.

NOTE

It is possible that after upgrading to Windows Server 2019, a gateway becomes over-provisioned (as the allocation logic changes from Windows Server 2016 to Windows Server 2019). In this case, the existing connections on the gateway continue to exist. The REST resource for the Gateway throws a warning that the gateway is over-provisioned. In this case, you should move some connections to another gateway.

Troubleshoot SDN

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

The topics in this section provide information about troubleshooting the Software Defined Networking (SDN) technologies that are included in Windows Server 2016.

NOTE

For additional Software Defined Networking documentation, you can use the following library sections.

- [SDN Technologies](#)
- [Plan SDN](#)
- [Deploy SDN](#)
- [Manage SDN](#)
- [Security for SDN](#)

This section contains the following topics.

- [Troubleshoot the Windows Server Software Defined Networking Stack](#)
- Blog post [Troubleshoot Configuring SDN RAS Gateway VPN Bandwidth Settings in Virtual Machine Manager](#)
- Blog post [SDN Troubleshooting: Find the Local SDN RAS Gateway Server IP Address](#)
- Blog post [SDN Troubleshooting: UDP Communication and Changing Network Controller Cert](#)

Troubleshoot the Windows Server Software Defined Networking Stack

9/1/2018 • 28 minutes to read • [Edit Online](#)

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

This guide examines the common Software Defined Networking (SDN) errors and failure scenarios and outlines a troubleshooting workflow that leverages the available diagnostic tools.

For more information about Microsoft's Software Defined Networking, see [Software Defined Networking](#).

Error types

The following list represents the class of problems most often seen with Hyper-V Network Virtualization (HNVv1) in Windows Server 2012 R2 from in-market production deployments and coincides in many ways with the same types of problems seen in Windows Server 2016 HNVv2 with the new Software Defined Network (SDN) Stack.

Most errors can be classified into a small set of classes:

- **Invalid or unsupported configuration**

A user invokes the NorthBound API incorrectly or with invalid policy.

- **Error in policy application**

Policy from Network Controller was not delivered to a Hyper-V Host, significantly delayed and / or not up to date on all Hyper-V hosts (for example, after a Live Migration).

- **Configuration drift or software bug**

Data-path issues resulting in dropped packets.

- **External error related to NIC hardware / drivers or the underlay network fabric**

Misbehaving task offloads (such as VMQ) or underlay network fabric misconfigured (such as MTU)

This troubleshooting guide examines each of these error categories and recommends best practices and diagnostic tools available to identify and fix the error.

Diagnostic tools

Before discussing the troubleshooting workflows for each of these type of errors, let's examine the diagnostic tools available.

To use the Network Controller (control-path) diagnostic tools, you must first install the RSAT-NetworkController feature and import the `NetworkControllerDiagnostics` module:

```
Add-WindowsFeature RSAT-NetworkController -IncludeManagementTools  
Import-Module NetworkControllerDiagnostics
```

To use the HNV Diagnostics (data-path) diagnostic tools, you must import the `HNVDiagnistics` module:

```
# Assumes RSAT-NetworkController feature has already been installed  
Import-Module hnvdiagnostics
```

Network controller diagnostics

These cmdlets are documented on TechNet in the [Network Controller Diagnostics Cmdlet Topic](#). They help identify problems with network policy consistency in the control-path between Network Controller nodes and between the Network Controller and the NC Host Agents running on the Hyper-V hosts.

The *Debug-ServiceFabricNodeStatus* and *Get-NetworkControllerReplica* cmdlets must be run from one of the Network Controller node virtual machines. All other NC Diagnostic cmdlets can be run from any host which has connectivity to the Network Controller and is in either in the Network Controller Management security group (Kerberos) or has access to the X.509 certificate for managing the Network Controller.

Hyper-V host diagnostics

These cmdlets are documented on TechNet in the [Hyper-V Network Virtualization \(HNV\) Diagnostics Cmdlet Topic](#). They help identify problems in the data-path between tenant virtual machines (East/West) and ingress traffic through an SLB VIP (North/South).

The *Debug-VirtualMachineQueueOperation*, *Get-CustomerRoute*, *Get-PACAMapping*, *Get-ProviderAddress*, *Get-VMNetworkAdapterPortId*, *Get-VMSwitchExternalPortId*, and *Test-EncapOverheadSettings* are all local tests which can be run from any Hyper-V host. The other cmdlets invoke data-path tests through the Network Controller and therefore need access to the Network Controller as described above.

GitHub

The [Microsoft/SDN GitHub Repo](#) has a number of sample scripts and workflows which build on top of these inbox cmdlets. In particular, diagnostic scripts can be found in the [Diagnostics](#) folder. Please help us contribute to these scripts by submitting Pull Requests.

Troubleshooting Workflows and Guides

[Hoster] Validate System Health

There is an embedded resource named *Configuration State* in several of the Network Controller resources. Configuration state provides information about system health including the consistency between the network controller's configuration and the actual (running) state on the Hyper-V hosts.

To check configuration state, run the following from any Hyper-V host with connectivity to the Network Controller.

NOTE

The value for the *NetworkController* parameter should either be the FQDN or IP address based on the subject name of the X.509 >certificate created for Network Controller.

The *Credential* parameter only needs to be specified if the network controller is using Kerberos authentication (typical in VMM deployments). The credential must be for a user who is in the Network Controller Management Security Group.

```
Debug-NetworkControllerConfigurationState -NetworkController <FQDN or NC IP> [-Credential <PS Credential>]

# Healthy State Example - no status reported
$cred = Get-Credential
Debug-NetworkControllerConfigurationState -NetworkController 10.127.132.211 -Credential $cred

Fetching ResourceType: accessControlLists
Fetching ResourceType: servers
Fetching ResourceType: virtualNetworks
Fetching ResourceType: networkInterfaces
Fetching ResourceType: virtualGateways
Fetching ResourceType: loadbalancerMuxes
Fetching ResourceType: Gateways
```

A sample Configuration State message is shown below:

```
Fetching ResourceType: servers
-----
ResourcePath: https://10.127.132.211/Networking/v1/servers/4c4c4544-0056-4b10-8058-b8c04f395931
Status: Warning

Source: SoftwareLoadBalancerManager
Code: HostNotConnectedToController
Message: Host is not Connected.
```

NOTE

There is a bug in the system where the Network Interface resources for the SLB Mux Transit VM NIC are in a Failure state with error "Virtual Switch - Host Not Connected To Controller". This error can be safely ignored if the IP configuration in the VM NIC resource is set to an IP Address from the Transit Logical Network's IP Pool. There is a second bug in the system where the Network Interface resources for the Gateway HNV Provider VM NICs are in a Failure state with error "Virtual Switch - PortBlocked". This error can also be safely ignored if the IP configuration in the VM NIC resource is set to null (by design).

The table below shows the list of error codes, messages, and follow-up actions to take based on the configuration state observed.

| CODE | MESSAGE | ACTION |
|---------------------------------|---|---|
| Unknown | Unknown error | |
| HostUnreachable | The host machine is not reachable | Check the Management network connectivity between Network Controller and Host |
| PAIpAddressExhausted | The PA Ip addresses exhausted | Increase the HNV Provider logical subnet's IP Pool Size |
| PAMacAddressExhausted | The PA Mac addresses exhausted | Increase the Mac Pool Range |
| PAAddressConfigurationFailure | Failed to plumb PA addresses to the host | Check the Management network connectivity between Network Controller and Host. |
| CertificateNotTrusted | Certificate is not trusted | Fix the certificates used for communication with the host. |
| CertificateNotAuthorized | Certificate not authorized | Fix the certificates used for communication with the host. |
| PolicyConfigurationFailureOnVfp | Failure in configuring VFP policies | This is a runtime failure. No definite work arounds. Collect logs. |
| PolicyConfigurationFailure | Failure in pushing policies to the hosts, due to communication failures or others error in the NetworkController. | No definite actions. This is due to failure in goal state processing in the Network Controller modules. Collect logs. |

| Code | Message | Action |
|---|---|--|
| HostNotConnectedToController | The Host is not yet connected to the Network Controller | Port Profile not applied on the host or the host is not reachable from the Network Controller. Validate that HostID registry key matches the Instance ID of the server resource |
| MultipleVfpEnabledSwitches | There are multiple VFp enabled Switches on the host | Delete one of the switches, since Network Controller Host Agent only supports one vSwitch with the VFP extension enabled |
| PolicyConfigurationFailure | Failed to push VNet policies for a VmNic due to certificate errors or connectivity errors | Check if proper certificates have been deployed (Certificate subject name must match FQDN of host). Also verify the host connectivity with the Network Controller |
| PolicyConfigurationFailure | Failed to push vSwitch policies for a VmNic due to certificate errors or connectivity errors | Check if proper certificates have been deployed (Certificate subject name must match FQDN of host). Also verify the host connectivity with the Network Controller |
| PolicyConfigurationFailure | Failed to push Firewall policies for a VmNic due to certificate errors or connectivity errors | Check if proper certificates have been deployed (Certificate subject name must match FQDN of host). Also verify the host connectivity with the Network Controller |
| DistributedRouterConfigurationFailure | Failed to configure the Distributed router settings on the host vNic | TCPIP stack error. May require cleaning up the PA and DR Host vNICs on the server on which this error was reported |
| DhcpAddressAllocationFailure | DHCP address allocation failed for a VMNic | Check if the static IP address attribute is configured on the NIC resource |
| CertificateNotTrusted
CertificateNotAuthorized | Failed to connect to Mux due to network or cert errors | Check the numeric code provided in the error message code: this corresponds to the winsock error code. Certificate errors are granular (for example, cert cannot be verified, cert not authorized, etc.) |
| HostUnreachable | MUX is Unhealthy (Common case is BGPRouter disconnected) | BGP peer on the RRAS (BGP virtual machine) or Top-of-Rack (ToR) switch is unreachable or not peering successfully. Check BGP settings on both Software Load Balancer Multiplexer resource and BGP peer (ToR or RRAS virtual machine) |
| HostNotConnectedToController | SLB host agent is not connected | Check that SLB Host Agent service is running; Refer to SLB host agent logs (auto running) for reasons why, in case SLBM (NC) rejected the cert presented by the host agent running state will show nuanced information |

| CODE | MESSAGE | ACTION |
|--------------------------|---|--|
| PortBlocked | The VFP port is blocked, due to lack of VNET / ACL policies | Check if there are any other errors, which might cause the policies to be not configured. |
| Overloaded | Loadbalancer MUX is overloaded | Performance issue with MUX |
| RoutePublicationFailure | Loadbalancer MUX is not connected to a BGP router | Check if the MUX has connectivity with the BGP routers and that BGP peering is setup correctly |
| VirtualServerUnreachable | Loadbalancer MUX is not connected to SLB manager | Check connectivity between SLBM and MUX |
| QosConfigurationFailure | Failed to configure QOS policies | See if sufficient bandwidth is available for all VM's if QOS reservation is used |

Check network connectivity between the network controller and Hyper-V Host (NC Host Agent service)

Run the `netstat` command below to validate that there are three ESTABLISHED connections between the NC Host Agent and the Network Controller node(s) and one LISTENING socket on the Hyper-V Host

- LISTENING on port TCP:6640 on Hyper-V Host (NC Host Agent Service)
- Two established connections from Hyper-V host IP on port 6640 to NC node IP on ephemeral ports (> 32000)
- One established connection from Hyper-V host IP on ephemeral port to Network Controller REST IP on port 6640

NOTE

There may only be two established connections on a Hyper-V host if there are no tenant virtual machines deployed on that particular host.

```
netstat -anp tcp |findstr 6640

# Successful output
TCP    0.0.0.0:6640          0.0.0.0:0              LISTENING
TCP    10.127.132.153:6640   10.127.132.213:50095  ESTABLISHED
TCP    10.127.132.153:6640   10.127.132.214:62514  ESTABLISHED
TCP    10.127.132.153:50023  10.127.132.211:6640   ESTABLISHED
```

Check Host Agent services

The network controller communicates with two host agent services on the Hyper-V hosts: SLB Host Agent and NC Host Agent. It is possible that one or both of these services is not running. Check their state and restart if they're not running.

```
Get-Service SlbHostAgent
Get-Service NcHostAgent

# (Re)start requires -Force flag
Start-Service NcHostAgent -Force
Start-Service SlbHostAgent -Force
```

Check health of network controller

If there are not three ESTABLISHED connections or if the Network Controller appears unresponsive, check to see that all nodes and service modules are up and running by using the following cmdlets.

```
# Prints a DIFF state (status is automatically updated if state is changed) of a particular service module
replica
Debug-ServiceFabricNodeStatus [-ServiceTypeName] <Service Module>
```

The network controller service modules are:

- ControllerService
- ApiService
- SlbManagerService
- ServiceInsertion
- FirewallService
- VSwitchService
- GatewayManager
- FnmService
- HelperService
- UpdateService

Check that ReplicaStatus is **Ready** and HealthState is **Ok**.

In a production deployment is with a multi-node Network Controller, you can also check which node each service is primary on and its individual replica status.

```
Get-NetworkControllerReplica

# Sample Output for the API service module
Replicas for service: ApiService

ReplicaRole    : Primary
NodeName       : SA18N30NC3.sa18.nttest.microsoft.com
ReplicaStatus  : Ready
```

Check that the Replica Status is Ready for each service.

Check for corresponding HostIDs and certificates between network controller and each Hyper-V Host

On a Hyper-V Host, run the following commands to check that the HostID corresponds to the Instance Id of a server resource on the Network Controller

```
Get-ItemProperty "hklm:\system\currentcontrolset\services\nchostagent\parameters" -Name HostId | fl HostId

HostId : **162cd2c8-08d4-4298-8cb4-10c2977e3cf**

Get-NetworkControllerServer -ConnectionUri $uri |where { $_.InstanceId -eq "162cd2c8-08d4-4298-8cb4-10c2977e3cf" }

Tags          :
ResourceRef   : /servers/4c4c4544-0056-4a10-8059-b8c04f395931
InstanceId    : **162cd2c8-08d4-4298-8cb4-10c2977e3cf**
Etag          : W/"50f89b08-215c-495d-8505-0776baab9cb3"
ResourceMetadata : Microsoft.Windows.NetworkController.ResourceMetadata
ResourceId     : 4c4c4544-0056-4a10-8059-b8c04f395931
Properties      : Microsoft.Windows.NetworkController.ServerProperties
```

Remediation If using SDNExpress scripts or manual deployment, update the HostId key in the registry to match the Instance Id of the server resource. Restart the Network Controller Host Agent on the Hyper-V host (physical server) If using VMM, delete the Hyper-V Server from VMM and remove the HostId registry key. Then, re-add the server through VMM.

Check that the thumbprints of the X.509 certificates used by the Hyper-V host (the hostname will be the cert's Subject Name) for (SouthBound) communication between the Hyper-V Host (NC Host Agent service) and Network Controller nodes are the same. Also check that the Network Controller's REST certificate has subject name of CN=.

```
# On Hyper-V Host
dir cert:\\localmachine\\my

Thumbprint           Subject
-----
2A3A674D07D8D7AE11EBDAC25B86441D68D774F9  CN=SA18n30-4.sa18.nttest.microsoft.com
...
dir cert:\\localmachine\\root

Thumbprint           Subject
-----
30674C020268AA4E40FD6817BA6966531FB9ADA4  CN=10.127.132.211 *** NC REST IP ADDRESS***

# On Network Controller Node VM
dir cert:\\localmachine\\root

Thumbprint           Subject
-----
2A3A674D07D8D7AE11EBDAC25B86441D68D774F9  CN=SA18n30-4.sa18.nttest.microsoft.com
30674C020268AA4E40FD6817BA6966531FB9ADA4  CN=10.127.132.211 *** NC REST IP ADDRESS***
...
```

You can also check the following parameters of each cert to make sure the subject name is what is expected (hostname or NC REST FQDN or IP), the certificate has not yet expired, and that all certificate authorities in the certificate chain are included in the trusted root authority.

- Subject Name
- Expiration Date
- Trusted by Root Authority

Remediation If multiple certificates have the same subject name on the Hyper-V host, the Network Controller Host Agent will randomly choose one to present to the Network Controller. This may not match the thumbprint of the server resource known to the Network Controller. In this case, delete one of the certificates with the same subject name on the Hyper-V host and then re-start the Network Controller Host Agent service. If a connection can still not be made, delete the other certificate with the same subject name on the Hyper-V Host and delete the corresponding server resource in VMM. Then, re-create the server resource in VMM which will generate a new X.509 certificate and install it on the Hyper-V host.

Check the SLB Configuration State

The SLB Configuration State can be determined as part of the output to the Debug-NetworkController cmdlet. This cmdlet will also output the current set of Network Controller resources in JSON files, all IP configurations from each Hyper-V host (server) and local network policy from Host Agent database tables.

Additional traces will be collected by default. To not collect traces, add the -IncludeTraces:\$false parameter.

```
Debug-NetworkController -NetworkController <FQDN or IP> [-Credential <PS Credential>] [-IncludeTraces:$false]

# Don't collect traces
$cred = Get-Credential
Debug-NetworkController -NetworkController 10.127.132.211 -Credential $cred -IncludeTraces:$false

Transcript started, output file is C:\\NCDiagnostics.log
Collecting Diagnostics data from NC Nodes
```

NOTE

The default output location will be the <working_directory>\NCDiagnostics\ directory. The default output directory can be changed by using the `-OutputDirectory` parameter.

The SLB Configuration State information can be found in the *diagnostics-slbstateResults.Json* file in this directory.

This JSON file can be broken down into the following sections:

- Fabric
 - SlbmVips - This section lists the IP address of the SLB Manager VIP address which is used by the Network Controller to coordinate configuration and health between the SLB Muxes and SLB Host Agents.
 - MuxState - This section will list one value for each SLB Mux deployed giving the state of the mux
 - Router Configuration - This section will list the Upstream Router's (BGP Peer) Autonomous System Number (ASN), Transit IP Address, and ID. It will also list the SLB Muxes ASN and Transit IP.
 - Connected Host Info - This section will list the Management IP address all of the Hyper-V hosts available to run load-balanced workloads.
 - Vip Ranges - This section will list the public and private VIP IP pool ranges. The SLBM VIP will be included as an allocated IP from one of these ranges.
 - Mux Routes - This section will list one value for each SLB Mux deployed containing all of the Route Advertisements for that particular mux.
- Tenant
 - VipConsolidatedState - This section will list the connectivity state for each Tenant VIP including advertised route prefix, Hyper-V Host and DIP endpoints.

NOTE

SLB State can be ascertained directly by using the [DumpSlbRestState](#) script available on the [Microsoft SDN GitHub repository](#).

Gateway Validation**From Network Controller:**

```
Get-NetworkControllerLogicalNetwork  
Get-NetworkControllerPublicIPAddress  
Get-NetworkControllerGatewayPool  
Get-NetworkControllerGateway  
Get-NetworkControllerVirtualGateway  
Get-NetworkControllerNetworkInterface
```

From Gateway VM:

```
Ipconfig /allcompartments /all  
Get-NetRoute -IncludeAllCompartments -AddressFamily  
Get-NetBgpRouter  
Get-NetBgpRouter | Get-BgpPeer  
Get-NetBgpRouter | Get-BgpRouteInformation
```

From Top of Rack (ToR) Switch:

```
sh ip bgp summary (for 3rd party BGP Routers)
```

Windows BGP Router

```
Get-BgpRouter  
Get-BgpPeer  
Get-BgpRouteInformation
```

In addition to these, from the issues we have seen so far (especially on SDNExpress based deployments), the most common reason for Tenant Compartment not getting configured on GW VMs seem to be the fact that the GW Capacity in FabricConfig.psd1 is less compared to what folks try to assign to the Network Connections (S2S Tunnels) in TenantConfig.psd1. This can be checked easily by comparing outputs of the following commands:

```
PS > (Get-NetworkControllerGatewayPool -ConnectionUri $uri).properties.Capacity  
PS > (Get-NetworkControllerVirtualgatewayNetworkConnection -ConnectionUri $uri -VirtualGatewayId  
"TenantName").properties.OutboundKiloBitsPerSecond  
PS > (Get-NetworkControllerVirtualgatewayNetworkConnection -ConnectionUri $uri -VirtualGatewayId  
"TenantName").property
```

[Hoster] Validate Data-Plane

After the Network Controller has been deployed, tenant virtual networks and subnets have been created, and VMs have been attached to the virtual subnets, additional fabric level tests can be performed by the hoster to check tenant connectivity.

Check HNV Provider Logical Network Connectivity

After the first guest VM running on a Hyper-V host has been connected to a tenant virtual network, the Network Controller will assign two HNV Provider IP Addresses (PA IP Addresses) to the Hyper-V Host. These IPs will come from the HNV Provider logical network's IP Pool and be managed by the Network Controller. To find out what these two HNV IP Addresses are 's

```
PS > Get-ProviderAddress  
  
# Sample Output  
ProviderAddress : 10.10.182.66  
MAC Address     : 40-1D-D8-B7-1C-04  
Subnet Mask     : 255.255.255.128  
Default Gateway : 10.10.182.1  
VLAN           : VLAN11  
  
ProviderAddress : 10.10.182.67  
MAC Address     : 40-1D-D8-B7-1C-05  
Subnet Mask     : 255.255.255.128  
Default Gateway : 10.10.182.1  
VLAN           : VLAN11
```

These HNV Provider IP Addresses (PA IPs) are assigned to Ethernet Adapters created in a separate TCPIP network compartment and have an adapter name of *VLANX* where X is the VLAN assigned to the HNV Provider (transport) logical network.

Connectivity between two Hyper-V hosts using the HNV Provider logical network can be done by a ping with an additional compartment (-c Y) parameter where Y is the TCPIP network compartment in which the PAhostVNics are created. This compartment can be determined by executing:

```
C:\> ipconfig /allcompartments /all

<snip> ...

=====
Network Information for *Compartment 3*
=====

  Host Name . . . . . : SA18n30-2
<snip> ...

Ethernet adapter VLAN11:

  Connection-specific DNS Suffix . :
  Description . . . . . : Microsoft Hyper-V Network Adapter
  Physical Address. . . . . : 40-1D-D8-B7-1C-04
  DHCP Enabled. . . . . : No
  Autoconfiguration Enabled . . . . . : Yes
  Link-local IPv6 Address . . . . . : fe80::5937:a365:d135:2899%39(Preferred)
  IPv4 Address. . . . . : 10.10.182.66(Preferred)
  Subnet Mask . . . . . : 255.255.255.128
  Default Gateway . . . . . : 10.10.182.1
  NetBIOS over Tcpip. . . . . : Disabled

Ethernet adapter VLAN11:

  Connection-specific DNS Suffix . :
  Description . . . . . : Microsoft Hyper-V Network Adapter
  Physical Address. . . . . : 40-1D-D8-B7-1C-05
  DHCP Enabled. . . . . : No
  Autoconfiguration Enabled . . . . . : Yes
  Link-local IPv6 Address . . . . . : fe80::28b3:1ab1:d9d9:19ec%44(Preferred)
  IPv4 Address. . . . . : 10.10.182.67(Preferred)
  Subnet Mask . . . . . : 255.255.255.128
  Default Gateway . . . . . : 10.10.182.1
  NetBIOS over Tcpip. . . . . : Disabled

*Ethernet adapter vEthernet (PAhostVNic):*
<snip> ...
```

NOTE

The PA Host vNIC Adapters are not used in the data-path and so do not have an IP assigned to the "vEthernet (PAhostVNic) adapter".

For instance, assume that Hyper-V hosts 1 and 2 have HNV Provider (PA) IP Addresses of:

| - HYPER-V HOST - | - PA IP ADDRESS 1 | - PA IP ADDRESS 2 |
|------------------|-------------------|-------------------|
| Host 1 | 10.10.182.64 | 10.10.182.65 |
| Host 2 | 10.10.182.66 | 10.10.182.67 |

we can ping between the two using the following command to check HNV Provider logical network connectivity.

```

# Ping the first PA IP Address on Hyper-V Host 2 from the first PA IP address on Hyper-V Host 1 in compartment
(-c) 3
C:\> ping -c 3 10.10.182.66 -S 10.10.182.64

# Ping the second PA IP Address on Hyper-V Host 2 from the first PA IP address on Hyper-V Host 1 in compartment
(-c) 3
C:\> ping -c 3 10.10.182.67 -S 10.10.182.64

# Ping the first PA IP Address on Hyper-V Host 2 from the second PA IP address on Hyper-V Host 1 in compartment
(-c) 3
C:\> ping -c 3 10.10.182.66 -S 10.10.182.65

# Ping the second PA IP Address on Hyper-V Host 2 from the second PA IP address on Hyper-V Host 1 in
compartment (-c) 3
C:\> ping -c 3 10.10.182.67 -S 10.10.182.65

```

Remediation If HNV Provider ping does not work, check your physical network connectivity including VLAN configuration. The physical NICs on each Hyper-V host should be in trunk mode with no specific VLAN assigned. The Management Host vNIC should be isolated to the Management Logical Network's VLAN.

```

PS C:\> Get-NetAdapter "Ethernet 4" |fl

Name : Ethernet 4
InterfaceDescription : <NIC> Ethernet Adapter
InterfaceIndex : 2
MacAddress : F4-52-14-55-BC-21
MediaType : 802.3
PhysicalMediaType : 802.3
InterfaceOperationalStatus : Up
AdminStatus : Up
LinkSpeed(Gbps) : 10
MediaConnectionState : Connected
ConnectorPresent : True
*VlanID : 0*
DriverInformation : Driver Date 2016-08-28 Version 5.25.12665.0 NDIS 6.60

# VMM uses the older PowerShell cmdlet <Verb>-VMNetworkAdapterVlan to set VLAN isolation
PS C:\> Get-VMNetworkAdapterVlan -ManagementOS -VMNetworkAdapterName <Mgmt>

VMName VMNetworkAdapterName Mode VlanList
----- -----
<snip> ...
    Mgmt          Access    7
<snip> ...

# SDNExpress deployments use the newer PowerShell cmdlet <Verb>-VMNetworkAdapterIsolation to set VLAN isolation
PS C:\> Get-VMNetworkAdapterIsolation -ManagementOS

<snip> ...

IsolationMode : Vlan
AllowUntaggedTraffic : False
DefaultIsolationID : 7
MultiTenantStack : Off
ParentAdapter : VMInternalNetworkAdapter, Name = 'Mgmt'
IsTemplate : True
CimSession : CimSession: .
ComputerName : SA18N30-2
IsDeleted : False

<snip> ...

```

Check MTU and Jumbo Frame support on HNV Provider Logical Network

Another common problem in the HNV Provider logical network is that the physical network ports and/or Ethernet

card do not have a large enough MTU configured to handle the overhead from VXLAN (or NVGRE) encapsulation.

NOTE

Some Ethernet cards and drivers support the new *EncapOverhead keyword which will automatically be set by the Network Controller Host Agent to a value of 160. This value will then be added to the value of the *JumboPacket keyword whose summation is used as the advertised MTU. e.g. *EncapOverhead = 160 and *JumboPacket = 1514 => MTU = 1674B

```
# Check whether or not your Ethernet card and driver support *EncapOverhead
PS C:\ > Test-EncapOverheadSettings

Verifying Physical Nic : <NIC> Ethernet Adapter #2
Physical Nic <NIC> Ethernet Adapter #2 can support SDN traffic. Encapoverhead value set on the nic is 160
Verifying Physical Nic : <NIC> Ethernet Adapter
Physical Nic <NIC> Ethernet Adapter can support SDN traffic. Encapoverhead value set on the nic is 160
```

To test whether or not the HNV Provider logical network supports the larger MTU size end-to-end, use the *Test-LogicalNetworkSupportsJumboPacket* cmdlet:

```
# Get credentials for both source host and destination host (or use the same credential if in the same domain)
$sourcehostcred = Get-Credential
$desthostcred = Get-Credential

# Use the Management IP Address or FQDN of the Source and Destination Hyper-V hosts
Test-LogicalNetworkSupportsJumboPacket -SourceHost sa18n30-2 -DestinationHost sa18n30-3 -SourceHostCreds
$sourcehostcred -DestinationHostCreds $desthostcred

# Failure Results
SourceCompartment : 3
pinging Source PA: 10.10.182.66 to Destination PA: 10.10.182.64 with Payload: 1632
pinging Source PA: 10.10.182.66 to Destination PA: 10.10.182.64 with Payload: 1472
Checking if physical nics support jumbo packets on host
Physical Nic <NIC> Ethernet Adapter #2 can support SDN traffic. Encapoverhead value set on the nic is 160
Cannot send jumbo packets to the destination. Physical switch ports may not be configured to support jumbo
packets.
Checking if physical nics support jumbo packets on host
Physical Nic <NIC> Ethernet Adapter #2 can support SDN traffic. Encapoverhead value set on the nic is 160
Cannot send jumbo packets to the destination. Physical switch ports may not be configured to support jumbo
packets.

# TODO: Success Results after updating MTU on physical switch ports
```

Remediation

- Adjust the MTU size on the physical switch ports to be at least 1674B (including 14B Ethernet header and trailer)
- If your NIC card does not support the EncapOverhead keyword, adjust the JumboPacket keyword to be at least 1674B

Check Tenant VM NIC connectivity

Each VM NIC assigned to a guest VM has a CA-PA mapping between the private Customer Address (CA) and the HNV Provider Address (PA) space. These mappings are kept in the OVSDB server tables on each Hyper-V host and can be found by executing the following cmdlet.

```
# Get all known PA-CA Mappings from this particular Hyper-V Host
PS > Get-PACAMapping

CA IP Address CA MAC Address Virtual Subnet ID PA IP Address
----- ----- -----
10.254.254.2 00-1D-D8-B7-1C-43 4115 10.10.182.67
10.254.254.3 00-1D-D8-B7-1C-43 4115 10.10.182.67
192.168.1.5 00-1D-D8-B7-1C-07 4114 10.10.182.65
10.254.254.1 40-1D-D8-B7-1C-06 4115 10.10.182.66
192.168.1.1 40-1D-D8-B7-1C-06 4114 10.10.182.66
192.168.1.4 00-1D-D8-B7-1C-05 4114 10.10.182.66
```

NOTE

If the CA-PA mappings you expect are not output for a given tenant VM, please check the VM NIC and IP Configuration resources on the Network Controller using the *Get-NetworkControllerNetworkInterface* cmdlet. Also, check the established connections between the NC Host Agent and Network Controller nodes.

With this information, a tenant VM ping can now be initiated by the Hoster from the Network Controller using the *Test-VirtualNetworkConnection* cmdlet.

Specific Troubleshooting Scenarios

The following sections provide guidance for troubleshooting specific scenarios.

No network connectivity between two tenant virtual machines

- [Tenant] Ensure Windows Firewall in tenant virtual machines is not blocking traffic.
- [Tenant] Check that IP addresses have been assigned to the tenant virtual machine by running *ipconfig*.
- [Hoster] Run **Test-VirtualNetworkConnection** from the Hyper-V host to validate connectivity between the two tenant virtual machines in question.

NOTE

The VSID refers to the Virtual Subnet ID. In the case of VXLAN, this is the VXLAN Network Identifier (VNI). You can find this value by running the **Get-PACAMapping** cmdlet.

Example

```
$password = ConvertTo-SecureString -String "password" -AsPlainText -Force
$cred = New-Object pscredential -ArgumentList (".\administrator", $password)
```

Create CA-ping between "Green Web VM 1" with SenderCA IP of 192.168.1.4 on Host "sa18n30-2.sa18.nttest.microsoft.com" with Mgmt IP of 10.127.132.153 to ListenerCA IP of 192.168.1.5 both attached to Virtual Subnet (VSID) 4114.

```
Test-VirtualNetworkConnection -OperationId 27 -HostName sa18n30-2.sa18.nttest.microsoft.com -MgmtIp
10.127.132.153 -Creds $cred -VMName "Green Web VM 1" -VMNetworkAdapterName "Green Web VM 1" -SenderCAIP
192.168.1.4 -SenderVSID 4114 -ListenerCAIP 192.168.1.5 -ListenerVSID 4114
```

```
Test-VirtualNetworkConnection at command pipeline position 1
```

Starting CA-space ping test Starting trace session Ping to 192.168.1.5 succeeded from address 192.168.1.4 Rtt = 0 ms

CA Routing Information:

```
Local IP: 192.168.1.4
Local VSID: 4114
Remote IP: 192.168.1.5
Remote VSID: 4114
Distributed Router Local IP: 192.168.1.1
Distributed Router Local MAC: 40-1D-D8-B7-1C-06
Local CA MAC: 00-1D-D8-B7-1C-05
Remote CA MAC: 00-1D-D8-B7-1C-07
Next Hop CA MAC Address: 00-1D-D8-B7-1C-07
```

PA Routing Information:

```
Local PA IP: 10.10.182.66
Remote PA IP: 10.10.182.65
```

...

1. [Tenant] Check that there is no distributed firewall policies specified on the virtual subnet or VM network interfaces which would block traffic.

Query the Network Controller REST API found in demo environment at sa18n30nc in the sa18.nttest.microsoft.com domain.

```
$uri = "https://sa18n30nc.sa18.nttest.microsoft.com"
Get-NetworkControllerAccessControlList -ConnectionUri $uri
```

Look at IP Configuration and Virtual Subnets which are referencing this ACL

1. [Hoster] Run `Get-ProviderAddress` on both Hyper-V hosts hosting the two tenant virtual machines in question and then run `Test-LogicalNetworkConnection` or `ping -c <compartment>` from the Hyper-V host to validate connectivity on the HNV Provider logical network
2. [Hoster] Ensure that the MTU settings are correct on the Hyper-V hosts and any Layer-2 switching devices in between the Hyper-V Hosts. Run `Test-EncapOverheadValue` on all Hyper-V hosts in question. Also check that all Layer-2 switches in between have MTU set to least 1674 bytes to account for maximum overhead of 160 bytes.
3. [Hoster] If PA IP Addresses are not present and/or CA Connectivity is broken, check to ensure network policy has been received. Run `Get-PACAMapping` to see if the encapsulation rules and CA-PA mappings required for creating overlay virtual networks are correctly established.
4. [Hoster] Check that the Network Controller Host Agent is connected to the Network Controller. Run `netstat -anp tcp |findstr 6640` to see if the
5. [Hoster] Check that the Host ID in HKLM/ matches the Instance ID of the server resources hosting the tenant virtual machines.
6. [Hoster] Check that the Port Profile ID matches the Instance ID of the VM Network Interfaces of the tenant virtual machines.

Logging, Tracing and advanced diagnostics

The following sections provide information on advanced diagnostics, logging, and tracing.

Network controller centralized logging

The Network Controller can automatically collect debugger logs and store them in a centralized location. Log

collection can be enabled when you deploy the Network Controller for the first time or any time later. The logs are collected from the Network Controller, and network elements managed by Network Controller: host machines, software load balancers (SLB) and gateway machines.

These logs include debug logs for the Network Controller cluster, the Network Controller application, gateway logs, SLB, virtual networking and the distributed firewall. Whenever a new host/SLB/gateway is added to the Network Controller, logging is started on those machines. Similarly, when a host/SLB/gateway is removed from the Network Controller, logging is stopped on those machines.

Enable logging

Logging is automatically enabled when you install the Network Controller cluster using the **Install-NetworkControllerCluster** cmdlet. By default, the logs are collected locally on the Network Controller nodes at `%systemdrive%\SDNDiagnostics`. It is **STRONGLY RECOMMENDED** that you change this location to be a remote file share (not local).

The Network Controller cluster logs are stored at `%programData%\Windows Fabric\log\Traces`. You can specify a centralized location for log collection with the **DiagnosticLogLocation** parameter with the recommendation that this is also be a remote file share.

If you want to restrict access to this location, you can provide the access credentials with the **LogLocationCredential** parameter. If you provide the credentials to access the log location, you should also provide the **CredentialEncryptionCertificate** parameter, which is used to encrypt the credentials stored locally on the Network Controller nodes.

With the default settings, it is recommended that you have at least 75 GB of free space in the central location, and 25 GB on the local nodes (if not using a central location) for a 3-node Network Controller cluster.

Change logging settings

You can change logging settings at any time using the `Set-NetworkControllerDiagnostic` cmdlet. The following settings can be changed:

- **Centralized log location.** You can change the location to store all the logs, with the `DiagnosticLogLocation` parameter.
- **Credentials to access log location.** You can change the credentials to access the log location, with the `LogLocationCredential` parameter.
- **Move to local logging.** If you have provided centralized location to store logs, you can move back to logging locally on the Network Controller nodes with the `UseLocalLogLocation` parameter (not recommended due to large disk space requirements).
- **Logging scope.** By default, all logs are collected. You can change the scope to collect only Network Controller cluster logs.
- **Logging level.** The default logging level is Informational. You can change it to Error, Warning, or Verbose.
- **Log Aging time.** The logs are stored in a circular fashion. You will have 3 days of logging data by default, whether you use local logging or centralized logging. You can change this time limit with `LogTimeLimitInDays` parameter.
- **Log Aging size.** By default, you will have a maximum 75 GB of logging data if using centralized logging and 25 GB if using local logging. You can change this limit with the `LogSizeLimitInMBs` parameter.

Collecting Logs and Traces

VMM deployments use centralized logging for the Network Controller by default. The file share location for these logs is specified when deploying the Network Controller service template.

If a file location has not been specified, local logging will be used on each Network Controller node with logs saved under `C:\Windows\tracing\SDNDiagnostics`. These logs are saved using the following hierarchy:

- CrashDumps
- NCApplicationCrashDumps

- NCAplicationLogs
- PerfCounters
- SDNDiagnostics
- Traces

The Network Controller uses (Azure) Service Fabric. Service Fabric logs may be required when troubleshooting certain issues. These logs can be found on each Network Controller node at C:\ProgramData\Microsoft\ServiceFabric.

If a user has run the *Debug-NetworkController* cmdlet, additional logs will be available on each Hyper-V host which has been specified with a server resource in the Network Controller. These logs (and traces if enabled) are kept under C:\NCDiagnostics

SLB Diagnostics

SLBM Fabric errors (Hosting service provider actions)

1. Check that Software Load Balancer Manager (SLBM) is functioning and that the orchestration layers can talk to each other: SLBM -> SLB Mux and SLBM -> SLB Host Agents. Run [DumpSlbRestState](#) from any node with access to Network Controller REST Endpoint.
2. Validate the *SDNSLBMPerfCounters* in PerfMon on one of the Network Controller node VMs (preferably the primary Network Controller node - Get-NetworkControllerReplica):
 - a. Is Load Balancer (LB) engine connected to SLBM? (*SLBM LBEngine Configurations Total > 0*)
 - b. Does SLBM at least know about its own endpoints? (*VIP Endpoints Total >= 2*)
 - c. Are Hyper-V (DIP) hosts connected to SLBM? (*HP clients connected == num servers*)
 - d. Is SLBM connected to Muxes? (*Muxes Connected == Muxes Healthy on SLBM == Muxes reporting healthy = # SLB Muxes VMs*).
3. Ensure the BGP router configured is successfully peering with the SLB MUX
 - a. If using RRAS with Remote Access (i.e. BGP virtual machine):
 - a. Get-BgpPeer should show connected
 - b. Get-BgpRouteInformation should show at least a route for the SLBM self VIP
 - b. If using physical Top-of-Rack (ToR) switch as BGP Peer, consult your documentation
 - a. For example: # show bgp instance
4. Validate the *SlbMuxPerfCounters* and *SLBMUX* counters in PerfMon on the SLB Mux VM
5. Check configuration state and VIP ranges in Software Load Balancer Manager Resource
 - a. Get-NetworkControllerLoadBalancerConfiguration -ConnectionUri <<https://> convertto-json -depth 8
(check VIP ranges in IP Pools and ensure SLBM self-VIP (*LoadBalancerManagerIPAddress*) and any tenant-facing VIPs are within these ranges)
 1. Get-NetworkControllerIpPool -NetworkId "<Public/Private VIP Logical Network Resource ID>" -SubnetId "<Public/Private VIP Logical Subnet Resource ID>" -ResourceId "" -ConnectionUri \$uri |convertto-json -depth 8
 - b. Debug-NetworkControllerConfigurationState -

If any of the checks above fail, the tenant SLB state will also be in a failure mode.

Remediation

Based on the following diagnostic information presented, fix the following:

- Ensure SLB Multiplexers are connected
 - Fix certificate issues
 - Fix network connectivity issues
- Ensure BGP peering information is successfully configured
- Ensure Host ID in the registry matches Server Instance ID in Server Resource (reference Appendix for *HostNotConnected* error code)

- Collect logs

SLBM Tenant errors (Hosting service provider and tenant actions)

1. [Hoster] Check *Debug-NetworkControllerConfigurationState* to see if any LoadBalancer resources are in an error state. Try to mitigate by following the Action items Table in the Appendix.
 - a. Check that a VIP endpoint is present and advertising routes
 - b. Check how many DIP endpoints have been discovered for the VIP endpoint
2. [Tenant] Validate Load Balancer Resources are correctly specified
 - a. Validate DIP endpoints which are registered in SLBM are hosting tenant virtual machines which correspond to the LoadBalancer Back-end Address pool IP configurations
3. [Hoster] If DIP endpoints are not discovered or connected:
 - a. Check *Debug-NetworkControllerConfigurationState*
 - a. Validate that NC and SLB Host Agent is successfully connected to the Network Controller Event Coordinator using `netstat -anp tcp |findstr 6640`
 - b. Check *HostId* in *nhostagent* service regkey (reference *HostNotConnected* error code in the Appendix) matches the corresponding server resource's instance Id (`Get-NCServer |convertto-json -depth 8`)
 - c. Check port profile id for virtual machine port matches corresponding virtual machine NIC resource's Instance Id
4. [Hosting provider] Collect logs

SLB Mux Tracing

Information from the Software Load Balancer Muxes can also be determined through Event Viewer.

1. Click on "Show Analytic and Debug Logs" under the Event Viewer View menu
2. Navigate to "Applications and Services Logs" > Microsoft > Windows > SLbMuxDriver > Trace in Event Viewer
3. Right click on it and select "Enable Log"

NOTE

It is recommended that you only have this logging enabled for a short time while you are trying to reproduce a problem

VFP and vSwitch Tracing

From any Hyper-V host which is hosting a guest VM attached to a tenant virtual network, you can collect a VFP trace to determine where problems might lie.

```
netsh trace start provider=Microsoft-Windows-Hyper-V-VfpExt overwrite=yes tracefile=vfp.etl report=disable
provider=Microsoft-Windows-Hyper-V-VmSwitch
netsh trace stop
netsh trace convert .\vfp.etl ov=yes
```

System Center Technologies for SDN

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

System Center includes the following technologies for use with Software Defined Networking (SDN):

- [System Center Operations Manager](#)
- [System Center Virtual Machine Manager](#)

System Center Operations Manager

System Center 2016 Operations Manager provides infrastructure monitoring that is flexible and cost-effective, helps ensure the predictable performance and availability of vital applications, and offers comprehensive monitoring for your datacenter and cloud, both private and public.

For more information, see [Operations Manager](#).

System Center Virtual Machine Manager

With System Center 2016 Virtual Machine Manager (VMM), you can:

- Provision and manage virtual networks at scale.
- Deploy and manage the SDN infrastructure, including network controllers, software load balancers, and gateways.
- Define and control virtual network policies centrally and link them to your applications or workloads.

When your workload is deployed or moved, the network configuration adjusts itself automatically. This is important because it removes the need for manual reconfiguration of network hardware, thereby reducing operational complexity while saving your valuable resources for higher-impact work.

- Helps you to control traffic flow between virtual networks, including the ability to define guaranteed bandwidth for your critical applications and workloads.

For more information, see [Set up a Software Defined Network \(SDN\) infrastructure in the VMM fabric](#).

Microsoft Azure and Software Defined Networking

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

Microsoft Azure is Microsoft's cloud platform: a growing collection of integrated services - compute, storage, data, networking, and app - that help you move faster, do more, and save money.

Microsoft's approach to software defined networking includes designing, building, and operating global-scale datacenter networks for services like Microsoft Azure. Microsoft Azure global datacenters perform tens of thousands of network changes every day, which is possible only because of software defined networking.

Microsoft Azure runs on the same Windows Server and Hyper-V platform that are included in Windows Server. Windows Server and System Center include improvements and best practices from Microsoft's experience in operating global scale datacenter networks like Microsoft Azure to you so that you can deploy the same technologies for flexibility, automation, and control when using software designed networking technologies.

For more information, see [What is Microsoft Azure?](#).

Contact the Datacenter and Cloud Networking Team

9/1/2018 • 2 minutes to read • [Edit Online](#)

Applies To: Windows Server 2016

Microsoft's **Software Defined Networking (SDN)** and **Container Networking** solutions are created by the Datacenter and Cloud Networking Team. Please use this page to be in touch with the team--to ask questions, provide feedback, report bugs or make feature requests.

There are many avenues for contacting Microsoft teams, and while we do our best on the SDN team to follow all of the avenues used by our community, here's a list of forums that tend to be the most active. *These are key resources for our users, and as such they are the avenues that we watch the closest.*

Twitter

Recently, we launched our presence on Twitter as [@Microsoft_SDN](#). Feel free to use our Twitter handle to ask questions, provide feedback or make feature/documentation requests.

In addition to a place where you can reach out with questions/feedback/requests, consider Twitter the place to get your "feed" on everything SDN and Windows container networking related -- Twitter is the first place we'll **post news**, announce **new features** and point the community to all of our **latest blogs and resources**.

GitHub ([Microsoft/SDN repo](#))

Go [here](#) to submit issues to the SDN team via our GitHub repository. This is the best place to **get help troubleshooting** or **report bugs**.

GitHub is the best place to contact us about topics that involve more detail than the kinds of things you could easily fit in a Tweet. *Need help with your SDN deployment? Unsure about how our features could suit your organization's unique needs? Being held up by a potential bug?* All good reasons to get in touch with us by submitting a GitHub issue.

Microsoft Docs

Our [Container Networking documentation](#) can be found on [Microsoft Docs \(docs.microsoft.com\)](#), which has **built-in commenting functionality**. To leave or to reply to a comment on Microsoft Docs just sign in, scroll down to the bottom of the Microsoft Docs page that you wish to reference, then make and submit your comment there.

[Microsoft Docs](#) is Microsoft's new unified documentation site. While most of our team's [SDN documentation](#) remains on TechNet, our [Container Networking documentation](#) is now on Microsoft Docs.

In general, If you run into a topic on Microsoft Docs that sparks a question or leaves you wanting more, just leave a comment on that page to share your feedback with the Microsoft team that can help.

Container-Specific Forums

Feel free to use any avenue on this page to provide feedback related to containers and container networking. However, if you're looking for Microsoft's primary forums for the container community specifically, refer to the following:

- [User voice](#) - best for *feature requests*
- [Github \(Virtualization repo\)](#) - best for seeking *troubleshooting help* and *reporting bugs*

Not seeing the forum for you?

Whenever possible, we encourage use of our public forums so that the broader community can benefit from access to the questions and comments that come our way. However, we also recognize that there are situations where email is simply the preferred way to get in touch with us. If you're in one of those situations, please send us an email at sdn_feedback@microsoft.com and we'll be happy to hear from you.

Virtual Private Networking (VPN)

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

VPN documentation is now located in the **Remote** section of this library, under **Remote Access**. Go to [Virtual Private Networking \(VPN\)](#).

Windows Internet Name Service (WINS)

9/1/2018 • 2 minutes to read • [Edit Online](#)

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

Windows Internet Name Service (WINS) is a legacy computer name registration and resolution service that maps computer NetBIOS names to IP addresses.

If you do not already have WINS deployed on your network, do not deploy WINS - instead, deploy Domain Name System (DNS). DNS also provides computer name registration and resolution services, and includes many additional benefits over WINS, such as integration with Active Directory Domain Services.

For more information, see [Domain Name System \(DNS\)](#)

If you have already deployed WINS on your network, it is recommended that you deploy DNS and then decommission WINS.

Windows Time Service (W32Time)

9/21/2018 • 2 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows 10 or later

The Windows Time service (W32Time) synchronizes the date and time for all computers running in Active Directory Domain Services (AD DS). Time synchronization is critical for the proper operation of many Windows services and line-of-business (LOB) applications. The Windows Time service uses the Network Time Protocol (NTP) to synchronize computer clocks on the network. NTP ensures that an accurate clock value, or timestamp, can be assigned to network validation and resource access requests.

In the Windows Time Service (W32Time) topic, the following content is available:

- **Windows Server 2016 Accurate Time.** Time synchronization accuracy in Windows Server 2016 has been improved substantially, while maintaining full backwards NTP compatibility with older Windows versions. Under reasonable operating conditions you can maintain a 1 ms accuracy with respect to UTC or better for Windows Server 2016 and Windows 10 Anniversary Update domain members.
- **Support boundary for high-accuracy environments.** This article describes the support boundaries for the Windows Time service (W32Time) in environments that require highly accurate and stable system time.
- **Configuring Systems for high accuracy.** Time synchronization in Windows 10 and Windows Server 2016 has been substantially improved. Under reasonable operating conditions, systems can be configured to maintain 1ms (millisecond) accuracy or better (with respect to UTC).
- **Windows Time for Traceability.** Regulations in many sectors require systems to be traceable to UTC. This means that a system's offset can be attested with respect to UTC. To enable regulatory compliance scenarios, Windows 10 and Server 2016 provides new event logs to provide a picture from the perspective of the Operating System to form an understanding of the actions taken on the system clock. These event logs are generated continuously for Windows Time service and can be examined or archived for later analysis.
- **Windows Time service technical reference.** The W32Time service provides network clock synchronization for computers without the need for extensive configuration. The W32Time service is essential to the successful operation of Kerberos V5 authentication and, therefore, to AD DS-based authentication.
 - **How the Windows Time service works.** Although the Windows Time service is not an exact implementation of the Network Time Protocol (NTP), it uses the complex suite of algorithms that is defined in the NTP specifications to ensure that clocks on computers throughout a network are as accurate as possible.
 - **Windows Time service tools and settings.** Most domain member computers have a time client type of NT5DS, which means that they synchronize time from the domain hierarchy. The only typical exception to this is the domain controller that functions as the primary domain controller (PDC) emulator operations master of the forest root domain, which is usually configured to synchronize time with an external time source.

Related Topics

For more information about the domain hierarchy and scoring system, see the "[What is Windows Time Service?](#)" blog post.

The windows time provider plugin model is [documented on TechNet](#).

An addendum referenced by the Windows 2016 Accurate Time article can be downloaded [here](#).

For a quick overview of Windows Time service, take a look at this [high-level overview video](#).

Insider preview

9/21/2018 • 2 minutes to read • [Edit Online](#)

Leap second support

Applies to: Windows Server 2019 and Windows 10, version 1809

A leap second is an occasional 1-second adjustment to UTC. As the earth's rotation slows, [UTC](#) (an atomic timescale) diverges from [mean solar time](#) or astronomical time. Once UTC has diverged by at most .9 seconds, a [leap second](#) is inserted to keep UTC in-sync with mean solar time.

Leap seconds have become important to meet the accuracy and traceability regulatory requirements both in the United States and the European Union.

For more information, see:

- Our [announcement blog](#)
- Validation Guide for the [Developers](#)
- Validation Guide for the [IT Pro](#)

Precision time protocol

Applies to: Windows Server 2019 and Windows 10, version 1809

A new time provider included in Windows Server 2019 and Windows 10 (version 1809) allows you to synchronize time using the Precision Time Protocol (PTP). As time distributes across a network, it encounters delay (latency), which if not accounted for, or if it is not symmetric, it becomes increasingly difficult to understand the time-stamp sent from the time server. PTP enables network devices to add the latency introduced by each network device into the timing measurements thereby providing a far more accurate time sample to the windows client.

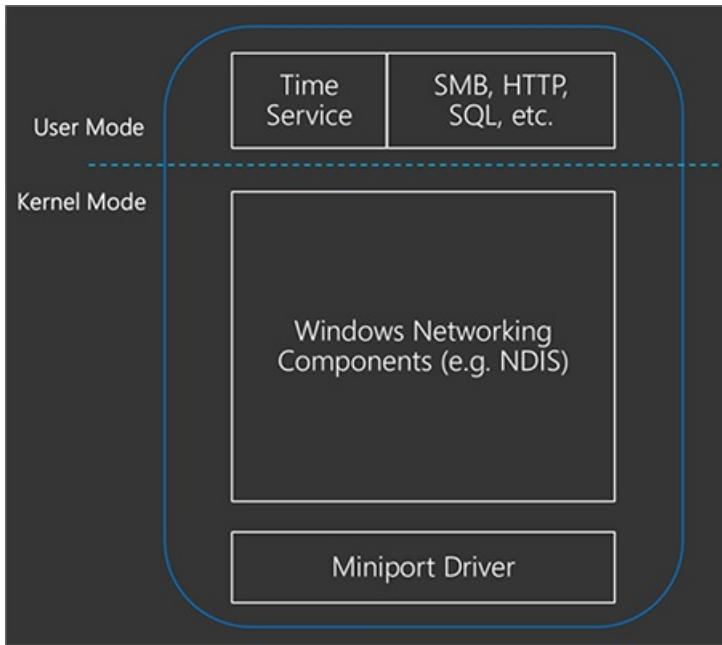
For more information, see:

- Our [announcement blog](#)
- Validation Guide for the [IT Pro](#)

Software timestamping

Applies to: Windows Server 2019 and Windows 10, version 1809

When receiving a timing packet over the network from a time server, it must be processed by the operating system's networking stack before being consumed in the time service. Each component in the networking stack introduces a variable amount of latency that affects the accuracy of the timing measurement.



To address this problem, software timestamping allows us to timestamp packets before and after the "Windows Networking Components" shown above to account for the delay in the operating system.

For more information, see:

- Our [announcement blog](#)
 - Validation Guide for the [IT Pro](#)
-

Accurate Time for Windows Server 2016

9/1/2018 • 37 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows 10 or later

The Windows Time service is a component that uses a plug-in model for client and server time synchronization providers. There are two built-in client providers on Windows, and there are third-party plug-ins available. One provider uses [NTP \(RFC 1305\)](#) or [MS-NTP](#) to synchronize the local system time to an NTP and/or MS-NTP compliant reference server. The other provider is for Hyper-V and synchronizes virtual machines (VM) to the Hyper-V host. When multiple providers exist, Windows will pick the best provider using stratum level first, followed by root delay, root dispersion, and finally time offset.

NOTE

For a quick overview of Windows Time service, take a look at this [high-level overview video](#).

In this topic, we discuss ... these topics as they relate to enabling accurate time:

- Improvements
- Measurements
- Best Practices

IMPORTANT

An addendum referenced by the Windows 2016 Accurate Time article can be downloaded [here](#). This document provides more details about our testing and measurement methodologies.

NOTE

The windows time provider plugin model is [documented on TechNet](#).

Domain Hierarchy

Domain and Standalone configurations work differently.

- Domain members use a secure NTP protocol, which uses authentication to ensure the security and authenticity of the time reference. Domain members synchronize with a master clock determined by the domain hierarchy and a scoring system. In a domain, there is a hierarchical layer of time strata, whereby each DC points to a parent DC with a more accurate time stratum. The hierarchy resolves to the PDC or a DC in the root forest, or a DC with the GTIMESERV domain flag, which denotes a Good Time Server for the domain. See the [Specify a Local Reliable Time Service Using GTIMESERV](#) section below.
- Standalone machines are configured to use time.windows.com by default. This name is resolved by your ISP, which should point to a Microsoft owned resource. Like all remotely located time references, network outages, may prevent synchronization. Network traffic loads and asymmetrical network paths may reduce the accuracy of the time synchronization. For 1 ms accuracy, you can't depend on a remote time sources.

Since Hyper-V guests will have at least two Windows Time providers to choose from, the host time and NTP, you might see different behaviors with either Domain or Standalone when running as a guest.

NOTE

For more information about the domain hierarchy and scoring system, see the "[What is Windows Time Service?](#)" blog post.

NOTE

Stratum is a concept used in both the NTP and Hyper-V providers, and its value indicates the clock's location in the hierarchy. Stratum 1 is reserved for the highest-level clock, and stratum 0 is reserved for the hardware assumed to be accurate and has little or no delay associated with it. Stratum 2 talk to stratum 1 servers, stratum 3 to stratum 2 and so on. While a lower stratum often indicates a more accurate clock, it is possible to find discrepancies. Also, W32time only accepts time from stratum 15 or below. To see the stratum of a client, use `w32tm /query /status`.

Critical Factors for Accurate Time

In every case for accurate time, there are three critical factors:

1. **Solid Source Clock** - The source clock in your domain needs to be stable and accurate. This usually means installing a GPS device or pointing to a Stratum 1 source, taking #3 into account. The analogy goes, if you have two boats on the water, and you are trying to measure the altitude of one compared to the other, your accuracy is best if the source boat is very stable and not moving. The same goes for time, and if your source clock isn't stable, then the entire chain of synchronized clocks is affected and magnified at each stage. It also must be accessible because disruptions in the connection will interfere with time synchronization. And finally, it must be secure. If the time reference is not properly maintained, or operated by a potentially malicious party, you could expose your domain to time based attacks.
2. **Stable client clock** - A stable client clock assures that the natural drift of the oscillator is containable. NTP uses multiple samples from potentially multiple NTP servers to condition and discipline your local computer's clock. It does not step the time changes, but rather slows or speeds up the local clock until you approach the accurate time quickly and stay accurate between NTP requests. However, if the client computer clock's oscillator is not stable, then more fluctuations in between adjustments can occur and the algorithms Windows uses to condition the clock don't work accurately. In some cases, firmware updates might be needed for accurate time.
3. **Symmetrical NTP communication** - It is critical that the connection for NTP communication is symmetrical. NTP uses calculations to adjust the time that assume the network path is symmetrical. If the path the NTP packet takes going to the server takes a different amount of time to return, the accuracy is affected. For example, the path could change due to changes in network topology, or packets being routed through devices that have different interface speeds.

For battery powered devices, both mobile and portable, you must consider different strategies. As per our recommendation, keeping accurate time requires the clock to be disciplined once a second, which correlates to the Clock Update Frequency. These settings will consume more battery power than expected and can interfere with power saving modes available in Windows for such devices. Battery powered devices also have certain power modes which stop all applications from running, which interferes with W32time's ability to discipline the clock and maintain accurate time. Additionally, clocks in mobile devices may not be very accurate to begin with. Ambient environmental conditions affect clock accuracy and a mobile device can move from one ambient condition to the next which may interfere with its ability to keep time accurately. Therefore, Microsoft does not recommend that you set up battery powered portable devices with high accuracy settings.

Why is time important?

There are many different reasons you might need accurate time. The typical case for Windows is Kerberos, which requires 5 minutes of accuracy between the client and server. However, there are many other areas that can be

affected by time accuracy including:

- Government Regulations like:
 - 50 ms accuracy for FINRA in the US
 - 1 ms ESMA (MiFID II) in the EU.
- Cryptography Algorithms
- Distributed systems like Cluster/SQL/Exchange and Document DBs
- Blockchain framework for bitcoin transactions
- Distributed Logs and Threat Analysis
- AD Replication
- PCI (Payment Card Industry), currently 1 second accuracy

Windows Server 2016 Improvements

Windows Time Service and NTP

Windows Server 2016 has improved the algorithms it uses to correct time and condition the local clock to synchronize with UTC. NTP uses 4 values to calculate the time offset, based on the timestamps of the client request/response and server request/response. However, networks are noisy, and there can be spikes in the data from NTP due to network congestion and other factors that affect network latency. Windows 2016 algorithms average out this noise using a number of different techniques which results in a stable and accurate clock. Additionally, the source we use for accurate time references an improved API which gives us better resolution. With these improvements we are able to achieve 1 ms accuracy with regards to UTC across a domain.

Hyper-V

Windows 2016 has improved the Hyper-V TimeSync service. Improvements include more accurate initial time on VM start or VM restore and interrupt latency correction for samples provided to w32time. This improvement allows us to stay within 10 μ s of the host with an RMS, (Root Mean Squared, which indicates variance), of 50 μ s, even on a machine with 75% load. For more information, see [Hyper-V architecture](#).

NOTE

Load was created using prime95 benchmark using balanced profile.

Additionally, the stratum level that the Host reports to the guest is more transparent. Previously the Host would present a fixed stratum of 2, regardless of its accuracy. With the changes in Windows Server 2016, the host reports a stratum one greater than the host stratum, which results in better time for virtual guests. The host stratum is determined by w32time through normal means based on its source time. Domain joined Windows 2016 guests will find the most accurate clock, rather than defaulting to the host. It was for this reason that we advised to manually disable Hyper-V Time Provider setting for machines participating in a domain in Windows 2012R2 and below.

Monitoring

Performance monitor counters have been added. These allow you to baseline, monitor, and troubleshoot time accuracy. These counters include:

| COUNTER | DESCRIPTION |
|---------|-------------|
|---------|-------------|

| COUNTER | DESCRIPTION |
|-------------------------------|--|
| Computed Time Offset | The absolute time offset between the system clock and the chosen time source, as computed by W32Time Service in microseconds. When a new valid sample is available, the computed time is updated with the time offset indicated by the sample. This is the actual time offset of the local clock. W32time initiates clock correction using this offset and updates the computed time in between samples with the remaining time offset that needs to be applied to the local clock. Clock accuracy can be tracked using this performance counter with a low polling interval (eg:256 seconds or less) and looking for the counter value to be smaller than the desired clock accuracy limit. |
| Clock Frequency Adjustment | The absolute clock frequency adjustment made to the local system clock by W32Time in parts per billion. This counter helps visualize the actions being taken by W32time. |
| NTP Roundtrip Delay | Most recent round-trip delay experienced by the NTP Client in receiving a response from the server in microseconds. This is the time elapsed on the NTP client between transmitting a request to the NTP server and receiving a valid response from the server. This counter helps characterize the delays experienced by the NTP client. Larger or varying roundtrips can add noise to NTP time computations, which in turn may affect the accuracy of time synchronization through NTP. |
| NTP Client Source Count | Active number of NTP Time sources being used by the NTP Client. This is a count of active, distinct IP addresses of time servers that are responding to this client's requests. This number may be larger or smaller than the configured peers, depending on DNS resolution of peer names and current reach-ability. |
| NTP Server Incoming Requests | Number of requests received by the NTP Server (Requests/Sec). |
| NTP Server Outgoing Responses | Number of requests answered by NTP Server (Responses/Sec). |

The first 3 counters target scenarios for troubleshooting accuracy issues. The Troubleshooting Time Accuracy and NTP section below, under [Best Practices](#), has more detail. The last 3 counters cover NTP server scenarios and are helpful when determine the load and baselining your current performance.

Configuration Updates per Environment

The following describes the changes in default configuration between Windows 2016 and previous versions for each Role. The settings for Windows Server 2016 and Windows 10 Anniversary Update (build 14393), are now unique which is why there are shown as separate columns.

| ROLE | SETTING | WINDOWS SERVER 2016 | WINDOWS 10 | WINDOWS SERVER 2012 R2
WINDOWS SERVER 2008 R2
WINDOWS 10 |
|-------------------------------|--------------------|---------------------|------------|--|
| Standalone/Nano Server | | | | |
| | <i>Time Server</i> | time.windows.com | NA | time.windows.com |

| ROLE | SETTING | WINDOWS SERVER 2016 | WINDOWS 10 | WINDOWS SERVER 2012 R2
WINDOWS SERVER 2008 R2
WINDOWS 10 |
|-----------------------------|-------------------------------|----------------------------|----------------------|---|
| | <i>Polling Frequency</i> | 64 - 1024 seconds | NA | Once a week |
| | <i>Clock Update Frequency</i> | Once a second | NA | Once a hour |
| Standalone Client | | | | |
| | <i>Time Server</i> | NA | time.windows.com | time.windows.com |
| | <i>Polling Frequency</i> | NA | Once a day | Once a week |
| | <i>Clock Update Frequency</i> | NA | Once a day | Once a week |
| Domain Controller | | | | |
| | <i>Time Server</i> | PDC/GTIMESERV | NA | PDC/GTIMESERV |
| | <i>Polling Frequency</i> | 64 -1024 seconds | NA | 1024 - 32768 seconds |
| | <i>Clock Update Frequency</i> | Once a day | NA | Once a week |
| Domain Member Server | | | | |
| | <i>Time Server</i> | DC | NA | DC |
| | <i>Polling Frequency</i> | 64 -1024 seconds | NA | 1024 - 32768 seconds |
| | <i>Clock Update Frequency</i> | Once a second | NA | Once every 5 minutes |
| Domain Member Client | | | | |
| | <i>Time Server</i> | NA | DC | DC |
| | <i>Polling Frequency</i> | NA | 1204 - 32768 seconds | 1024 - 32768 seconds |
| | <i>Clock Update Frequency</i> | NA | Once every 5 minutes | Once every 5 minutes |
| Hyper-V Guest | | | | |

| ROLE | SETTING | WINDOWS SERVER 2016 | WINDOWS 10 | WINDOWS SERVER 2012 R2
WINDOWS SERVER 2008 R2
WINDOWS 10 |
|------|-------------------------------|--|--|--|
| | <i>Time Server</i> | Chooses best option based on stratum of Host and Time server | Chooses best option based on stratum of Host and Time server | Defaults to Host |
| | <i>Polling Frequency</i> | Based on Role above | Based on Role above | Based on Role above |
| | <i>Clock Update Frequency</i> | Based on Role above | Based on Role above | Based on Role above |

NOTE

For Linux in Hyper-V, see the [Allowing Linux to use Hyper-V Host Time](#) section below.

Impact of increased polling and clock update frequency

In order to provide more accurate time, the defaults for polling frequencies and clock updates are increased which allow us to make small adjustments more frequently. This will cause more UDP/NTP traffic, however, these packets are small so there should be very little or no impact over broadband links. The benefit, however, is that time should be better on a wider variety of hardware and environments.

For battery backed devices, increasing the polling frequency can cause issues. Battery devices don't store the time while turned off. When they resume, it may require frequent corrections to the clock. Increasing the polling frequency will cause the clock to become unstable and could also use more power. Microsoft recommends you do not change the client default settings.

Domain Controllers should be minimally impacted even with the multiplied effect of the increased updates from NTP Clients in an AD Domain. NTP has a much smaller resource consumption as compared to other protocols and a marginal impact. You are more likely to reach limits for other domain functionality before being impacted by the increased settings for Windows Server 2016. Active Directory does use secure NTP, which tends to sync time less accurately than simple NTP, but we've verified it will scale up to clients two stratum away from the PDC.

As a conservative plan, you should reserve 100 NTP requests per second per core. For instance, a domain made up of 4 DCs with 4 cores each, you should be able to serve 1600 NTP requests per second. If you have 10k clients configured to sync time once every 64 seconds, and the requests are received uniformly over time, you would see 10,000/64 or around 160 requests/second, spread across all DCs. This falls easily within our 1600 NTP requests/sec based on this example. These are conservative planning recommendations and of course have a large dependency on your network, processor speeds and loads, so as always baseline and test in your environments.

It is also important to note that if your DCs are running with a considerable CPU load, greater than 40%, this will almost certainly add noise to NTP responses and affect your time accuracy in your domain. Again, you need to test in your environment to understand the actual results.

Time Accuracy Measurements

Methodology

To measure the time accuracy for Windows Server 2016, we used a variety of tools, methods and environments. You can use these techniques to measure and tune your environment and determine if the accuracy results meet your requirements.

Our domain source clock consisted of two high precision NTP servers with GPS hardware. We also used a separate reference test machine for measurements, which also had high precision GPS hardware installed from a different manufacturer. For some of the testing, you will need an accurate and reliable clock source to use as a reference in addition to your domain clock source.

We used four different methods to measure accuracy with both physical and virtual machines. Multiple methods provided independent means to validate the results.

1. Measure the local clock, that is conditioned by w32tm, against our reference test machine which has separate GPS hardware.
2. Measure NTP pings from the NTP server to clients using W32tm "stripchart"
3. Measure NTP pings from the client to the NTP server using W32tm "stripchart"
4. Measure Hyper-V results from the host to the guest using the Time Stamp Counter (TSC). This counter is shared between both partitions and the system time in both partitions. We calculated the difference of the host time and the client time in the virtual machine. Then we use the TSC clock to interpolate the host time from the guest, since the measurements don't happen at the same time. Also, we use the TSV clock factor out delays and latency in the API.

W32tm is built-in, but the other tools we used during our testing are available for the Microsoft repository on GitHub as open source for your testing and usage. The WIKI on the repository has more information describing how to use the tools to do measurements.

<https://github.com/Microsoft/Windows-Time-Calibration-Tools>

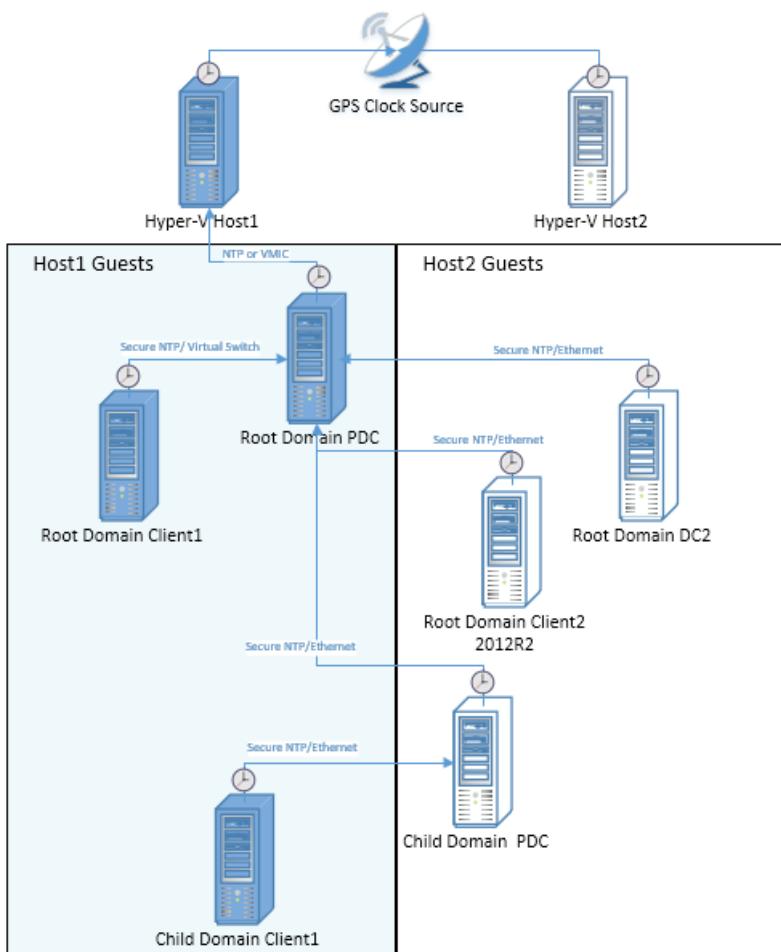
The test results shown below are a subset of measurements we made in one of the test environments. They illustrate the accuracy maintained at the start of the time hierarchy, and child domain client at the end of the time hierarchy. This is compared to the same machines in a 2012 based topology for comparison.

Topology

For comparison, we tested both a Windows Server 2012R2 and Windows Server 2016 based topology. Both topologies consist of two physical Hyper-V host machines that reference a Windows Server 2016 machine with GPS clock hardware installed. Each host runs 3 domain joined windows guests, which are arranged according to the following topology. The lines represent the time hierarchy, and the protocol/transport that is used.

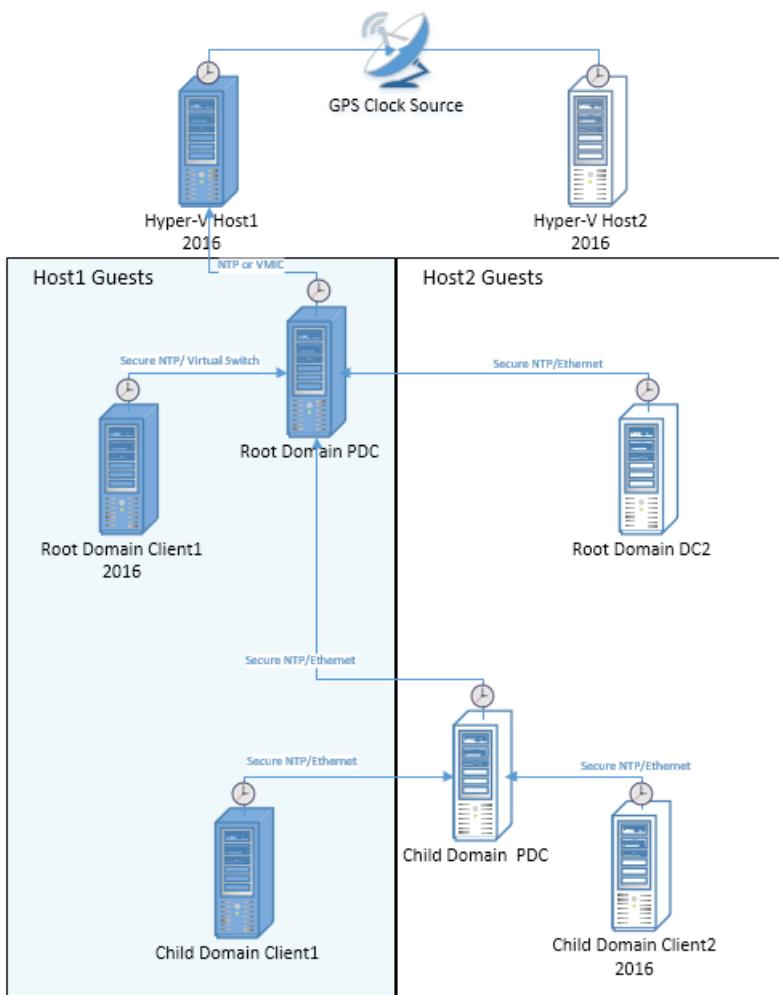
Windows Server 2016 Forest Time Hierarchy hosted between two 2016 Hyper-V Hosts

Servers are running Server2016 unless marked otherwise



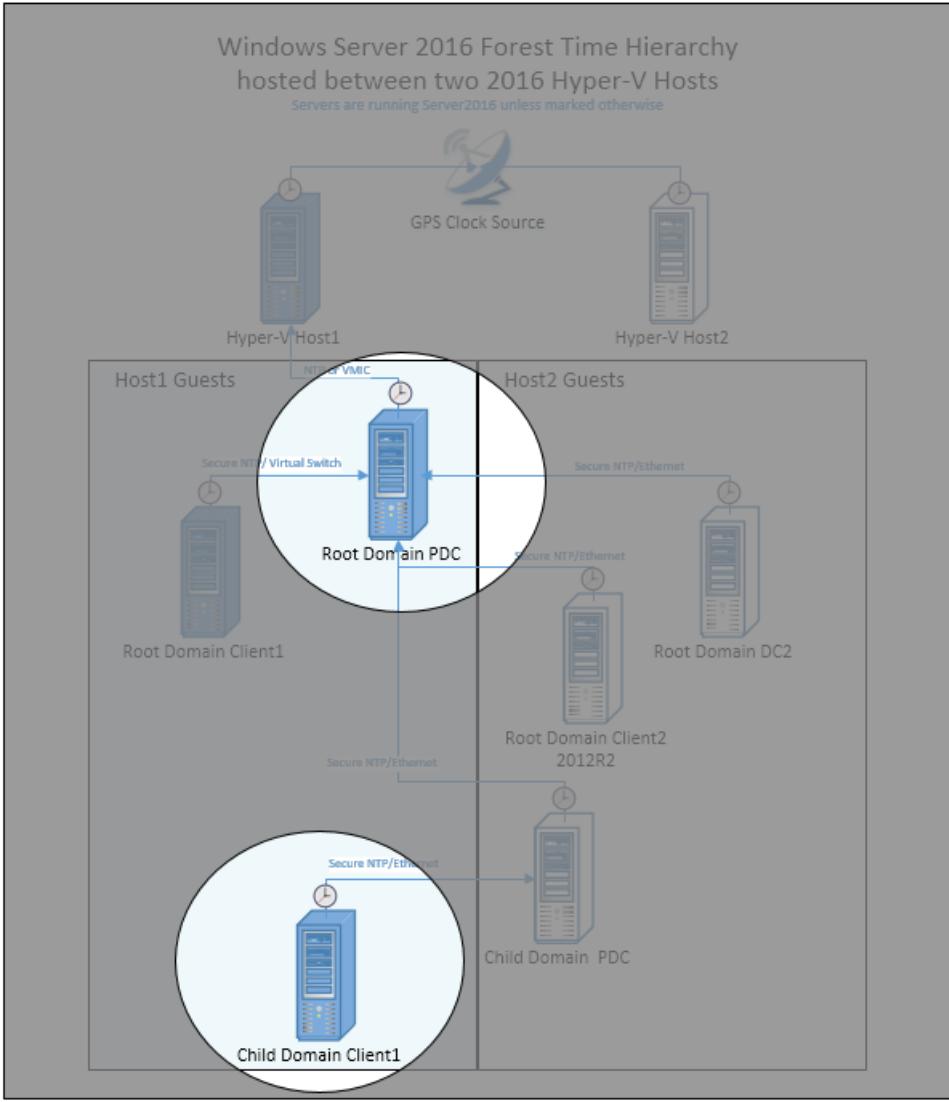
Windows Server 2012R2 Forest Time Hierarchy hosted between two 2016 Hyper-V Hosts

Servers are running Server2012R2 unless marked otherwise



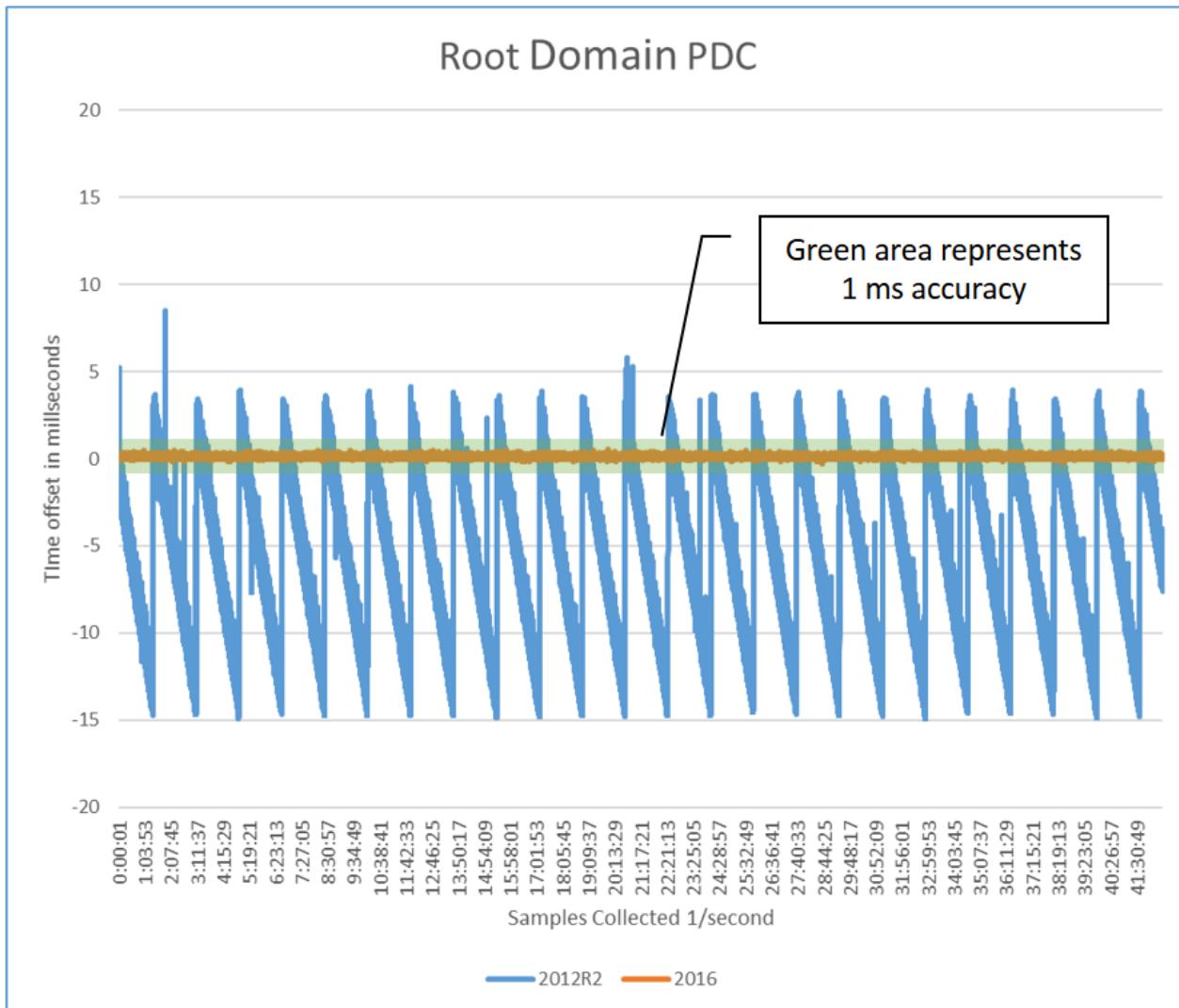
Graphical Results Overview

The following two graphs represent the time accuracy for two specific members in a domain based on the topology above. Each graph displays both the Windows Server 2012R2 and 2016 results overlaid, which demonstrates the improvements visually. The accuracy was measured from within the guest machine compared to the host. The graphical data represents a subset of the entire set of tests we've done and shows the best case and worst case scenarios.



Performance of the Root Domain PDC

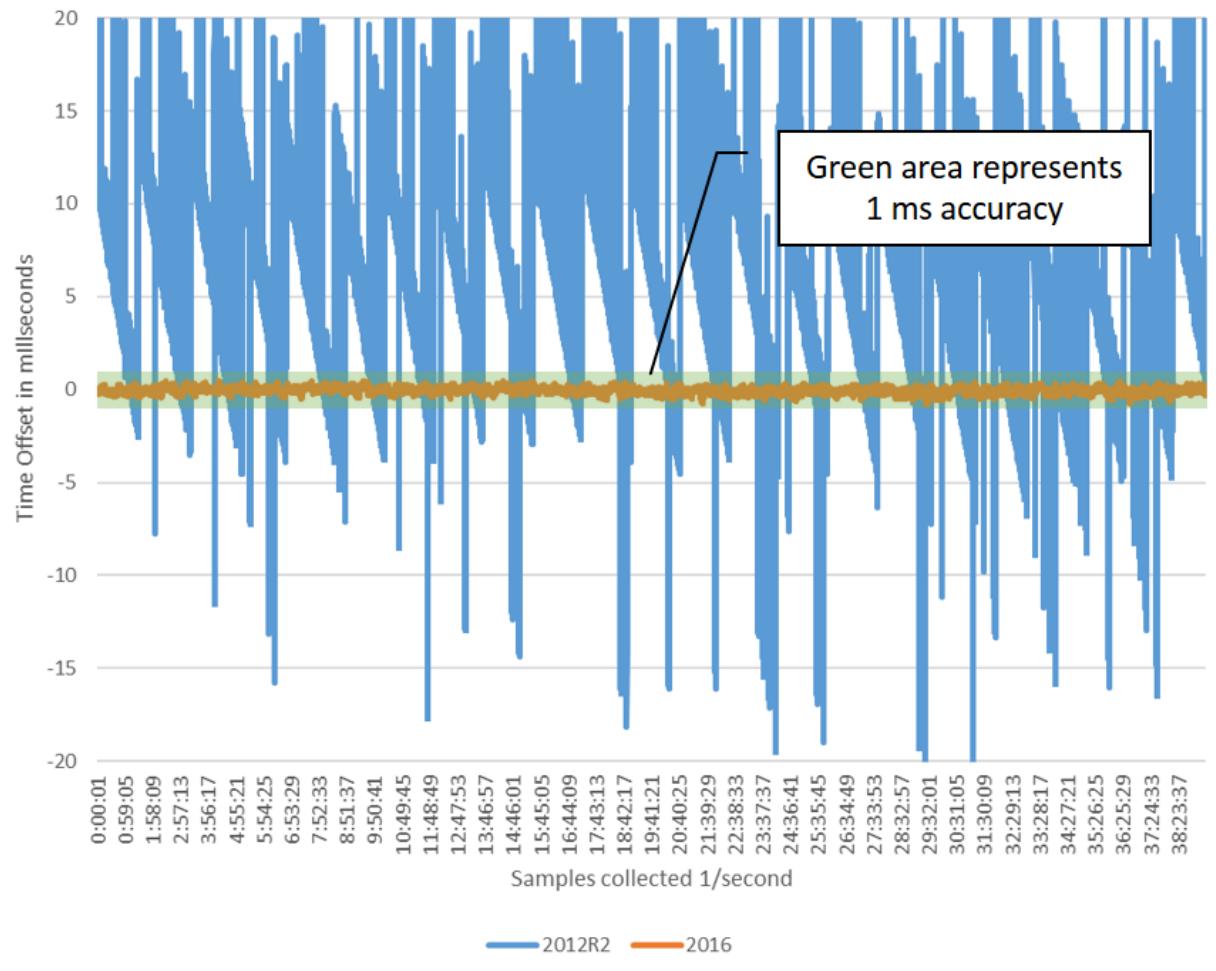
The Root PDC is synchronized to the Hyper-V host (using VMIC) which is a Windows Server 2016 with GPS hardware that is proven to be both accurate and stable. This is a critical requirement for 1 ms accuracy, which is shown as the green shaded area.



Performance of the Child Domain Client

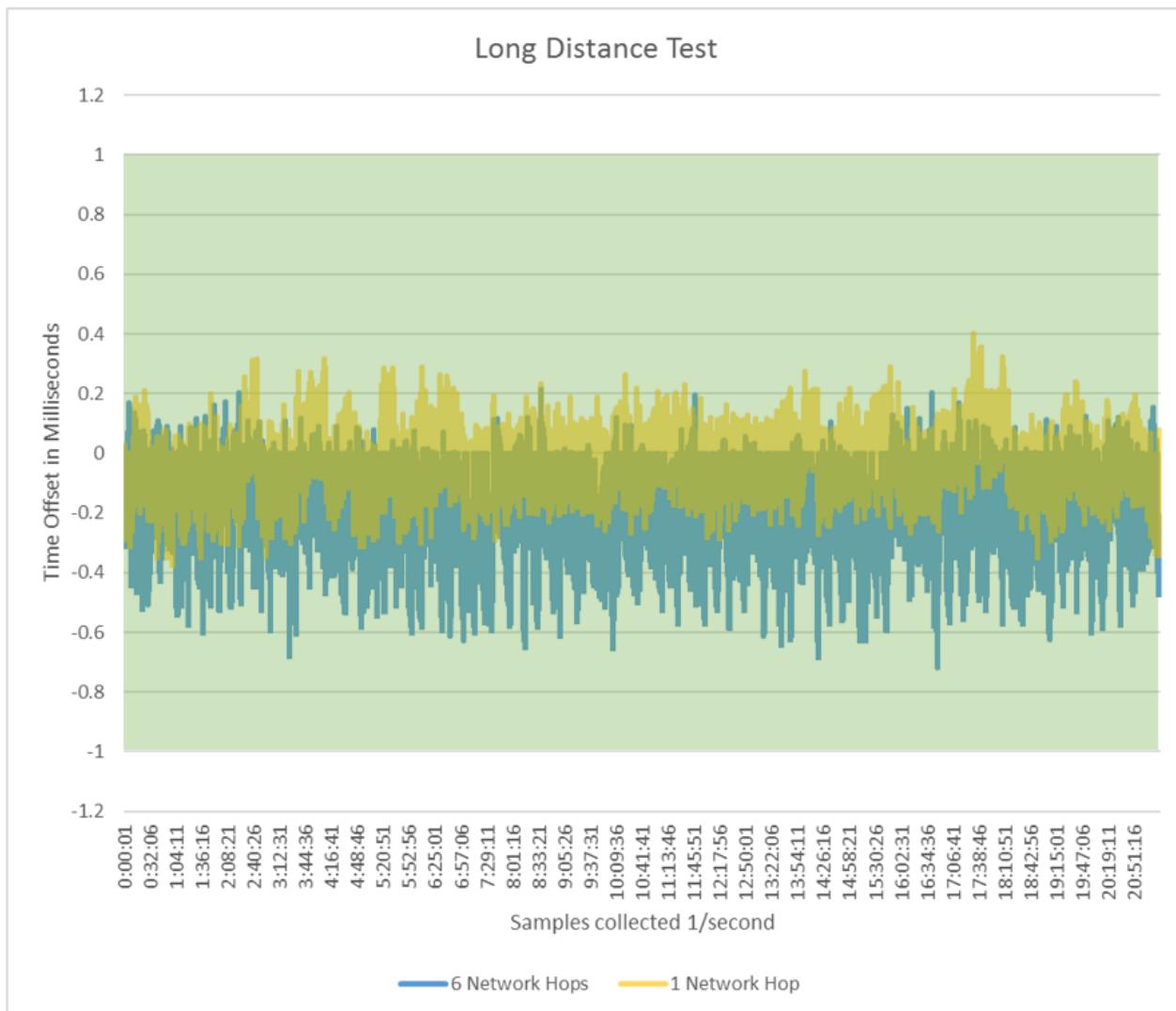
The Child Domain Client is attached to a Child Domain PDC which communicates to the Root PDC. Its time is also within the 1 ms requirement.

Child Domain Client



Long Distance Test

The following chart compares 1 virtual network hop to 6 physical network hops with Windows Server 2016. Two charts are overlaid on each other with transparency to show overlapping data. Increasing network hops mean higher latency, and larger time deviations. The chart is magnified and so the 1 ms bounds, represented by the green area, is larger. As you can see, the time is still within 1 ms with multiple hops. It's negatively shifted, which demonstrates a network asymmetry. Of course, every network is different, and measurements depend on a multitude of environmental factors.



Best Practices for accurate timekeeping

Solid Source Clock

A machine's time is only as good as the source clock it synchronizes with. In order to achieve 1 ms of accuracy, you'll need GPS hardware or a time appliance on your network you reference as the master source clock. Using the default of time.windows.com, may not provide a stable and local time source. Additionally, as you get further away from the source clock, the network affects the accuracy. Having a master source clock in each data center is required for the best accuracy.

Hardware GPS Options

There are various hardware solutions that can offer accurate time. In general, solutions today are based on GPS antennas. There are also radio and dial-up modem solutions using dedicated lines. They attach to your network as either an appliance, or plug into a PC, for instance Windows via a PCIe or USB device. Different options will deliver different levels of accuracy, and as always, results depend on your environment. Variables which affect accuracy include GPS availability, network stability and load, and PC Hardware. These are all important factors when choosing a source clock, which as we stated, is a requirement for stable and accurate time.

Domain and Synchronizing Time

Domain members use the domain hierarchy to determine which machine they use as a source to synchronize time. Each domain member will find another machine to sync with and save it as its clock source. Each type of domain member follows a different set of rules in order to find a clock source for time synchronization. The PDC in the Forest Root is the default clock source for all Domains. Listed below are different roles and high level description for how they find a source:

- **Domain Controller with PDC role** – This machine is the authoritative time source for a domain. It will have the most accurate time available in the domain, and must sync with a DC in the parent domain, except in cases where [GTIMESERV](#) role is enabled.
- **Any other Domain Controller** – This machine will act as a time source for clients and member servers in the domain. A DC can sync with the PDC of its own domain, or any DC in its parent domain.
- **Clients/Member Servers** – This machine can sync with any DC or PDC of its own domain, or a DC or PDC in the parent domain.

Based on the available candidates, a scoring system is used to find the best time source. This system takes into account the reliability of the time source and its relative location. This happens once when the time service is started. If you need to have finer control of how time synchronizes, you can add good time servers in specific locations or add redundancy. See the [Specify a Local Reliable Time Service Using GTIMESERV](#) section for more information.

Mixed OS Environments (Win2012R2 and Win2008R2)

While a pure Windows Server 2016 Domain environment is required for the best accuracy, there are still benefits in a mixed environment. Deploying Windows Server 2016 Hyper-V in a Windows 2012 domain will benefit the guests because of the improvements we mentioned above, but only if the guests are also Windows Server 2016. A Windows Server 2016 PDC, will be able to deliver more accurate time because of the improved algorithms it will be a more stable source. As replacing your PDC might not be an option, you can instead add a Windows Server 2016 DC with the [GTIMESERV](#) roll set which would be an upgrade in accuracy for your domain. A Windows Server 2016 DC can deliver better time to downstream time clients, however, it's only as good as its source NTP time.

Also as stated above, the clock polling and refresh frequencies have been modified with Windows Server 2016. These can be changed manually to your down-level DCs or applied via group policy. While we haven't tested these configurations, they should behave well in Win2008R2 and Win2012R2 and deliver some benefits.

Versions before Windows Server 2016 had multiple issues keeping accurate time which resulted in the system time drifting immediately after an adjustment was made. Because of this, obtaining time samples from an accurate NTP source frequently and conditioning the local clock with the data leads to smaller drift in their system clocks in the intra-sampling period, resulting in better time keeping on down-level OS versions. The best observed accuracy was approximately 5 ms when a Windows Server 2012R2 NTP Client, configured with the high-accuracy settings, synchronized its time from an accurate Windows 2016 NTP server.

In some scenarios involving guest domain controllers, Hyper-V TimeSync samples can disrupt domain time synchronization. This should no longer be an issue for Server 2016 guests running on Server 2016 Hyper-V hosts.

To disable the Hyper-V TimeSync service from providing samples to w32time, set the following guest registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\VMCTimeProvider
"Enabled"=dword:00000000
```

Allowing Linux to use Hyper-V Host Time

For Linux guests running in Hyper-V, clients are typically configured to use the NTP daemon for time synchronization against NTP servers. If the Linux distribution supports the TimeSync version 4 protocol and the Linux guest has the TimeSync integration service enabled, then it will synchronize against the host time. This could lead to inconsistent time keeping if both methods are enabled.

To synchronize exclusively against the host time, it is recommended to disable NTP time synchronization by either:

- Disabling any NTP servers in the ntp.conf file
- or Disabling the NTP daemon

In this configuration, the Time Server parameter is this host. Its Polling Frequency is 5 seconds and the Clock

Update Frequency is also 5 seconds.

To synchronize exclusively over NTP, it is recommended to disable the TimeSync integration service in the guest.

NOTE

Note: Support for accurate time with Linux guests requires a feature that is only supported in the latest upstream Linux kernels and it isn't something that's widely available across all Linux distros yet. Please reference [Supported Linux and FreeBSD virtual machines for Hyper-V on Windows](#) for more details about support distributions.

Specify a Local Reliable Time Service Using GTIMESERV

You can specify one or more domain controllers as accurate source clocks by using the GTIMESERV, Good Time Server, flags. For instance, specific domain controllers equipped with GPS hardware can be flagged as a GTIMESERV. This will insure your domain references a clock based on the GPS hardware.

NOTE

More information about domain flags can be found in the [MS-ADTS protocol documentation](#).

TIMESERV is another related Domain Services Flag which indicates whether a machine is currently authoritative, which can change if a DC loses connection. A DC in this state will return "Unknown Stratum" when queried via NTP. After trying multiple times, the DC will log System Event Time-Service Event 36.

If you want to configure a DC as a GTIMESERV, this can be configured manually using the following command. In this case the DC is using another machine(s) as the master clock. This could be an appliance or dedicated machine.

```
w32tm /config /manualpeerlist:"master_clock1,0x8 master_clock2,0x8" /syncfromflags:manual /reliable:yes  
/update
```

NOTE

For more information, see [Configure the Windows Time Service](#)

If the DC has the GPS hardware installed, you need to use these steps to disable the NTP client and enable the NTP server.

Start by disabling the NTP Client and enable the NTP Server using these registry key changes.

```
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\TimeProviders\NtpClient /v Enabled /t  
REG_DWORD /d 0 /f  
  
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\TimeProviders\NtpServer /v Enabled /t  
REG_DWORD /d 1 /f
```

Next, restart the Windows Time Service

```
net stop w32time && net start w32time
```

Finally, you indicate that this machine has a reliable time source using.

```
w32tm /config /reliable:yes /update
```

To check that the changes have been done properly, you can run the following commands which affect the results shown below.

```
w32tm /query /configuration
```

| VALUE | EXPECTED SETTING |
|---------------|---|
| AnnounceFlags | 5 (Local) |
| NtpServer | (Local) |
| DllName | C:\WINDOWS\SYSTEM32\w32time.DLL (Local) |
| Enabled | 1 (Local) |
| NtpClient | (Local) |

```
w32tm /query /status /verbose
```

| VALUE | EXPECTED SETTING |
|--------------|--|
| Stratum | 1 (primary reference - syncd by radio clock) |
| Referenceld | 0x4C4F434C (source name: "LOCAL") |
| Source | Local CMOS Clock |
| Phase Offset | 0.0000000s |
| Server Role | 576 (Reliable Time Service) |

Windows Server 2016 on 3rd Party Virtual Platforms

When Windows is virtualized, by default the Hypervisor is responsible for providing time. But domain joined members need to be synchronized with the Domain Controller in order for Active Directory to work properly. It is best to disable any time virtualization between the guest and the host of any 3rd party virtual platforms.

Discovering the Hierarchy

Since the chain of time hierarchy to the master clock source is dynamic in a domain, and negotiated, you will need to query the status of a particular machine to understand it's time source and chain to the master source clock. This can help diagnose time synchronization problems.

Given you want to troubleshoot a specific client; the first step is to understand its time source by using this w32tm command.

```
w32tm /query /status
```

The results display the Source among other things. The Source indicates with whom you synchronize time in the domain. This is the first step of this machines time hierarchy. Next use Source entry from above and use the /StripChart parameter to find the next time source in the chain.

```
w32tm /stripchart /computer:MySourceEntry /packetinfo /samples:1
```

Also useful, the following command lists each domain controller it can find in the specified domain and prints a result which lets you determine each partner. This command will include machines that have been configured manually.

```
w32tm /monitor /domain:my_domain
```

Using the list, you can trace the results through the domain and understand the hierarchy as well as the time offset at each step. By locating the point where the time offset gets significantly worse, you can pinpoint the root of the incorrect time. From there you can try to understand why that time is incorrect by turning on [w32tm logging](#).

Using Group Policy

You can use Group Policy to accomplish stricter accuracy by, for instance, assigning clients to use specific NTP servers or to control how down-level OS's are configured when virtualized.

Below is a list of possible scenarios and relevant Group Policy settings:

Virtualized Domains - In order to control Virtualized Domain Controllers in Windows 2012R2 so that they synchronize time with their domain, rather than with the Hyper-V host, you can disable this registry entry. For the PDC, you don't want to disable the entry as the Hyper-V host will deliver the most stable time source. The registry entry requires that you restart the w32time service after it is changed.

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\VMCTimeProvider]
"Enabled"=dword:00000000
```

Accuracy Sensitive Loads - For time accuracy sensitive workloads, you could configure groups of machines to set the NTP servers and any related time settings, such as polling and clock update frequency. This is normally handled by the domain, but for more control you could target specific machines to point directly to the master clock.

| GROUP POLICY SETTING | NEW VALUE |
|----------------------|------------------------------|
| NtpServer | ClockMasterName,0x8 |
| MinPollInterval | 6 – 64 seconds |
| MaxPollInterval | 6 |
| UpdateInterval | 100 – Once per second |
| EventLogFlags | 3 – All special time logging |

NOTE

The NtpServer and EventLogFlags settings are located under System\Windows Time Service\Time Providers using the Configure Windows NTP Client settings. The other 3 are located under System\Windows Time Service using the Global Configuration settings.

Remote Accuracy Sensitive Loads Remote – For systems in branch domains for instance Retail and the Payment Credit Industry (PCI), Windows uses the current site information and DC Locator to find a local DC, unless there is a manual NTP time source configured. This environment requires 1 second of accuracy, which uses

faster convergence to the correct time. This option allows the w32time service to move the clock backwards. If this is acceptable and meets your requirements, you can create the following policy. As with any environment, make sure to test and baseline your network.

| GROUP POLICY SETTING | NEW VALUE |
|-----------------------|--|
| MaxAllowedPhaseOffset | 1, if more than one second, set clock to correct time. |

The MaxAllowedPhaseOffset setting is located under System\Windows Time Service using the Global Configuration settings.

NOTE

For more information on group policy and related entries, see [Windows Time Service Tools](#) and Settings article on TechNet.

Azure and Windows IaaS considerations

Azure Virtual Machine: Active Directory Domain Services

If the Azure VM running Active Directory Domain Services is part of an existing on-premises Active Directory Forest, then TimeSync(VMIC), should be disabled. This is to allow all DCs in the Forest, both physical and virtual, to use a single time sync hierarchy. Refer to the best practice whitepaper "[Running Domain Controllers in Hyper-V](#)"

Azure Virtual Machine: Domain-joined machine

If you are hosting a machine which is domain joined to an existing Active Directory Forest, virtual or physical, the best practice is to disable TimeSync for the guest and ensure W32Time is configured to synchronize with its Domain Controller via configuring time for Type=NTP5

Azure Virtual Machine: Standalone workgroup machine

If the Azure VM is not joined to a domain, nor is it a Domain Controller, the recommendation is to keep the default time configuration and have the VM synchronize with the host.

Windows Application Requiring Accurate Time

Time Stamp API

Programs which require the greatest accuracy with regards to UTC, and not the passage of time, should use the [GetSystemTimePreciseAsFileTime API](#). This assures your application gets System Time, which is conditioned by the Windows Time service.

UDP Performance

If you have an application that uses UDP communication for transactions and it's important to minimize latency, there are some related registry entries you can use to configure a range of ports to be excluded from port the base filtering engine. This will improve both the latency and increase your throughput. However, changes to the registry should be limited to experienced administrators. Additionally, this work around excludes ports from being secured by the firewall. See the article reference below for more information.

For Windows Server 2012 and Windows Server 2008, you will need to install a Hotfix first. You can reference this KB article: [Datagram loss when you run a multicast receiver application in Windows 8 and in Windows Server 2012](#)

Update Network Drivers

Some network vendors have driver updates which improve performance with regards to driver latency and buffering UDP packets. Please contact your network vendor to see if there are updates to help with UDP

throughput.

Logging for Auditing Purposes

To comply with time tracing regulations you can manually archive w32tm logs, event logs and performance monitor information. Later, the archived information can be used to attest compliance at a specific time in the past. The following factors are used to indicate the accuracy.

1. Clock accuracy using the Computed Time Offset performance monitor counter. This shows the clock with in the desired accuracy.
2. Clock source looking for "Peer Response from" in the w32tm logs. Following the message text is the IP address or VMIC, which describes the time source and the next in chain of reference clocks to validate.
3. Clock condition status using the w32tm logs to validate that "ClockDispl Discipline: *SKEW*TIME*" are occurring. This indicates that w32tm is active at the time.

Event Logging

To get the complete story, you will also need Event log information. By collecting the System Event log, and filtering on Time-Server, Microsoft-Windows-Kernel-Boot, Microsoft-Windows-Kernel-General, you may be able to discover if there are other influences that have changed the time, for instance, third parties. These logs might be necessary to rule out external interference. Group policy can affect which event logs are written to the log. See the section above on Using Group Policy for more details.

W32time Debug Logging

To enable w32tm for auditing purposes, the following command enables logging that shows the periodic updates of the clock and indicates the source clock. Restart the service to enable the new logging.

For more information, see [How to turn on debug logging in the Windows Time Service](#).

```
w32tm /debug /enable /file:C:\Windows\Temp\w32time-test.log /size:10000000 /entries:0-73,103,107,110
```

Performance Monitor

The Windows Server 2016 Windows Time service exposes performance counters which can be used to collect logging for auditing. These can be logged locally or remotely. You can record the Computer Time Offset and Round Trip delay counters.

And like any performance counter, you can monitor them remotely and create alerts using System Center Operations Manager. You can, for instance, use an alert to alarm you when the Time Offset drifts from the desired accuracy. The [System Center Management Pack](#) has more information.

Windows Traceability Example

From w32tm log files you will want to validate two pieces of information. The first is an indication that the log file is currently condition clock. This prove that your clock was being conditioned by the Windows Time Service at the disputed time.

```
151802 20:18:32.9821765s - ClockDispln Discipline: *SKEW*TIME* - PhCRR:223 CR:156250 UI:100 phcT:65 KPh0:14307  
151802 20:18:33.9898460s - ClockDispln Discipline: *SKEW*TIME* - PhCRR:1 CR:156250 UI:100 phcT:64 KPh0:41  
151802 20:18:44.1090410s - ClockDispln Discipline: *SKEW*TIME* - PhCRR:1 CR:156250 UI:100 phcT:65 KPh0:38
```

The main point is that you see messages prefixed with ClockDispln Discipline which is proof w32time is interacting with your system clock.

Next you need to find the last report in the log before the disputed time which reports the source computer which is currently being used as the reference clock. This could be an IP address, computer name, or the VMIC provider, which indicates that it's syncing with the Host for Hyper-V. The following example provides an IPv4 address of

10.197.216.105.

```
151802 20:18:54.6531515s - Response from peer 10.197.216.105,0x8 (ntp.m|0x8|0.0.0.0:123->10.197.216.105:123),  
ofs: +00.0012218s
```

Now that you've validated the first system in the reference time chain, you need to investigate the log file on reference time source and repeat the same steps. This continues until you get to a physical clock, like GPS or a known time source like NIST. If the reference clock is GPS hardware, then logs from the manufacturer might also be required.

Network Considerations

The NTP protocol algorithms have a dependency on the Symmetry of your network. As you increase the number of network hops, the probability of asymmetry increases. Therefore, it's difficult to predict what types of accuracies you will see in your specific environments.

Performance Monitor and the new Windows Time counters in Windows Server 2016 can be used to assess your environments accuracy and create baselines. Additionally, you can perform troubleshooting to determine the current offset of any machine on your network.

There are two general standards for accurate time over the network. PTP ([Precision Time Protocol - IEEE 1588](#)) has tighter requirements on network infrastructure but can often provide sub-microsecond accuracy. NTP ([Network Time Protocol – RFC 1305](#)) works on a larger variety of networks and environments, which makes it easier to manage.

Windows supports Simple NTP (RFC2030) by default for non-domain joined machines. For Domain joined machines, we use a secure NTP called [MS-SNTP](#), which leverages domain negotiated secrets which provide a management advantage over Authenticated NTP described in RFC1305 and RFC5905.

Both the domain and non-domain joined protocols require UDP port 123. For more information about NTP best practices, refer to [Network Time Protocol Best Current Practices IETF Draft](#).

Reliable Hardware Clock (RTC)

Windows does not step time, unless certain bounds are exceeded, but rather disciplines the clock. That means w32tm adjusts the frequency of the clock at a regular interval, using the Clock Update Frequency setting, which defaults to once a second with Windows Server 2016. If the clock is behind, it accelerates the frequency and if it's ahead, it slows the frequency down. However, during that time between clock frequency adjustments, the hardware clock is in control. If there's an issue with the firmware or the hardware clock, the time on the machine can become less accurate.

This is another reason you need to test and baseline in your environment. If the "Computed Time Offset" performance counter does not stabilize at the accuracy you are targeting, then you might want to verify your firmware is up to date. As another test, you can see if duplicate hardware reproduce the same issue.

Troubleshooting Time Accuracy and NTP

You can use the Discovering the Hierarchy section above to understand the source of the inaccurate time. Looking at the time offset, find the point in the hierarchy where time diverges the most from its NTP Source. Once you understand the hierarchy, you'll want to try and understand why that particular time source doesn't receive accurate time.

Focusing on the system with divergent time, you can use these tools below to gather more information to help you determine the issue and to find a resolution. The UpstreamClockSource reference below, is the clock discovered using "w32tm /config /status".

- System Event logs

- Enable logging using: w32tm logs -w32tm /debug /enable /file:C:\Windows\Temp\w32time-test.log /size:10000000 /entries:0-300
- w32Time Registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time
- Local network traces
- Performance Counters (from the local machine or the UpstreamClockSource)
- W32tm /stripchart /computer:UpstreamClockSource
- PING UpstreamClockSource to understand latency and number of hops to Source
- Tracert UpstreamClockSource

| PROBLEM | SYMPTOMS | RESOLUTION |
|---|--|---|
| Local TSC clock is not stable. | Using Perfmon - Physical Computer – Sync clock stable clock, but you still see that every 1-2 minutes of several 100us. | Update Firmware or validate different hardware doesn't display the same issue. |
| Network Latency | w32tm stripchart displays a RoundTripDelay of more than 10 ms. Variation in the delay cause noise as large as ½ of the round trip time, for instance a delay that is only in one direction.
UpstreamClockSource is multiple hops, as indicated by PING. TTL should be close to 128.

Use Tracert to find the latency at each hop. | Find a closer clock source for time. One solution is to install a source clock on the same segment or manually point to source clock that is geographically closer. For a domain scenario, add a machine with the GTimeServ role. |
| Unable to reliably reach the NTP source | W32tm /stripchart intermittently returns "Request timed out" | NTP Source isn't responsive |
| NTP Source isn't responsive | Check Perfmon counters for NTP Client Source Count, NTP Server Incoming Requests, NTP Server Outgoing Responses and determine your usage as compared to your baselines. | Using server performance counters, determine if load has changed in reference to your baselines.
Are there network congestion issues? |
| Domain Controller not using the most accurate clock | Changes in the topology or recently added master time clock. | w32tm /resync /rediscover |
| Client Clocks are drifting | Time-Service event 36 in System event log and/or text in log file describing that: "NTP Client Time Source Count" counter going from 1 to 0 | Troubleshoot the upstream source and understand if it's running into performance issues. |

Baselining Time

Baselining is important so that you can first, understand the performance and accuracy of your network, and compare with the baseline in the future when problems occur. You'll want to baseline the root PDC or any machines marked with the GTIMERSRV. We would also recommend you baseline the PDC in every forest. Finally pick any critical DCs or machines that have interesting characteristics, like distance or high loads and baseline those.

It is also useful to baseline Windows Server 2016 vs 2012 R2, however you only have w32tm /stripchart as a tool you can use to compare, since Windows Server 2012R2 doesn't have performance counters. You should pick two machines with the same characteristics, or upgrade a machine and compare the results after the update. The

Windows Time Measurements addendum has more information on how to do detailed measurements between 2016 and 2012.

Using the all the w32time performance counters, collect data for at least a week. This will insure you have enough of a reference to account for various in the network over time and enough of a run to provide confidence that your time accuracy is stable.

NTP Server Redundancy

For manual NTP Server configuration used with non-domain joined machines or the PDC, having more than one server is a good redundancy measure in case of availability. It might also give better accuracy, assuming the all the sources are accurate and stable. However, if the topology is not well designed, or the time sources are not stable, the resulting accuracy could be worse so caution is advised. The limit of supported time servers w32time can manually reference is 10.

Leap Seconds

The earth's rotation period varies over time, caused by climatic and geological events. Typically, the variation is about a second every couple of years. Whenever the variation from atomic time grows too large, a correction of one second (up or down) is inserted, called a leap second. This is done in such a way that the difference never exceeds 0.9 seconds. This correction is announced six months ahead of the actual correction. Before Windows Server 2016, the Microsoft Time Service was not aware of leap seconds, but relied on the external time service to take care of this. With the increased time accuracy of Windows Server 2016, Microsoft is working on a more suitable solution for the leap second problem.

Secure Time Seeding

W32time in Server 2016 includes the Secure Time Seeding feature. This feature determines the approximate current time from outgoing SSL connections. This time value is used to monitor the local system clock and correct any gross errors. You can read more about the feature in [this blog post](#). In deployments with a reliable time source(s) and well monitored machines that include monitoring for time offsets, you may choose to not use the Secure Time Seeding feature and rely on your existing infrastructure instead.

You can disable the feature with these steps:

1. Set the UtilizeSSLTimeData registry configuration value to 0 on a specific machine:

```
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\Config /v UtilizeSslTimeData /t REG_DWORD /d 0 /f
```

2. If you are unable to reboot the machine immediately due to some reason, you can notify W32time service about the configuration update. This stops time monitoring and enforcement based on time data collected from SSL connections.

```
W32tm.exe /config /update
```

3. Rebooting the machine makes the setting effective immediately and also causes it to stop collecting any time data from SSL connections. The latter part has a very small overhead and should not be a perf concern.

4. To apply this setting in an entire domain, please set the UtilizeSSLTimeData value in W32time group policy setting to 0 and publish the setting. When the setting is picked up by a Group Policy Client, W32time service is notified and it will stop time monitoring and enforcement using SSL time data. The SSL time data collection will stop when each machine reboots. If your domain has portable slim laptops/tablets and other devices, you may want to exclude such machines from this policy change. These devices will eventually face battery drain and need the Secure Time Seeding feature to bootstrap their time.

Support boundary to configure the Windows Time service for high-accuracy environments

9/21/2018 • 3 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016, and Windows 10 version 1607 or later

This article describes the support boundaries for the Windows Time service (W32Time) in environments that require highly accurate and stable system time.

High Accuracy support for Windows 8.1 and 2012 R2 (or Prior)

Earlier versions of Windows (Prior to Windows 10 1607 or Windows Server 2016 1607) cannot guarantee highly accurate time. The Windows Time service on these systems:

- Provided the necessary time accuracy to satisfy Kerberos version 5 authentication requirements
- Provided loosely accurate time for Windows clients and servers joined to a common Active Directory forest

Tighter accuracy requirements were outside of the design specification of the Windows Time Service on these operating systems and is not supported.

Windows 10 and Windows Server 2016

Time accuracy in Windows 10 and Windows Server 2016 has been substantially improved, while maintaining full backwards NTP compatibility with older Windows versions. Under the right operating conditions, systems running Windows 10 or Windows Server 2016 and newer releases can deliver 1 second, 50ms (milliseconds), or 1ms accuracy.

IMPORTANT

Highly accurate time sources

The resulting time accuracy in your topology is highly dependent on using an accurate, stable root (stratum 1) time source. There are Windows based and non-Windows based highly accurate, Windows compatible, NTP Time source hardware sold by 3rd-party vendors. Please check with your vendor on the accuracy of their products.

IMPORTANT

Time accuracy

Time accuracy entails the end-to-end distribution of accurate time from a highly accurate authoritative time source to the end device. Anything that introduces network asymmetry will negatively influence accuracy, for example physical network devices or high CPU load on the target system.

High Accuracy Requirements

The rest of this document outlines the environmental requirements that must be satisfied to support the respective high accuracy targets.

Target Accuracy: 1 Second (1s)

To achieve 1s accuracy for a specific target machine when compared to a highly accurate time source:

- The target system must run Windows 10, Windows Server 2016.
- The target system must synchronize time from an NTP hierarchy of time servers, culminating in a highly accurate, Windows compatible NTP time source.
- All Windows operating systems in the NTP hierarchy mentioned above must be configured as documented in the [Configuring Systems for High Accuracy](#) documentation.
- The cumulative one-way network latency between the target and source must not exceed 100ms. The cumulative network delay is measured by adding the individual one-way delays between pairs of NTP client-server nodes in the hierarchy starting with the target and ending at the source. For more information, please review the high accuracy time sync document.

Target Accuracy: 50 Milliseconds

All requirements outlined in the section **Target Accuracy: 1 Second** apply, except where stricter controls are outlined in this section.

The additional requirements to achieve 50ms accuracy for a specific target system are:

- The target computer must have better than 5ms of network latency between its time source.
- The target system must be no further than stratum 5 from a highly accurate time source

NOTE

Run "w32tm /query /status" from the command line to see the stratum.

- The target system must be within 6 or less network hops from the highly accurate time source
- The one-day average CPU utilization on all strata must not exceed 90%
- For virtualized systems, the one-day average CPU utilization of the host must not exceed 90%

Target Accuracy: 1 Millisecond

All requirements outlined in the sections **Target Accuracy: 1 Second** and **Target Accuracy: 50 Milliseconds** apply, except where stricter controls are outlined in this section.

The additional requirements to achieve 1 ms accuracy for a specific target system are:

- The target computer must have better than 0.1 ms of network latency between its time source
- The target system must be no further than stratum 5 from a highly accurate time source

NOTE

Run 'w32tm /query /status' from the command line to see the stratum

- The target system must be within 4 or less network hops from the highly accurate time source
- The one-day average CPU utilization across each stratum must not exceed 80%
- For virtualized systems, the one-day average CPU utilization of the host must not exceed 80%

Configuring Systems for High Accuracy

9/11/2018 • 4 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016, and Windows 10 version 1607 or later

Time synchronization in Windows 10 and Windows Server 2016 has been substantially improved. Under reasonable operating conditions, systems can be configured to maintain 1ms (millisecond) accuracy or better (with respect to UTC).

The following guidance will help you configure your systems to achieve high accuracy. This article discusses the following requirements:

- Supported Operating Systems
- System configuration

WARNING

Prior Operating Systems Accuracy Goals

Windows Server 2012 R2 and below can not meet the same high accuracy objectives. These operating systems are not supported for high accuracy.

In these versions, the Windows Time service satisfied the following requirements:

- Provided the necessary time accuracy to satisfy Kerberos version 5 authentication requirements.
- Provided loosely accurate time for Windows clients and servers joined to a common Active Directory forest.

Greater tolerances on 2012 R2 and below are outside the design specification of the Windows Time service.

Windows 10 and Windows Server 2016 Default Configuration

While we support accuracy up to 1ms on Windows 10 or Windows Server 2016, the majority of customers do not require highly accurate time.

As such, the **default configuration** is intended to satisfy the same requirements as prior operating systems which are to:

- Provide the necessary time accuracy to satisfy Kerberos version 5 authentication requirements.
- Provide loosely accurate time for Windows clients and servers joined to a common Active Directory forest.

How to Configure Systems for High Accuracy

IMPORTANT

Note Regarding Supportability of Highly Accurate Systems

Time accuracy entails the end-to-end distribution of accurate time from the authoritative time source to the end device.

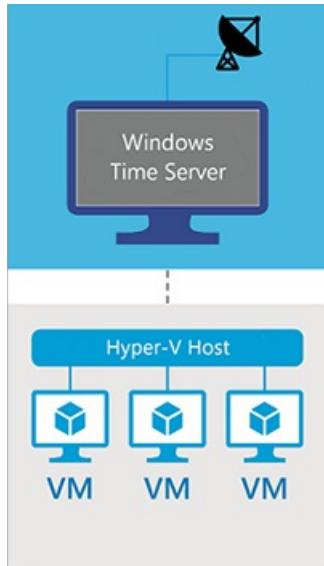
Anything that adds asymmetry in measurements along this path will negatively influence accuracy will affect the accuracy achievable on your devices.

For this reason, we have documented the [Support boundary to configure the Windows Time service for high-accuracy environments](#) outlining the environmental requirements that must also be satisfied to reach high accuracy targets.

Operating System Requirements

High accuracy configurations require Windows 10 or Windows Server 2016. All Windows devices in the time topology must meet this requirement including higher stratum Windows time servers, and in virtualized scenarios, the Hyper-V Hosts that run the time-sensitive virtual machines. All of these devices must be at least Windows 10 or Windows Server 2016.

In the illustration shown below, the virtual machines requiring high accuracy are running Windows 10 or Windows Server 2016. Likewise, the Hyper-V Host on which the virtual machines reside, and the upstream Windows time server must also run Windows Server 2016.



TIP

Determining the Windows Version

You can run the command `winver` at a command prompt to verify the OS version is 1607 (or higher) and OS Build is 14393 (or higher) as shown below:

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>winver
About Windows X

Windows Server® 2016

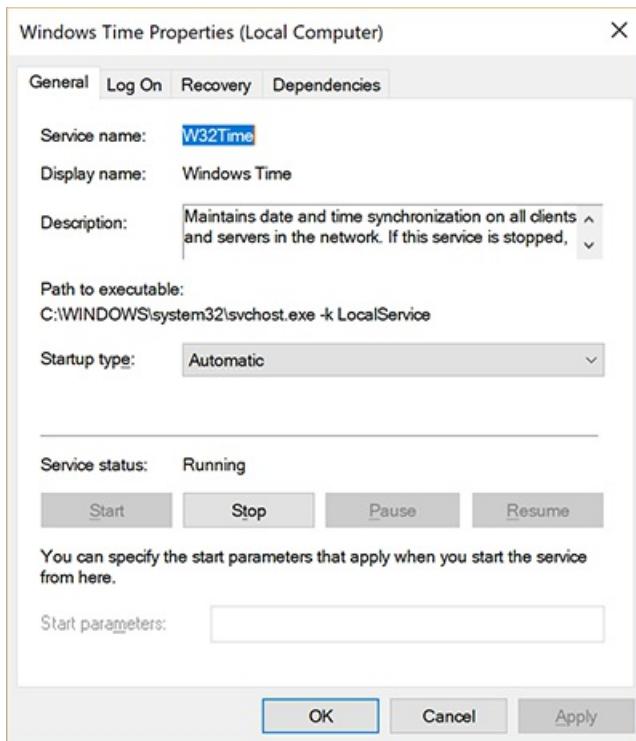
Microsoft Windows Server
Version 1607 (OS Build 14393.1914)
© 2016 Microsoft Corporation. All rights reserved.
The Windows Server 2016 Datacenter Evaluation operating system and its
user interface are protected by trademark and other pending or existing
intellectual property rights in the United States and other
countries/regions.
```

System Configuration

Reaching high accuracy targets requires system configuration. There are a variety of ways to perform this configuration, including directly in the registry or through group policy. More information for each of these settings can be found in the Windows Time Service Technical Reference – [Windows Time Service Tools](#).

Windows Time service Startup Type

The Windows Time service (W32Time) must run continuously. To do this, configure the Windows Time service's startup type to 'Automatic' start.



Cumulative one-way network latency

Measurement uncertainty and "noise" creeps in as network latency increases. As such, it is imperative that a network latency be within a reasonable boundary. The specific requirements are dependent on your target accuracy and are outlined in the [Support boundary to configure the Windows Time service for high-accuracy environments](#) article.

To calculate the cumulative one-way network latency, add the individual one-way delays between pairs of NTP client-server nodes in the time topology, starting with the target and ending at the high-accuracy stratum 1 time source.

For example: Consider a time sync hierarchy with a highly accurate source, two intermediary NTP servers A and B, and the target machine in that order. To obtain the cumulative network latency between the target and source, measure the average individual NTP roundtrip times (RTTs) between:

- The target and time server B
- Time server B and time server A
- Time server A and the Source

This measurement can be obtained using the inbox w32tm.exe tool. To do this:

1. Perform the calculation from the target and time server B.

```
w32tm /stripchart /computer:TimeServerB /rdtsc /samples:450 > c:\temp\Target_TsB.csv
```

2. Perform the calculation from time server b against (pointed at) time server a.

```
w32tm /stripchart /computer:TimeServerA /rdtsc /samples:450 > c:\temp\Target_TsA.csv
```

3. Perform the calculation from time server a against the source.

4. Next, add the average RoundTripDelay measured in the previous step and divide by 2 to obtain the cumulative network delay between target and source.

Registry Settings

- MinPollInterval
- MaxPollInterval
- UpdateInterval

- [SpecialPollInterval](#)
- [FrequencyCorrectRate](#)

Configures the smallest interval in log2 seconds allowed for system polling.

| | |
|--------------|--|
| Key location | HKLM:\SYSTEM\CurrentControlSet\Services\W32Time\Config |
| Setting | 6 |
| Outcome | The minimum polling interval is now 64 seconds. |

The following command signals Windows Time to pick up the updated settings:

```
w32tm /config /update
```

Windows Time for Traceability

9/21/2018 • 7 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016 version 1709 or later, and Windows 10 version 1703 or later

Regulations in many sectors require systems to be traceable to UTC. This means that a system's offset can be attested with respect to UTC. To enable regulatory compliance scenarios, Windows 10 (version 1703 or higher) and Windows Server 2016 (version 1709 or higher) provides new event logs to provide a picture from the perspective of the Operating System to form an understanding of the actions taken on the system clock. These event logs are generated continuously for Windows Time service and can be examined or archived for later analysis.

These new events enable the following questions to be answered:

- Was the system clock altered
- Was the clock frequency modified
- Was the Windows Time service configuration modified

Availability

These improvements are included in Windows 10 version 1703 or higher, and Windows Server 2016 version 1709 or higher.

Configuration

No configuration is required to realize this feature. These event logs are enabled by default and can be found in the event viewer under the **Applications and Services Log\Microsoft\Windows\Time-Service\Operational** channel.

List of Event Logs

The following section outlines the events logged for use in traceability scenarios.

<<<<< HEAD

- [257](#)
- [258](#)
- [259](#)
- [260](#)
- [261](#)
- [262](#)
- [263](#)
- [264](#)
- [265](#)
- [266](#)

This event is logged when the Windows Time Service (W32Time) is started and logs information about the current time, current tick count, runtime configuration, time providers, and current clock rate.

| | |
|----------------------|---|
| Event description | Service Start |
| Details | |
| Data logged | <ul style="list-style-type: none"> • Current Time in UTC • Current Tick Count • W32Time Configuration • Time Provider Configuration • Clock Rate |
| Throttling mechanism | None. This event fires every time the service starts. |

Example:

```
w32time service has started at 2018-02-27T04:25:17.156Z (UTC), System Tick Count 3132937.
```

Command:

This information can also be queried using the following commands

W32Time and Time Provider configuration

```
w32tm.exe /query /configuration
```

Clock Rate

```
w32tm.exe /query /status /verbose
```

=====

1f46ae5f5ef58300d1584b27f08e831eed98d620

Service Start

This event is logged when the Windows Time Service (W32Time) is started and logs information about the current time, current tick count, runtime configuration, time providers, and current clock rate.

| | |
|--------------------|---|
| Supported versions | Windows Server 2016 version 1709 or later, and Windows 10 version 1703 or later |
| Event ID | 257 |
| Event description | Service Start |
| Details | Occurs at W32time Startup |

| | |
|----------------------|---|
| Data logged | <ul style="list-style-type: none"> • Current Time in UTC • Current Tick Count • W32Time Configuration • Time Provider Configuration • Clock Rate |
| Throttling mechanism | None. This event fires every time the service starts. |

Example text: W32time service has started at 2018-02-27T04:25:17.156Z (UTC), System Tick Count 3132937.

Command:

This information can also be queried using the following commands

W32Time and Time Provider configuration `w32tm.exe /query /configuration`

Clock Rate `w32tm.exe /query /status /verbose`

Service Stop

This event is logged when the Windows Time Service (W32Time) is stopping and logs information about the current time and tick count.

| | |
|----------------------|---|
| Supported versions | Windows Server 2016 version 1709 or later, and Windows 10 version 1703 or later |
| Event ID | 258 |
| Event description | Service Stop |
| Details | Occurs at W32time Shutdown |
| Data logged | <ul style="list-style-type: none"> • Current Time in UTC • Current Tick Count |
| Throttling mechanism | None. This event fires every time the service stops. |

Example text: W32time service is stopping at 2018-03-01T05:42:13.944Z (UTC), System Tick Count 6370250.

NTP Status

This event periodically logs its current list of time sources and its chosen time source. In addition, it logs the current tick count. This event does not fire each time a time source changes. Other events listed later in this document provide this functionality.

| | |
|--------------------|---|
| Supported versions | Windows Server 2016 version 1709 or later, and Windows 10 version 1703 or later |
| Event ID | 259 |

| | |
|----------------------|---|
| Event description | NTP Client Provider Periodic Status |
| Details | List of time sources(s) used by NTP Client |
| Data logged | <ul style="list-style-type: none"> • Available time sources • The chosen reference time server at the time of logging • Current Tick Count |
| Throttling mechanism | Logged once every 8 hours. |

Example text: NTP Client provider periodic status:

Ntp Client is receiving time data from the following NTP Servers:

```
server1.fabrikam.com,0x8 (ntp.m|0x8|[:]::123->[IPAddress]:123)server2.fabrikam.com,0x8 (ntp.m|0x8|[:]::123->[IPAddress]:123); and the chosen reference time server is Server1.fabrikam.com,0x8 (ntp.m|0x8|[:]::123->[IPAddress]:123) (RefID:0x08d6648e63). System Tick Count 13187937
```

Command This information can also be queried using the following commands

Identify Peers `w32tm.exe /query /peers`

Service Status

| | |
|----------------------|---|
| Supported versions | Windows Server 2016 version 1709 or later, and Windows 10 version 1703 or later |
| Event ID | 260 |
| Event description | Time service configuration and status |
| Details | W32time periodically logs its configuration and status. This is the equivalent of calling: |
| | <code>w32tm /query /configuration /verbose</code>
OR
<code>w32tm /query /status /verbose</code> |
| Throttling mechanism | Logged once every 8 hours. |

System Time Set

This logs each instance when System Time is modified using SetSystemTime API.

| | |
|--------------------|---|
| Supported versions | Windows Server 2016 version 1709 or later, and Windows 10 version 1703 or later |
| Event ID | 261 |
| Event description | System Time is set |

| | |
|----------------------|---|
| Throttling mechanism | <p>None.</p> <p>This should happen rarely on systems with reasonable time synchronization, and we want to log it each time it occurs. We ignore TimeJumpAuditOffset setting while logging this event since that setting was meant to throttle events in the Windows System event log.</p> |
|----------------------|---|

Frequency Adjustment

| | |
|----------------------|---|
| Supported versions | Windows Server 2016 version 1709 or later, and Windows 10 version 1703 or later |
| Event ID | 262 |
| Event description | System clock frequency adjusted |
| Details | <p>System clock frequency is constantly modified by W32time when the clock is in close synchronization. We want to capture "reasonably significant" adjustments made to the clock frequency without overrunning the event log.</p> |
| Throttling mechanism | <p>All clock adjustments below TimeAdjustmentAuditThreshold (min = 128 part per million, default = 800 part per million) are not logged.</p> <p>2 PPM change in clock frequency with current granularity yields 120 μsec/sec change in clock accuracy.</p> <p>On a synchronized system, the majority of the adjustments are below this level. If you want finer tracking, this setting can be adjusted down or you can use PerfCounters, or you can do both.</p> |

Loaded Providers

| | |
|--------------------|---|
| Supported versions | Windows Server 2016 version 1709 or later, and Windows 10 version 1703 or later |
| Event ID | 263 |
| Event description | Change in the Time service settings or list of loaded time providers. |
| Details | <p>Re-reading W32time settings can cause certain critical settings to be modified in-memory, which can affect the overall accuracy of the time synchronization.</p> <p>W32time logs each occurrence when rereading its settings which gives the potential impact on time synchronization.</p> |

| | |
|----------------------|--|
| Throttling mechanism | <p>None.</p> <p>This event occurs only when an admin or GP update changes the time providers and then triggers W32time. We want to record each instance of change of settings.</p> |
|----------------------|--|

Time Source

| | |
|----------------------|--|
| Supported versions | Windows Server 2016 version 1709 or later, and Windows 10 version 1703 or later |
| Event ID | 264 |
| Event description | Change in time source(s) used by NTP Client |
| Details | NTP Client records an event with the current state of the time servers/peers when a time server/peer changes state
(Pending ->Sync, Sync -> unreachable, or other transitions) |
| Throttling mechanism | Max frequency – only once every 5 minutes to protect the log from transient issues and bad provider implementation. |

Stratum Change

| | |
|----------------------|--|
| Supported versions | Windows Server 2016 version 1709 or later, and Windows 10 version 1703 or later |
| Event ID | 265 |
| Event description | Time service source or stratum number changes |
| Details | W32time Time Source and Stratum Number are important factors in time traceability and any changes to these must be logged. If W32time has no source of time and you have not configured as a reliable time source, then it will stop advertising as a time server, and by-design respond to requests with some invalid parameters. This event is critical to track the state changes in an NTP topology. |
| Throttling mechanism | None. |

Resynchronization Requested

| | |
|--------------------|---|
| Supported versions | Windows Server 2016 version 1709 or later, and Windows 10 version 1703 or later |
| Event ID | 266 |

| | |
|----------------------|--|
| Event description | Time re-synchronization is requested |
| Details | <p>This operation is triggered:</p> <ul style="list-style-type: none"> • When network changes occur • System returns from connected standby/hibernation • When we didn't sync for a long time • Admin issues the resync command <p>This operation results in immediate loss of fine-grained time sync accuracy because it causes NTP client to clear its filters.</p> |
| Throttling mechanism | <p>Max frequency - once every 5 minutes.</p> <p>It is possible that a bad network card (or a poor script) can trigger this operation repeatedly and result in logs getting overwhelmed. Hence the need to throttle this event.</p> <p>Note that accurate time sync takes far more than 5 minutes to achieve, and throttling does not lose information about the original event that resulted in loss of time accuracy.</p> |

Windows Time Service Technical Reference

9/1/2018 • 5 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows 10 or later

The W32Time service provides network clock synchronization for computers without the need for extensive configuration. The W32Time service is essential to the successful operation of Kerberos V5 authentication and, therefore, to AD DS-based authentication. Any Kerberos-aware application, including most security services, relies on time synchronization between the computers that are participating in the authentication request. AD DS domain controllers must also have synchronized clocks to help to ensure accurate data replication.

NOTE

In Windows Server 2003 and Microsoft Windows 2000 Server, the directory service is named Active Directory directory service. In Windows Server 2008 R2 and Windows Server 2008 , the directory service is named Active Directory Domain Services (AD DS). The rest of this topic refers to AD DS, but the information is also applicable to Active Directory Domain Services in Windows Server 2016.

The W32Time service is implemented in a dynamic link library called W32Time.dll, which is installed by default in **%Systemroot%\System32**. W32Time.dll was originally developed for Windows 2000 Server to support a specification by the Kerberos V5 authentication protocol that required clocks on a network to be synchronized. Starting with Windows Server 2003, W32Time.dll provided increased accuracy in network clock synchronization over the Windows Server 2000 operating system. Additionally, in Windows Server 2003, W32Time.dll supported a variety of hardware devices and network time protocols using time providers.

Although originally designed to provide clock synchronization for Kerberos authentication, many current applications use timestamps to ensure transactional consistency, record the time of important events, and other business-critical, time-sensitive information. These applications benefit from time synchronization between computers that are provided by the Windows Time service.

Importance of Time Protocols

Time protocols communicate between two computers to exchange time information and then use that information to synchronize their clocks. With the Windows Time service time protocol, a client requests time information from a server and synchronizes its clock based on the information that is received.

The Windows Time service uses NTP to help synchronize time across a network. NTP is an Internet time protocol that includes the discipline algorithms necessary for synchronizing clocks. NTP is a more accurate time protocol than the Simple Network Time Protocol (SNTP) that is used in some versions of Windows; however, W32Time continues to support SNTP to enable backward compatibility with computers running SNTP-based time services such as Windows 2000.

Where to find Windows Time service configuration-related information

This guide does **not** discuss configuring the Windows Time service. There are several different topics on Microsoft TechNet and in the Microsoft Knowledge Base that do explain procedures for configuring the Windows Time service. If you require configuration information, the following topics should help you locate the appropriate information.

- To configure the Windows Time service for the forest root primary domain controller (PDC) emulator, see:

- [Configure the Windows Time service on the PDC emulator in the Forest Root Domain](#)
- [Configuring a time source for the forest](#)
- Microsoft Knowledge Base article 816042, [How to configure an authoritative time server in Windows Server](#), which describes configuration settings for computers running Windows Server 2008 R2, Windows Server 2008, Windows Server 2003, and Windows Server 2003 R2.
- To configure the Windows Time service on any domain member client or server, or even domain controllers that are not configured as the forest root PDC emulator, see [Configure a client computer for automatic domain time synchronization](#).

WARNING

Some applications may require their computers to have high-accuracy time services. If that is the case, you may choose to configure a manual time source, but be aware that the Windows Time service was not designed to function as a highly accurate time source. Ensure that you are aware of the support limitations for high-accuracy time environments as described in Microsoft Knowledge Base article 939322, [Support boundary to configure the Windows Time service for high-accuracy environments](#).

- To configure the Windows Time service on any Windows-based client or server computers that are configured as workgroup members instead of domain members see [Configure a manual time source for a selected client computer](#).
- To configure the Windows Time service on a host computer that runs a virtual environment, see Microsoft Knowledge Base article 816042, [How to configure an authoritative time server in Windows Server](#). If you are working with a non-Microsoft virtualization product, be sure to consult the documentation of the vendor for that product.
- To configure the Windows Time service on a domain controller that is running in a virtual machine, it is recommended that you partially disable time synchronization between the host system and guest operating system acting as a domain controller. This enables your guest domain controller to synchronize time for the domain hierarchy, but protects it from having a time skew if it is restored from a Saved state. For more information, see Microsoft Knowledge Base article 976924, [You receive Windows Time Service event IDs 24, 29, and 38 on a virtualized domain controller that is running on a Windows Server 2008-based host server with Hyper-V](#) and [Deployment Considerations for Virtualized Domain Controllers](#).
- To configure the Windows Time service on a domain controller acting as the forest root PDC emulator that is also running in a virtual computer, follow the same instructions for a physical computer as described in [Configure the Windows Time service on the PDC emulator in the Forest Root Domain](#).
- To configure the Windows Time service on a member server running as a virtual computer, use the domain time hierarchy as described in [Configure a client computer for automatic domain time synchronization](#).

IMPORTANT

Prior to Windows Server 2016, the W32Time service was not designed to meet time-sensitive application needs. However, updates to Windows Server 2016 now allow you to implement a solution for 1ms accuracy in your domain. For more information about, see [Windows 2016 Accurate Time](#) and [Support boundary to configure the Windows Time service for high-accuracy environments](#) for more information.

Related topics

- [Windows 2016 Accurate Time](#)
- [Time Accuracy Improvements for Windows Server 2016](#)

- [How the Windows Time Service Works](#)
- [Windows Time Service Tools and Settings](#)
- [Support boundary to configure the Windows Time service for high-accuracy environments](#)
- [Microsoft Knowledge Base article 902229](#)

How the Windows Time Service Works

9/21/2018 • 23 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows 10 or later

In this section

- [Windows Time Service Architecture](#)
- [Windows Time Service Time Protocols](#)
- [Windows Time Service Processes and Interactions](#)
- [Network Ports Used by Windows Time Service](#)

NOTE

This topic explains only how the Windows Time service (W32Time) works. For information about how to configure Windows Time service, see [Configuring Systems for High Accuracy](#).

NOTE

In Windows Server 2003 and Microsoft Windows 2000 Server, the directory service is named Active Directory directory service. In Windows Server 2008 and later versions, the directory service is named Active Directory Domain Services (AD DS). The rest of this topic refers to AD DS, but the information is also applicable to Active Directory.

Although the Windows Time service is not an exact implementation of the Network Time Protocol (NTP), it uses the complex suite of algorithms that is defined in the NTP specifications to ensure that clocks on computers throughout a network are as accurate as possible. Ideally, all computer clocks in an AD DS domain are synchronized with the time of an authoritative computer. Many factors can affect time synchronization on a network. The following factors often affect the accuracy of synchronization in AD DS:

- Network conditions
- The accuracy of the computer's hardware clock
- The amount of CPU and network resources available to the Windows Time service

IMPORTANT

Prior to Windows Server 2016, the W32Time service was not designed to meet time-sensitive application needs. However, updates to Windows Server 2016 now allow you to implement a solution for 1ms accuracy in your domain. See [Windows 2016 Accurate Time](#) and [Support boundary to configure the Windows Time service for high-accuracy environments](#) for more information.

Computers that synchronize their time less frequently or are not joined to a domain are configured, by default, to synchronize with time.windows.com. Therefore, it is impossible to guarantee time accuracy on computers that have intermittent or no network connections.

An AD DS forest has a predetermined time synchronization hierarchy. The Windows Time service synchronizes time between computers within the hierarchy, with the most accurate reference clocks at the top. If more than one

time source is configured on a computer, Windows Time uses NTP algorithms to select the best time source from the configured sources based on the computer's ability to synchronize with that time source. The Windows Time service does not support network synchronization from broadcast or multicast peers. For more information about these NTP features, see RFC 1305 in the IETF RFC Database.

Every computer that is running the Windows Time service uses the service to maintain the most accurate time. Computers that are members of a domain act as a time client by default, therefore, in most cases it is not necessary to configure the Windows Time Service. However, the Windows Time Service can be configured to request time from a designated reference time source, and can also provide time to clients.

The degree to which a computer's time is accurate is called a stratum. The most accurate time source on a network (such as a hardware clock) occupies the lowest stratum level, or stratum one. This accurate time source is called a reference clock. An NTP server that acquires its time directly from a reference clock occupies a stratum that is one level higher than that of the reference clock. Resources that acquire time from the NTP server are two steps away from the reference clock, and therefore occupy a stratum that is two higher than the most accurate time source, and so on. As a computer's stratum number increases, the time on its system clock may become less accurate. Therefore, the stratum level of any computer is an indicator of how closely that computer is synchronized with the most accurate time source.

When the W32Time Manager receives time samples, it uses special algorithms in NTP to determine which of the time samples is the most appropriate for use. The time service also uses another set of algorithms to determine which of the configured time sources is the most accurate. When the time service has determined which time sample is best, based on the above criteria, it adjusts the local clock rate to allow it to converge toward the correct time. If the time difference between the local clock and the selected accurate time sample (also called the time skew) is too large to correct by adjusting the local clock rate, the time service sets the local clock to the correct time. This adjustment of clock rate or direct clock time change is known as clock discipline.

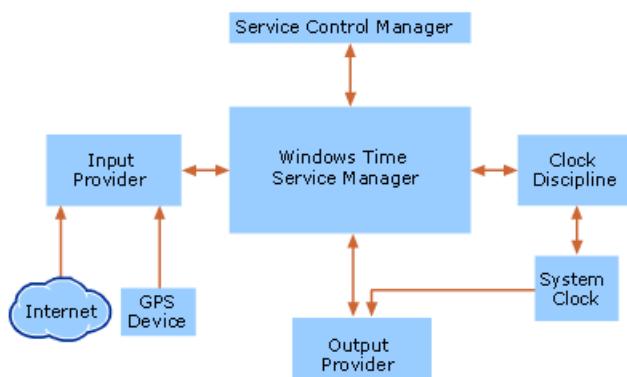
Windows Time Service Architecture

The Windows Time service consists of the following components:

- Service Control Manager
- Windows Time Service Manager
- Clock Discipline
- Time providers

The following figure shows the architecture of the Windows Time service.

Windows Time Service Architecture



The Service Control Manager is responsible for starting and stopping the Windows Time service. The Windows Time Service Manager is responsible for initiating the action of the NTP time providers included with the operating system. The Windows Time Service Manager controls all functions of the Windows Time service and the coalescing

of all time samples. In addition to providing information about the current system state, such as the current time source or the last time the system clock was updated, the Windows Time Service Manager is also responsible for creating events in the event log.

The time synchronization process involves the following steps:

- Input providers request and receive time samples from configured NTP time sources.
- These time samples are then passed to the Windows Time Service Manager, which collects all the samples and passes them to the clock discipline subcomponent.
- The clock discipline subcomponent applies the NTP algorithms which results in the selection of the best time sample.
- The clock discipline subcomponent adjusts the time of the system clock to the most accurate time by either adjusting the clock rate or directly changing the time.

If a computer has been designated as a time server, it can send the time on to any computer requesting time synchronization at any point in this process.

Windows Time Service Time Protocols

Time protocols determine how closely two computers' clocks are synchronized. A time protocol is responsible for determining the best available time information and converging the clocks to ensure that a consistent time is maintained on separate systems.

The Windows Time service uses the Network Time Protocol (NTP) to help synchronize time across a network. NTP is an Internet time protocol that includes the discipline algorithms necessary for synchronizing clocks. NTP is a more accurate time protocol than the Simple Network Time Protocol (SNTP) that is used in some versions of Windows; however W32Time continues to support SNTP to enable backward compatibility with computers running SNTP-based time services, such as Windows 2000.

Network Time Protocol

Network Time Protocol (NTP) is the default time synchronization protocol used by the Windows Time service in the operating system. NTP is a fault-tolerant, highly scalable time protocol and is the protocol used most often for synchronizing computer clocks by using a designated time reference.

NTP time synchronization takes place over a period of time and involves the transfer of NTP packets over a network. NTP packets contain time stamps that include a time sample from both the client and the server participating in time synchronization.

NTP relies on a reference clock to define the most accurate time to be used and synchronizes all clocks on a network to that reference clock. NTP uses Coordinated Universal Time (UTC) as the universal standard for current time. UTC is independent of time zones and enables NTP to be used anywhere in the world regardless of time zone settings.

NTP Algorithms

NTP includes two algorithms, a clock-filtering algorithm and a clock-selection algorithm, to assist the Windows Time service in determining the best time sample. The clock-filtering algorithm is designed to sift through time samples that are received from queried time sources and determine the best time samples from each source. The clock-selection algorithm then determines the most accurate time server on the network. This information is then passed to the clock discipline algorithm, which uses the information gathered to correct the local clock of the computer, while compensating for errors due to network latency and computer clock inaccuracy.

The NTP algorithms are most accurate under conditions of light-to-moderate network and server loads. As with any algorithm that takes network transit time into account, NTP algorithms might perform poorly under conditions of extreme network congestion. For more information about the NTP algorithms, see RFC 1305 in the IETF RFC

Database.

NTP Time Provider

The Windows Time service is a complete time synchronization package that can support a variety of hardware devices and time protocols. To enable this support, the service uses pluggable time providers. A time provider is responsible for either obtaining accurate time stamps (from the network or from hardware) or for providing those time stamps to other computers over the network.

The NTP provider is the standard time provider included with the operating system. The NTP provider follows the standards specified by NTP version 3 for a client and server, and can interact with SNTP clients and servers for backward compatibility with Windows 2000 and other SNTP clients. The NTP provider in the Windows Time service consists of the following two parts:

- **NtpServer output provider.** This is a time server that responds to client time requests on the network.
- **NtpClient input provider.** This is a time client that obtains time information from another source, either a hardware device or an NTP server, and can return time samples that are useful for synchronizing the local clock.

Although the actual operations of these two providers are closely related, they appear independent to the time service. Starting with Windows 2000 Server, when a Windows computer is connected to a network, it is configured as an NTP client. Also, computers running the Windows Time service only attempt to synchronize time with a domain controller or a manually specified time source by default. These are the preferred time providers because they are automatically available, secure sources of time.

NTP Security

Within an AD DS forest, the Windows Time service relies on standard domain security features to enforce the authentication of time data. The security of NTP packets that are sent between a domain member computer and a local domain controller that is acting as a time server is based on shared key authentication. The Windows Time service uses the computer's Kerberos session key to create authenticated signatures on NTP packets that are sent across the network. NTP packets are not transmitted inside the Net Logon secure channel. Instead, when a computer requests the time from a domain controller in the domain hierarchy, the Windows Time service requires that the time be authenticated. The domain controller then returns the required information in the form of a 64-bit value that has been authenticated with the session key from the Net Logon service. If the returned NTP packet is not signed with the computer's session key or is signed incorrectly, the time is rejected. All such authentication failures are logged in the Event Log. In this way, the Windows Time service provides security for NTP data in an AD DS forest.

Generally, Windows time clients automatically obtain accurate time for synchronization from domain controllers in the same domain. In a forest, the domain controllers of a child domain synchronize time with domain controllers in their parent domains. When a time server returns an authenticated NTP packet to a client that requests the time, the packet is signed by means of a Kerberos session key defined by an interdomain trust account. The interdomain trust account is created when a new AD DS domain joins a forest, and the Net Logon service manages the session key. In this way, the domain controller that is configured as reliable in the forest root domain becomes the authenticated time source for all of the domain controllers in both the parent and child domains, and indirectly for all computers located in the domain tree.

The Windows Time service can be configured to work between forests, but it is important to note that this configuration is not secure. For example, an NTP server might be available in a different forest. However, because that computer is in a different forest, there is no Kerberos session key with which to sign and authenticate NTP packets. To obtain accurate time synchronization from a computer in a different forest, the client needs network access to that computer and the time service must be configured to use a specific time source located in the other forest. If a client is manually configured to access time from an NTP server outside of its own domain hierarchy, the NTP packets sent between the client and the time server are not authenticated, and therefore are not secure. Even with the implementation of forest trusts, the Windows Time service is not secure across forests. Although the Net Logon secure channel is the authentication mechanism for the Windows Time service, authentication across forests

is not supported.

Hardware Devices That Are Supported by the Windows Time Service

Hardware-based clocks such as GPS or radio clocks are often used as highly accurate reference clock devices. By default, the Windows Time service NTP time provider does not support the direct connection of a hardware device to a computer, although it is possible to create a software-based independent time provider that supports this type of connection. This type of provider, in conjunction with the Windows Time service, can provide a reliable, stable time reference.

Hardware devices, such as a cesium clock or a Global Positioning System (GPS) receiver, provide accurate current time by following a standard to obtain an accurate definition of time. Cesium clocks are extremely stable and are unaffected by factors such as temperature, pressure, or humidity, but are also very expensive. A GPS receiver is much less expensive to operate and is also an accurate reference clock. GPS receivers obtain their time from satellites that obtain their time from a cesium clock. Without the use of an independent time provider, Windows time servers can acquire their time by connecting to an external NTP server, which is connected to a hardware device by means of a telephone or the Internet. Organizations such as the United States Naval Observatory provide NTP servers that are connected to extremely reliable reference clocks.

Many GPS receivers and other time devices can function as NTP servers on a network. You can configure your AD DS forest to synchronize time from these external hardware devices only if they are also acting as NTP servers on your network. To do so, configure the domain controller functioning as the primary domain controller (PDC) emulator in your forest root to synchronize with the NTP server provided by the GPS device. To do so, see [Configure the Windows Time service on the PDC emulator in the Forest Root Domain](#).

Simple Network Time Protocol

The Simple Network Time Protocol (SNTP) is a simplified time protocol that is intended for servers and clients that do not require the degree of accuracy that NTP provides. SNTP, a more rudimentary version of NTP, is the primary time protocol that is used in Windows 2000. Because the network packet formats of SNTP and NTP are identical, the two protocols are interoperable. The primary difference between the two is that SNTP does not have the error management and complex filtering systems that NTP provides. For more information about the Simple Network Time Protocol, see RFC 1769 in the IETF RFC Database.

Time Protocol Interoperability

The Windows Time service can operate in a mixed environment of computers running Windows 2000, Windows XP, and Windows Server 2003, because the SNTP protocol used in Windows 2000 is interoperable with the NTP protocol in Windows XP and Windows Server 2003.

The time service in Windows NT Server 4.0, called TimeServ, synchronizes time across a Windows NT 4.0 network. TimeServ is an add-on feature available as part of the *Microsoft Windows NT 4.0 Resource Kit* and does not provide the degree of reliability of time synchronization that is required by Windows Server 2003.

The Windows Time service can interoperate with computers running Windows NT 4.0 because they can synchronize time with computers running Windows 2000 or Windows Server 2003; however, a computer running Windows 2000 or Windows Server 2003 does not automatically discover Windows NT 4.0 time servers. For example, if your domain is configured to synchronize time by using the domain hierarchy-based method of synchronization and you want computers in the domain hierarchy to synchronize time with a Windows NT 4.0 domain controller, you have to configure those computers manually to synchronize with the Windows NT 4.0 domain controllers.

Windows NT 4.0 uses a simpler mechanism for time synchronization than the Windows Time service uses. Therefore, to ensure accurate time synchronization across your network, it is recommended that you upgrade any Windows NT 4.0 domain controllers to Windows 2000 or Windows Server 2003.

Windows Time Service Processes and Interactions

The Windows Time service is designed to synchronize the clocks of computers on a network. The network time synchronization process, also called time convergence, occurs throughout a network as each computer accesses time from a more accurate time server. Time convergence involves a process by which an authoritative server provides the current time to client computers in the form of NTP packets. The information provided within a packet indicates whether an adjustment needs to be made to the computer's current clock time so that it is synchronized with the more accurate server.

As part of the time convergence process, domain members attempt to synchronize time with any domain controller located in the same domain. If the computer is a domain controller, it attempts to synchronize with a more authoritative domain controller.

Computers running Windows XP Home Edition or computers that are not joined to a domain do not attempt to synchronize with the domain hierarchy, but are configured by default to obtain time from time.windows.com.

To establish a computer running Windows Server 2003 as authoritative, the computer must be configured to be a reliable time source. By default, the first domain controller that is installed on a Windows Server 2003 domain is automatically configured to be a reliable time source. Because it is the authoritative computer for the domain, it must be configured to synchronize with an external time source rather than with the domain hierarchy. Also by default, all other Windows Server 2003 domain members are configured to synchronize with the domain hierarchy.

After you have established a Windows Server 2003 network, you can configure the Windows Time service to use one of the following options for synchronization:

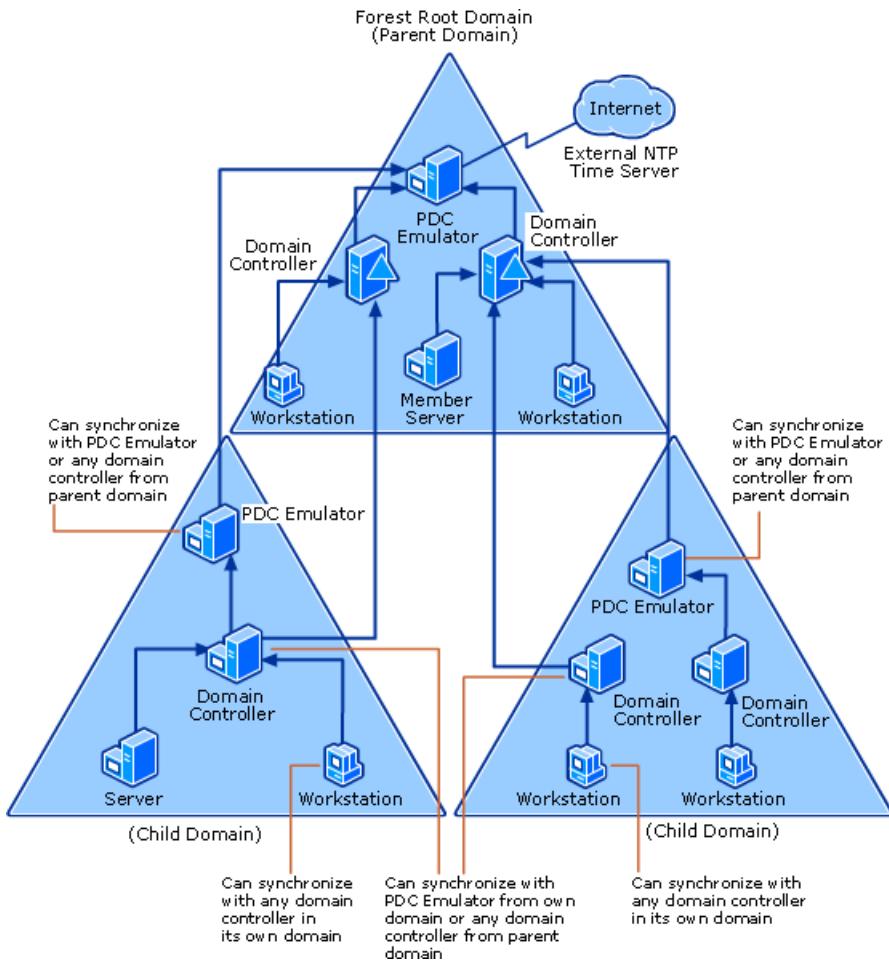
- Domain hierarchy-based synchronization
- A manually-specified synchronization source
- All available synchronization mechanisms
- No synchronization.

Each of these synchronization types is discussed in the following section.

Domain Hierarchy-Based Synchronization

Synchronization that is based on a domain hierarchy uses the AD DS domain hierarchy to find a reliable source with which to synchronize time. Based on domain hierarchy, the Windows Time service determines the accuracy of each time server. In a Windows Server 2003 forest, the computer that holds the primary domain controller (PDC) emulator operations master role, located in the forest root domain, holds the position of best time source, unless another reliable time source has been configured. The following figure illustrates a path of time synchronization between computers in a domain hierarchy.

Time Synchronization in an AD DS Hierarchy



Reliable Time Source Configuration

A computer that is configured to be a reliable time source is identified as the root of the time service. The root of the time service is the authoritative server for the domain and typically is configured to retrieve time from an external NTP server or hardware device. A time server can be configured as a reliable time source to optimize how time is transferred throughout the domain hierarchy. If a domain controller is configured to be a reliable time source, Net Logon service announces that domain controller as a reliable time source when it logs on to the network. When other domain controllers look for a time source to synchronize with, they choose a reliable source first if one is available.

Time Source Selection

The time source selection process can create two problems on a network:

- Additional synchronization cycles.
- Increased volume in network traffic.

A cycle in the synchronization network occurs when time remains consistent between a group of domain controllers and the same time is shared between them continuously without a resynchronization with another reliable time source. The Windows Time service's time source selection algorithm is designed to protect against these types of problems.

A computer uses one of the following methods to identify a time source to synchronize with:

- If the computer is not a member of a domain, it must be configured to synchronize with a specified time source.
- If the computer is a member server or workstation within a domain, by default, it follows the AD DS hierarchy and synchronizes its time with a domain controller in its local domain that is currently running the Windows Time service.

If the computer is a domain controller, it makes up to six queries to locate another domain controller to synchronize

with. Each query is designed to identify a time source with certain attributes, such as a type of domain controller, a particular location, and whether or not it is a reliable time source. The time source must also adhere to the following constraints:

- A reliable time source can only synchronize with a domain controller in the parent domain.
- A PDC emulator can synchronize with a reliable time source in its own domain or any domain controller in the parent domain.

If the domain controller is not able to synchronize with the type of domain controller that it is querying, the query is not made. The domain controller knows which type of computer it can obtain time from before it makes the query. For example, a local PDC emulator does not attempt to query numbers three or six because a domain controller does not attempt to synchronize with itself.

The following table lists the queries that a domain controller makes to find a time source and the order in which the queries are made.

Domain Controller Time Source Queries

| QUERY NUMBER | DOMAIN CONTROLLER | LOCATION | RELIABILITY OF TIME SOURCE |
|--------------|--------------------------|-------------|---|
| 1 | Parent domain controller | In-site | Prefers a reliable time source but it can synchronize with a non-reliable time source if that is all that is available. |
| 2 | Local domain controller | In-site | Only synchronizes with a reliable time source. |
| 3 | Local PDC emulator | In-site | Does not apply.
A domain controller does not attempt to synchronize with itself. |
| 4 | Parent domain controller | Out-of-site | Prefers a reliable time source but it can synchronize with a non-reliable time source if that is all that is available. |
| 5 | Local domain controller | Out-of-site | Only synchronizes with a reliable time source. |
| 6 | Local PDC emulator | Out-of-site | Does not apply.
A domain controller does not attempt to synchronize with itself. |

Note

- A computer never synchronizes with itself. If the computer attempting synchronization is the local PDC emulator, it does not attempt Queries 3 or 6.

Each query returns a list of domain controllers that can be used as a time source. Windows Time assigns each domain controller that is queried a score based on the reliability and location of the domain controller. The following table lists the scores assigned by Windows Time to each type of domain controller.

Score Determination

| DOMAIN CONTROLLER STATUS | SCORE |
|--|-------|
| Domain controller located in same site | 8 |
| Domain controller marked as a reliable time source | 4 |
| Domain controller located in the parent domain | 2 |
| Domain controller that is a PDC emulator | 1 |

When the Windows Time service determines that it has identified the domain controller with the best possible score, no more queries are made. The scores assigned by the time service are cumulative, which means that a PDC emulator located in the same site receives a score of nine.

If the root of the time service is not configured to synchronize with an external source, the internal hardware clock of the computer governs the time.

Manually-Specified Synchronization

Manually-specified synchronization enables you to designate a single peer or list of peers from which a computer obtains time. If the computer is not a member of a domain, it must be manually configured to synchronize with a specified time source. A computer that is a member of a domain is configured by default to synchronize from the domain hierarchy, manually-specified synchronization is most useful for the forest root of the domain or for computers that are not joined to a domain. Manually specifying an external NTP server to synchronize with the authoritative computer for your domain provides reliable time. However, configuring the authoritative computer for your domain to synchronize with a hardware clock is actually a better solution for providing the most accurate, secure time to your domain.

Manually-specified time sources are not authenticated unless a specific time provider is written for them, and they are therefore vulnerable to attackers. Also, if a computer synchronizes with a manually-specified source rather than its authenticating domain controller, the two computers might be out of synchronization, causing Kerberos authentication to fail. This might cause other actions requiring network authentication to fail, such as printing or file sharing. If only the forest root is configured to synchronize with an external source, all other computers within the forest remain synchronized with each other, making replay attacks difficult.

All Available Synchronization Mechanisms

The "all available synchronization mechanisms" option is the most valuable synchronization method for users on a network. This method allows synchronization with the domain hierarchy and may also provide an alternate time source if the domain hierarchy becomes unavailable, depending on the configuration. If the client is unable to synchronize time with the domain hierarchy, the time source automatically falls back to the time source specified by the **NtpServer** setting. This method of synchronization is most likely to provide accurate time to clients.

Stopping Time Synchronization

There are certain situations in which you will want to stop a computer from synchronizing its time. For example, if a computer attempts to synchronize from a time source on the Internet or from another site over a WAN by means of a dial-up connection, it can incur costly telephone charges. When you disable synchronization on that computer, you prevent the computer from attempting to access a time source over a dial-up connection.

You can also disable synchronization to prevent the generation of errors in the event log. Each time a computer attempts to synchronize with a time source that is unavailable, it generates an error in the Event Log. If a time source is taken off of the network for scheduled maintenance and you do not intend to reconfigure the client to synchronize from another source, you can disable synchronization on the client to prevent it from attempting synchronization while the time server is unavailable.

It is useful to disable synchronization on the computer that is designated as the root of the synchronization

network. This indicates that the root computer trusts its local clock. If the root of the synchronization hierarchy is not set to **NoSync** and if it is unable to synchronize with another time source, clients do not accept the packet that this computer sends out because its time cannot be trusted.

The only time servers that are trusted by clients even if they have not synchronized with another time source are those that have been identified by the client as reliable time servers.

Disabling the Windows Time Service

The Windows Time service (W32Time) can be completely disabled. If you choose to implement a third-party time synchronization product that uses NTP, you must disable the Windows Time service. This is because all NTP servers need access to User Datagram Protocol (UDP) port 123, and as long as the Windows Time service is running on the Windows Server 2003 operating system, port 123 remains reserved by Windows Time.

Network Ports Used by Windows Time Service

The Windows Time service communicates on a network to identify reliable time sources, obtain time information, and provide time information to other computers. It performs this communication as defined by the NTP and SNTP RFCs.

Port Assignments for the Windows Time Service

| SERVICE NAME | UDP | TCP |
|--------------|-----|-----|
| NTP | 123 | N/A |
| SNTP | 123 | N/A |

See Also

[Windows Time Service Technical Reference](#) [Windows Time Service Tools and Settings](#) [Microsoft Knowledge Base article 902229](#)

Windows Time Service Tools and Settings

9/21/2018 • 27 minutes to read • [Edit Online](#)

Applies to: Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows 10 or later

In this topic, you learn about tools and settings for Windows Time service (W32Time).

If you only want to synchronize time for a domain-joined client computer, see [Configure a client computer for automatic domain time synchronization](#). For additional topics about how to configure Windows Time service, see [Where to Find Windows Time Service Configuration Information](#).

Caution

You should not use the Net time command to configure or set time when the Windows Time service is running.

Also, on older computers that run Windows XP or earlier, the command Net time /querysnntp displays the name of a Network Time Protocol (NTP) server with which a computer is configured to synchronize, but that NTP server is used only when the computer's time client is configured as NTP or AllSync. That command has since been deprecated.

Most domain member computers have a time client type of NT5DS, which means that they synchronize time from the domain hierarchy. The only typical exception to this is the domain controller that functions as the primary domain controller (PDC) emulator operations master of the forest root domain, which is usually configured to synchronize time with an external time source. To view the time client configuration of a computer, run the W32tm /query /configuration command from an elevated Command Prompt in starting in Windows Server 2008, and Windows Vista, and read the **Type** line in the command output. For more information, see [How Windows Time Service Works](#). You can run the command **reg query**

HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Parameters and read the value of **NtpServer** in the command output.

IMPORTANT

Prior to Windows Server 2016, the W32Time service was not designed to meet time-sensitive application needs. However, updates to Windows Server 2016 now allow you to implement a solution for 1ms accuracy in your domain. See [Windows 2016 Accurate Time](#) and [Support boundary to configure the Windows Time service for high-accuracy environments](#) for more information.

Windows Time Service Tools

The following tools are associated with the Windows Time service.

W32tm.exe: Windows Time

Category

This tool is installed as part of Windows XP, Windows Vista, Windows 7 , Windows Server 2003, Windows Server 2003 R2, Windows Server 2008 , and Windows Server 2008 R2 default installations.

Version compatibility

This tool works on Windows XP, Windows Vista, Windows 7 , Windows Server 2003, Windows Server 2003 R2, Windows Server 2008 , and Windows Server 2008 R2 default installations.

W32tm.exe is used to configure Windows Time service settings. It can also be used to diagnose problems with the time service. W32tm.exe is the preferred command line tool for configuring, monitoring, or troubleshooting the

Windows Time service.

The following tables describe the parameters that are used with W32tm.exe.

W32tm.exe Primary Parameters

| PARAMETER | DESCRIPTION |
|--|---|
| W32tm /? | W32tm command line help |
| W32tm /register | Registers the time service to run as a service and adds default configuration to the registry. |
| W32tm /unregister | Unregisters the time service and removes all configuration information from the registry. |
| w32tm /monitor
[/domain:] [/computers:[,...]] [/threads:] | domain - specifies which domain to monitor. If no domain name is given, or neither the domain nor computers option is specified, the default domain is used. This option might be used more than once.

computers - monitors the given list of computers. Computer names are separated by commas, with no spaces. If a name is prefixed with a '*', it is treated as a PDC. This option might be used more than once.

threads - specifies the number of computers to analyze simultaneously. The default value is 3. Allowed range is 1-50. |
| w32tm /ntte | Convert an NT system time, in (10^-7)s intervals from 0h 1-Jan 1601, into a readable format. |
| w32tm /ntpe | Convert an NTP time, in (2^-32)s intervals from 0h 1-Jan 1900, into a readable format. |
| w32tm /resync
[/computer:]
[/nowait]
[/rediscover]
[/soft] | Tells a computer that it should resynchronize its clock as soon as possible, throwing out all accumulated error statistics.

computer: - Specifies the computer that should resynchronize. If not specified, the local computer will resynchronize.

nowait - do not wait for the resynchronize to occur; return immediately. Otherwise, wait for the resynchronize to complete before returning.

rediscover - Redetect the network configuration and rediscover network sources, then resynchronize.

soft - resynchronize using existing error statistics. Not useful, provided for compatibility. |

| PARAMETER | DESCRIPTION |
|--------------------------|--|
| w32tm /stripchart | Display a strip chart of the offset between this computer and another computer. |
| /computer:
[/period:] | computer: - the computer to measure the offset against.
period: - the time between samples, in seconds. The default is 2s. |
| [/dataonly] | dataonly - display only the data without graphics. |
| [/samples:] | samples: - collect samples, then stop. If not specified, samples will be collected until Ctrl+C is pressed. |
| [/rdtsc] | <p>rdtsc: for each sample, this option prints comma separated values along with the headers RdtscStart, RdtscEnd, FileTime, RoundtripDelay, NtpOffset instead of the text graphic.</p> <ul style="list-style-type: none"> • RdtscStart – RDTSC (Read TimeStamp Counter) value collected just before the NTP request was generated. • RdtscEnd – RDTSC (Read TimeStamp Counter) value collected just after the NTP response was received and processed. • FileTime – Local FILETIME value used in the NTP request. • RoundtripDelay – Time elapsed in seconds between generating the NTP request and processing the received NTP response, computed as per NTP roundtrip computations. • NtpOffset – Time offset in seconds between the local machine and the NTP server, computed as per NTP offset computations. |

| PARAMETER | DESCRIPTION |
|---|--|
| w32tm /config
[/computer:] | computer: - adjusts the configuration of . If not specified, the default is the local computer. |
| [/update]
[/manualpeerlist:] | update - notifies the time service that the configuration has changed, causing the changes to take effect. |
| [/syncfromflags:]
[/LocalClockDispersion:] | manualpeerlist: - sets the manual peer list to , which is a space-delimited list of DNS and/or IP addresses. When specifying multiple peers, this option must be enclosed in quotes. |
| [/reliable:(YES NO)]
[/largephaseoffset:] | syncfromflags: - sets what sources the NTP client should synchronize from. should be a comma separated list of these keywords (not case sensitive):

MANUAL - include peers from the manual peer list. |
| | DOMHIER - synchronize from a domain controller (DC) in the domain hierarchy. |
| | LocalClockDispersion: - configures the accuracy of the internal clock that W32Time will assume when it can't acquire time from its configured sources. |
| | reliable:(YES NO) - set whether this computer is a reliable time source. |
| | This setting is only meaningful on domain controllers.

YES - this computer is a reliable time service. |
| | NO - this computer is not a reliable time service.

largephaseoffset: - sets the time difference between local and network time which W32Time will consider a spike. |
| w32tm /tz | Display the current time zone settings. |
| w32tm /dumpreg
[/subkey:] | Display the values associated with a given registry key.

The default key is
HKLM\System\CurrentControlSet\Services\W32Time |
| [/computer:] | (the root key for the time service).

subkey: - displays the values associated with subkey of the default key. |
| | computer: - queries registry settings for computer |

| PARAMETER | DESCRIPTION |
|--|---|
| w32tm /query [/computer:] {/source /configuration /peers /status} [/verbose] | <p>This parameter was first made available in the Windows Time client versions of Windows Vista, and Windows Server 2008 .</p> <p>Display a computer's Windows Time service information.</p> <p>computer: - Query the information of . If not specified, the default value is the local computer.</p> <p>Source - Display the time source.</p> <p>Configuration - Display the configuration of run time and where the setting comes from. In verbose mode, display the undefined or unused setting too.</p> <p>peers - Display a list of peers and their status.</p> <p>status - Display Windows Time service status.</p> <p>verbose - Set the verbose mode to display more information.</p> |
| w32tm /debug {/disable {/enable /file: /size: /entries: [/truncate]}} | <p>This parameter was first made available in the Windows Time client versions of Windows Vista, and Windows Server 2008 .</p> <p>Enable or disable the local computer Windows Time service private log.</p> <p>disable - Disable the private log.</p> <p>enable - Enable the private log.</p> <ul style="list-style-type: none"> - file: - Specify the absolute file name. - size: - Specify the maximum size for circular logging. - entries: - Contains a list of flags, specified by number and separated by commas, that specify the types of information that should be logged. Valid numbers are 0 to 300. A range of numbers is valid, in addition to single numbers, such as 0-100,103,106. Value 0-300 is for logging all information. <p>truncate - Truncate the file if it exists.</p> |

For more information about **W32tm.exe**, see Help and Support Center in Windows XP, Windows Vista, Windows 7 , Windows Server 2003, Windows Server 2003 R2, Windows Server 2008 , and Windows Server 2008 R2.

Windows Time Service Registry Entries

The following registry entries are associated with the Windows Time service.

This information is provided as a reference for use in troubleshooting or verifying that the required settings are applied. It is recommended that you do not directly edit the registry unless there is no other alternative.

Modifications to the registry are not validated by the registry editor or by Windows before they are applied, and as a result, incorrect values can be stored. This can result in unrecoverable errors in the system.

When possible, use Group Policy or other Windows tools, such as Microsoft Management Console (MMC), to accomplish tasks rather than editing the registry directly. If you must edit the registry, use extreme caution.

WARNING

Some of the preset values that are configured in the System Administrative template file (System.adm) for the Group Policy object (GPO) settings are different from the corresponding default registry entries. If you plan to use a GPO to configure any Windows Time setting, be sure that you review [Preset values for the Windows Time service Group Policy settings are different from the corresponding Windows Time service registry entries in Windows Server 2003](#). This issue applies to Windows Server 2008 R2 , Windows Server 2008 , Windows Server 2003 R2, and Windows Server 2003.

Many registry entries for the Windows Time service are the same as the Group Policy setting of the same name. The Group Policy settings correspond to the registry entries of the same name located in:

HKLM\SYSTEM\CurrentControlSet\Services\W32Time

There are several registry keys at this registry location. The Windows Time settings are stored in values across all of these keys:

- [Parameters](#)
- [Config](#)
- [NtpClient](#)
- [NtpServer](#)

Many of the values in the W32Time section of the registry are used internally by W32Time to store information. These values should not be manually changed at any time. Do not modify any of the settings in this section unless you are familiar with the setting and are certain that the new value will work as expected. The following registry entries are located under:

HKLM\SYSTEM\CurrentControlSet\Services\W32Time

When you create a policy, the settings are configured in the following location, which does not take precedence over the next location:

HKLM\SOFTWARE\Policies\Microsoft\Windows\W32time

The W32time key is created with the policy. When you remove the policy, then this key is also removed.

The other default location:

HKLM\SYSTEM\CurrentControlSet\Services\W32time

Some of the parameters are stored in clock ticks in the registry and some are in seconds. To convert the time from clock ticks to seconds:

- 1 minute = 60 sec
- 1 sec = 1000 ms
- 1 ms = 10,000 clock ticks on a Windows system, as described at [DateTime.Ticks Property](#).

For example, 5 minutes would become $5*60*1000*10000 = 3000000000$ clock ticks.

All versions include Windows 7, Windows 8, Windows 10, Windows Server 2008 , and Windows Server 2008 R2, Windows Server 2012, Windows Server 2012R2, Windows Server 2016. Some entries are only available on newer Windows versions.

HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Parameters

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|----------------------------------|---------|--|
| AllowNonstandardModeCombinations | All | <p>Entry indicates that non-standard mode combinations are allowed in synchronization between peers. The default value for domain members is 1. The default value for stand-alone clients and servers is 1.</p> |
| NtpServer | All | <p>Entry specifies a space-delimited list of peers from which a computer obtains time stamps, consisting of one or more DNS names or IP addresses per line. Each DNS name or IP address listed must be unique. Computers connected to a domain must synchronize with a more reliable time source, such as the official U.S. time clock.</p> <ul style="list-style-type: none"> • 0x01 SpecialInterval • 0x02 UseAsFallbackOnly • 0x04 SymmetricActive - For more information about this mode, see Windows Time Server: 3.3 Modes of Operation. • 0x08 Client <p>There is no default value for this registry entry on domain members. The default value on stand-alone clients and servers is time.windows.com,0x1.</p> <p>Note: For more information on available NTP Servers, see Microsoft Knowledge Base article 262680 - A list of the Simple Network Time Protocol (SNTP) time servers that are available on the Internet</p> |
| ServiceDll | All | <p>Entry is maintained by W32Time. It contains reserved data that is used by the Windows operating system, and any changes to this setting can cause unpredictable results. The default location for this DLL on both domain members and stand-alone clients and servers is %windir%\System32\W32Time.dll.</p> |
| ServiceMain | All | <p>Entry is maintained by W32Time. It contains reserved data that is used by the Windows operating system, and any changes to this setting can cause unpredictable results. The default value on domain members is SvchostEntry_W32Time. The default value on stand-alone clients and servers is SvchostEntry_W32Time. "</p> |

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|----------------|---------|--|
| Type | All | <p>Entry indicates which peers to accept synchronization from:</p> <ul style="list-style-type: none"> • NoSync. The time service does not synchronize with other sources. • NTP. The time service synchronizes from the servers specified in the NtpServer registry entry. • NT5DS. The time service synchronizes from the domain hierarchy. • AllSync. The time service uses all the available synchronization mechanisms. <p>The default value on domain members is NT5DS. The default value on stand-alone clients and servers is NTP.</p> |

HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Config

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|----------------|---------|---|
| AnnounceFlags | All | <p>Entry controls whether this computer is marked as a reliable time server. A computer is not marked as reliable unless it is also marked as a time server.</p> <ul style="list-style-type: none"> - 0x00 Not a time server - 0x01 Always time server - 0x02 Automatic time server - 0x04 Always reliable time server - 0x08 Automatic reliable time server <p>The default value for domain members is 10. The default value for stand-alone clients and servers is 10.</p> |
| EventLogFlags | All | <p>Entry controls the events that the time service logs.</p> <ul style="list-style-type: none"> - Time Jump: 0x1 - Source Change: 0x2 <p>The default value on domain members is 2. The default value on stand-alone clients and servers is 2.</p> |

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|----------------------|---------|--|
| FrequencyCorrectRate | All | <p>Entry controls the rate at which the clock is corrected. If this value is too small, the clock is unstable and overcorrects. If the value is too large, the clock takes a long time to synchronize. The default value on domain members is 4. The default value on stand-alone clients and servers is 4.</p> <p>Note that 0 is an invalid value for the FrequencyCorrectRate registry entry. On Windows Server 2003, Windows Server 2003 R2, Windows Server 2008 , and Windows Server 2008 R2 computers, if the value is set to 0 the Windows Time service will automatically change it to 1.</p> |
| HoldPeriod | All | <p>Entry controls the period of time for which spike detection is disabled in order to bring the local clock into synchronization quickly. A spike is a time sample indicating that time is off a number of seconds, and is usually received after good time samples have been returned consistently. The default value on domain members is 5. The default value on stand-alone clients and servers is 5.</p> |
| LargePhaseOffset | All | <p>Entry specifies that a time offset greater than or equal to this value in 10^{-7} seconds is considered a spike. A network disruption such as a large amount of traffic might cause a spike. A spike will be ignored unless it persists for a long period of time. The default value on domain members is 50000000. The default value on stand-alone clients and servers is 50000000.</p> |
| LastClockRate | All | <p>Entry is maintained by W32Time. It contains reserved data that is used by the Windows operating system, and any changes to this setting can cause unpredictable results. The default value on domain members is 156250. The default value on stand-alone clients and servers is 156250.</p> |
| LocalClockDispersion | All | <p>Entry controls the dispersion (in seconds) that you must assume when the only time source is the built-in CMOS clock. The default value on domain members is 10. The default value on stand-alone clients and servers is 10.</p> |

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|-----------------------|---------|--|
| MaxAllowedPhaseOffset | All | Entry specifies the maximum offset (in seconds) for which W32Time attempts to adjust the computer clock by using the clock rate. When the offset exceeds this rate, W32Time sets the computer clock directly. The default value for domain members is 300. The default value for stand-alone clients and servers is 1. See below for more information. |
| MaxClockRate | All | Entry is maintained by W32Time. It contains reserved data that is used by the Windows operating system, and any changes to this setting can cause unpredictable results. The default value for domain members is 155860. The default value for stand-alone clients and servers is 155860. |
| MaxNegPhaseCorrection | All | Entry specifies the largest negative time correction in seconds that the service makes. If the service determines that a change larger than this is required, it logs an event instead. Special case: 0xFFFFFFFF means always make time correction. The default value for domain members is 0xFFFFFFFF. The default value for stand-alone clients and servers is 54,000 (15 hrs). |
| MaxPollInterval | All | Entry specifies the largest interval, in log2 seconds, allowed for the system polling interval. Note that while a system must poll according to the scheduled interval, a provider can refuse to produce samples when requested to do so. The default value for domain controllers is 10. The default value for domain members is 15. The default value for stand-alone clients and servers is 15. |
| MaxPosPhaseCorrection | All | Entry specifies the largest positive time correction in seconds that the service makes. If the service determines that a change larger than this is required, it logs an event instead. Special case: 0xFFFFFFFF means always make time correction. The default value for domain members is 0xFFFFFFFF. The default value for stand-alone clients and servers is 54,000 (15 hrs). |

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|------------------|---------|--|
| MinClockRate | All | Entry is maintained by W32Time. It contains reserved data that is used by the Windows operating system, and any changes to this setting can cause unpredictable results. The default value for domain members is 155860. The default value for stand-alone clients and servers is 155860. |
| MinPollInterval | All | Entry specifies the smallest interval, in log2 seconds, allowed for the system polling interval. Note that while a system does not request samples more frequently than this, a provider can produce samples at times other than the scheduled interval. The default value for domain controllers is 6. The default value for domain members is 10. The default value for stand-alone clients and servers is 10. |
| PhaseCorrectRate | All | <p>Entry controls the rate at which the phase error is corrected. Specifying a small value corrects the phase error quickly, but might cause the clock to become unstable. If the value is too large, it takes a longer time to correct the phase error.</p> <p>The default value on domain members is 1. The default value on stand-alone clients and servers is 7.</p> <p>Note: 0 is an invalid value for the PhaseCorrectRate registry entry. On Windows Server 2003, Windows Server 2003 R2, Windows Server 2008 , and Windows Server 2008 R2 computers, if the value is set to 0, the Windows Time service automatically changes it to 1.</p> |
| PollAdjustFactor | All | Entry controls the decision to increase or decrease the poll interval for the system. The larger the value, the smaller the amount of error that causes the poll interval to be decreased. The default value on domain members is 5. The default value on stand-alone clients and servers is 5. |
| SpikeWatchPeriod | All | Entry specifies the amount of time that a suspicious offset must persist before it is accepted as correct (in seconds). The default value on domain members is 900. The default value on stand-alone clients and workstations is 900. |

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|---------------------|----------------------------|--|
| TimeJumpAuditOffset | All | <p>An unsigned integer that indicates the time jump audit threshold, in seconds. If the time service adjusts the local clock by setting the clock directly, and the time correction is more than this value, then the time service logs an audit event.</p> |
| UpdateInterval | All | <p>Entry specifies the number of clock ticks between phase correction adjustments. The default value for domain controllers is 100. The default value for domain members is 30,000. The default value for stand-alone clients and servers is 360,000.</p> <p>NOTE: Zero is an invalid value for the UpdateInterval registry entry. On computers running Windows Server 2003, Windows Server 2003 R2, Windows Server 2008 , and Windows Server 2008 R2 , if the value is set to 0 the Windows Time service automatically changes it to 1.</p> <p>The following three registry entries are not a part of the W32Time default configuration but can be added to the registry to obtain increased logging capabilities. The information logged to the System Event log can be modified by changing value for the EventLogFlags setting in the Group Policy Object Editor. By default, the time service creates a log in Event Viewer every time that it switches to a new time source.</p> <p>WARNING: Some of the preset values that are configured in the System Administrative template file (System.adm) for the Group Policy object (GPO) settings are different from the corresponding default registry entries. If you plan to use a GPO to configure any Windows Time setting, be sure that you review Preset values for the Windows Time service Group Policy settings are different from the corresponding Windows Time service registry entries in Windows Server 2003. This issue applies to Windows Server 2008 R2, Windows Server 2008, Windows Server 2003 R2, and Windows Server 2003.</p> |
| UtilizeSslTimeData | Post Windows 10 build 1511 | <p>Entry of 1 indicates that the W32Time will use multiple SSL timestamps to Seed a clock that is grossly inaccurate.</p> |

The following registry entries must be added in order to enable W32Time logging:

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|----------------|---------|--|
| FileLogEntries | All | Entry controls the amount of entries created in the Windows Time log file. The default value is none, which does not log any Windows Time activity. Valid values are 0 to 300. This value does not affect the event log entries normally created by Windows Time |
| FileLogName | All | Entry controls the location and file name of the Windows Time log. The default value is blank, and should not be changed unless FileLogEntries is changed. A valid value is a full path and file name that Windows Time will use to create the log file. This value does not affect the event log entries normally created by Windows Time. |
| FileLogSize | All | Entry controls the circular logging behavior of Windows Time log files. When FileLogEntries and FileLogName are defined, Entry defines the size, in bytes, to allow the log file to reach before overwriting the oldest log entries with new entries. Any positive number is valid, and 3000000 is recommended. This value does not affect the event log entries normally created by Windows Time. |

HKLM\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpClient

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|----------------------------------|---------|--|
| AllowNonstandardModeCombinations | All | Entry indicates that non-standard mode combinations are allowed in synchronization between peers. The default value for domain members is 1. The default value for stand-alone clients and servers is 1. |
| CompatibilityFlags | All | Entry specifies the following compatibility flags and values:

<ul style="list-style-type: none"> - DispersionInvalid: 0x00000001 - IgnoreFutureRefTimeStamp: 0x00000002 - AutodetectWin2K: 0x80000000 - AutodetectWin2KStage2: 0x40000000
The default value for domain members is 0x80000000. The default value for stand-alone clients and servers is 0x80000000. |

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|--------------------|---------|---|
| CrossSiteSyncFlags | All | <p>Entry determines whether the service chooses synchronization partners outside the domain of the computer. The options and values are:</p> <ul style="list-style-type: none"> - None: 0 - PdcOnly: 1 - All: 2 <p>This value is ignored if the NT5DS value is not set. The default value for domain members is 2. The default value for stand-alone clients and servers is 2.</p> |
| DllName | All | <p>Entry specifies the location of the DLL for the time provider.</p> <p>The default location for this DLL on both domain members and stand-alone clients and servers is %windir%\System32\W32Time.dll.</p> |
| Enabled | All | <p>Entry indicates if the NtpClient provider is enabled in the current Time Service.</p> <ul style="list-style-type: none"> • Yes 1 • No 0 <p>The default value on domain members is 1. The default value on stand-alone clients and servers is 1.</p> |
| EventLogFlags | All | <p>Entry specifies the events logged by the Windows Time service.</p> <ul style="list-style-type: none"> • 0x1 reachability changes • 0x2 large sample skew (This is applicable to Windows Server 2003, Windows Server 2003 R2, Windows Server 2008 , and Windows Server 2008 R2 only) <p>The default value on domain members is 0x1. The default value on stand-alone clients and servers is 0x1.</p> |
| InputProvider | All | <p>Entry indicates whether to enable the NtpClient as an InputProvider, which obtains time information from the NtpServer. The NtpServer is a time server that responds to client time requests on the network by returning time samples that are useful for synchronizing the local clock.</p> <ul style="list-style-type: none"> • Yes = 1 • No = 0 <p>Default value for both domain members and stand-alone clients: 1</p> |

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|----------------------------|---------|--|
| LargeSampleSkew | All | Entry specifies the large sample skew for logging in seconds. To comply with Security and Exchange Commission (SEC) specifications, this should be set to three seconds. Events will be logged for this setting only when EventLogFlags is explicitly configured for 0x2 large sample skew. The default value on domain members is 3. The default value on stand-alone clients and servers is 3. |
| ResolvePeerBackOffMaxTimes | All | Entry specifies the maximum number of times to double the wait interval when repeated attempts to locate a peer to synchronize with fail. A value of zero means that the wait interval is always the minimum. The default value on domain members is 7. The default value on stand-alone clients and servers is 7. |
| ResolvePeerBackoffMinutes | All | Entry specifies the initial interval to wait, in minutes, before attempting to locate a peer to synchronize with. The default value on domain members is 15. The default value on stand-alone clients and servers is 15. |
| SpecialPollInterval | All | <p>Entry specifies the special poll interval in seconds for manual peers. When the SpecialInterval 0x1 flag is enabled, W32Time uses this poll interval instead of a poll interval determined by the operating system. The default value on domain members is 3,600. The default value on stand-alone clients and servers is 604,800.</p> <p>New for build 1702, SpecialPollInterval is contained by the MinPollInterval and MaxPollInterval Config registry values.</p> |
| SpecialPollTimeRemaining | All | Entry is maintained by W32Time. It contains reserved data that is used by the Windows operating system. It specifies the time in seconds before W32Time will resynchronize after the computer has restarted. Any changes to this setting can cause unpredictable results. The default value on both domain members and on stand-alone clients and servers is left blank. |

HKLM\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpServer

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|----------------|---------|-------------|
| | | |

| REGISTRY ENTRY | VERSION | DESCRIPTION |
|----------------------------------|---------|---|
| AllowNonstandardModeCombinations | All | Entry indicates that non-standard mode combinations are allowed in synchronization between clients and servers. The default value for domain members is 1. The default value for stand-alone clients and servers is 1. |
| DllName | All | Entry specifies the location of the DLL for the time provider.

The default location for this DLL on both domain members and stand-alone clients and servers is %windir%\System32\W32Time.dll. |
| Enabled | All | Entry indicates if the NtpServer provider is enabled in the current Time Service. <ul style="list-style-type: none"> • Yes 1 • No 0 The default value on domain members is 1. The default value on stand-alone clients and servers is 1. |
| InputProvider | All | Entry indicates whether to enable the NtpClient as an InputProvider, which obtains time information from the NtpServer. The NtpServer is a time server that responds to client time requests on the network by returning time samples that are useful for synchronizing the local clock. <ul style="list-style-type: none"> • Yes = 1 • No = 0 Default value for both domain members and stand-alone clients: 1 |

MaxAllowedPhaseOffset information

In order for W32Time to set the computer clock gradually, the offset must be less than the

MaxAllowedPhaseOffset value and satisfy the following equation at the same time:

```
|CurrentTimeOffset| / (PhaseCorrectRate*UpdateInterval) < SystemClockRate / 2
```

The CurrentTimeOffset is measured in clock ticks, where 1ms = 10,000 clock ticks on a Windows system.

SystemClockRate and PhaseCorrectRate are also measured in clock ticks. To get the SystemClockRate, you can use the following command and convert it from seconds to clock ticks using the formula of seconds*1000*10000:

```
W32tm /query /status /verbose
ClockRate: 0.0156000s
```

SystemClockRate is the rate of the clock on the system. Using 156000 seconds as an example, the SystemClockRate would be = 0.0156000 * 1000 * 10000 = 156000 clock ticks.

MaxAllowedPhaseOffset is also in seconds. To convert it to clock ticks, multiply

MaxAllowedPhaseOffset*1000*10000.

The following two examples show how to apply

Example 1: Time differs by 4 minutes (For example, your time is 11:05 AM and the time sample received from a peer and believed to be correct is 11:09 AM).

```
phasecorrectRate = 1  
  
UpdateInterval = 30000 (clock ticks)  
  
systemclockRate = 156000 (clock ticks)  
  
MaxAllowedPhaseOffset = 10min = 600 seconds = 600*1000\*1000=6000000000 clock ticks  
  
|currentTimeOffset| = 4mins = 4*60\*1000\*1000 = 2400000000 ticks  
  
Is CurrentTimeOffset < MaxAllowedPhaseOffset?  
  
2400000000 < 6000000000 = TRUE
```

AND does it satisfy the above equation?

```
(|CurrentTimeOffset| / (PhaseCorrectRate*UpdateInterval) < SystemClockRate / 2)  
  
Is 2,400,000,000 / (30000*1) < 156000/2  
  
Is 80,000 < 78,000  
  
NO/FALSE
```

Therefore W32tm would set the clock back immediately.

NOTE

In this case, if you want to set the clock back slowly, you would need to adjust the values of PhaseCorrectRate or updateInterval in the registry as well to ensure the equation results in TRUE.

Example 2: Time differs by 3 minutes.

```
phasecorrectRate = 1  
  
UpdateInterval = 30000 (clock ticks)  
  
systemclockRate = 156000 (clock ticks)  
  
MaxAllowedPhaseOffset = 10min = 600 seconds = 600*1000\*1000=6000000000 clock ticks  
  
currentTimeOffset = 3mins = 3*60\*1000\*1000 = 1800000000 clock ticks  
  
Is CurrentTimeOffset < MaxAllowedPhaseOffset?  
  
1800000000 < 6000000000 = TRUE
```

AND does it satisfy the above equation?

```

(|CurrentTimeOffset| / (PhaseCorrectRate*UpdateInterval) < SystemClockRate / 2)

Is 3 mins (1,800,000,000) / (30000*1) < 156000/2

Is 60,000 < 78,000

YES/TRUE

```

In this case the clock will be set back slowly.

Windows Time Service Group Policy Settings

You can configure most W32Time parameters by using the Group Policy Object Editor. This includes configuring a computer to be an NTPServer or NTPClient, configuring the time synchronization mechanism, and configuring a computer to be a reliable time source.

NOTE

Group Policy settings for the Windows Time service can be configured on Windows Server 2003, Windows Server 2003 R2, Windows Server 2008 , and Windows Server 2008 R2 domain controllers and can be applied only to computers running Windows Server 2003, Windows Server 2003 R2, Windows Server 2008 , and Windows Server 2008 R2 .

You can find the Group Policy settings used to configure W32Time in the Group Policy Object Editor snap-in in the following locations:

- Computer Configuration\Administrative Templates\System\Windows Time Service
 - Configure **Global Configuration Settings** here.
- Computer Configuration\Administrative Templates\System\Windows Time Service\Time Providers
 - Configure **Windows NTP Client** settings here.
 - Enable **Windows NTP Client** here.
 - Enable **Windows NTP Server** here.

WARNING

Some of the preset values that are configured in the System Administrative template file (System.adm) for the Group Policy object (GPO) settings are different from the corresponding default registry entries. If you plan to use a GPO to configure any Windows Time setting, be sure that you review [Preset values for the Windows Time service Group Policy settings are different from the corresponding Windows Time service registry entries in Windows Server 2003](#). This issue applies to Windows Server 2008 R2 , Windows Server 2008 , Windows Server 2003 R2, and Windows Server 2003.

The following table lists the global Group Policy settings that are associated with the Windows Time service and the pre-set value associated with each setting. For more information about each setting, see the corresponding registry entries in "[Windows Time Service Registry Entries](#)" earlier in this subject. The following settings are contained in a single GPO called **Global Configuration Settings**.

Global Group Policy Settings Associated with Windows Time

| GROUP POLICY SETTING | PRE-SET VALUE |
|----------------------|---------------|
| AnnounceFlags | 10 |

| GROUP POLICY SETTING | PRE-SET VALUE |
|-----------------------|-------------------|
| EventLogFlags | 2 |
| FrequencyCorrectRate | 4 |
| HoldPeriod | 5 |
| LargePhaseOffset | 1280000 |
| LocalClockDispersion | 10 |
| MaxAllowedPhaseOffset | 300 |
| MaxNegPhaseCorrection | 54,000 (15 hours) |
| MaxPollInterval | 15 |
| MaxPosPhaseCorrection | 54,000 (15 hours) |
| MinPollInterval | 10 |
| PhaseCorrectRate | 7 |
| PollAdjustFactor | 5 |
| SpikeWatchPeriod | 90 |
| UpdateInterval | 100 |

The following table lists the available settings for the **Configure Windows NTP Client GPO** and the pre-set values that are associated with the Windows Time service. For more information about each setting, see the corresponding registry entries in "[Windows Time Service Registry Entries](#)" earlier in this subject.

NTP Client Group Policy Settings Associated with Windows Time

| GROUP POLICY SETTING | DEFAULT VALUE |
|----------------------------|--|
| NtpServer | time.windows.com,0x1 |
| Type | Default options:
- NTP . Use on computers that are not joined to a domain.
- NT5DS . Use on computers that are joined to a domain. |
| CrossSiteSyncFlags | 2 |
| ResolvePeerBackoffMinutes | 15 |
| ResolvePeerBackoffMaxTimes | 7 |
| SpecialPollInterval | 3600 |

| GROUP POLICY SETTING | DEFAULT VALUE |
|----------------------|---------------|
| EventLogFlags | 0 |

Network Ports Used by the Windows Time Service

Windows Time follows the NTP specification, which requires the use of UDP port 123 for all time synchronization communication. This port is reserved by Windows Time and remains reserved at all times. Whenever the computer synchronizes its clock or provides time to another computer, that communication is performed on UDP port 123.

NOTE

If you have a computer with multiple network adapters (also called a multihomed computer), you cannot selectively enable the Windows Time service based on the network adapter.

Related Information

The following resources contain additional information that is relevant to this section.

- RFC 1305 in the IETF RFC Database