**Set up a NAT network**

* 05/02/2016
* 9 minutes to read
  + [[](https://github.com/MicrosoftDocs/Virtualization-Documentation/blob/live/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.md)](https://github.com/MicrosoftDocs/Virtualization-Documentation/blob/live/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.md" \o "10 Contributors)

* + [[](https://github.com/MicrosoftDocs/Virtualization-Documentation/blob/live/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.md)](https://github.com/MicrosoftDocs/Virtualization-Documentation/blob/live/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.md" \o "10 Contributors)

* + [[](https://github.com/MicrosoftDocs/Virtualization-Documentation/blob/live/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.md)](https://github.com/MicrosoftDocs/Virtualization-Documentation/blob/live/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.md" \o "10 Contributors)

* + [[](https://github.com/MicrosoftDocs/Virtualization-Documentation/blob/live/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.md)](https://github.com/MicrosoftDocs/Virtualization-Documentation/blob/live/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.md" \o "10 Contributors)

* + [[](https://github.com/MicrosoftDocs/Virtualization-Documentation/blob/live/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.md)](https://github.com/MicrosoftDocs/Virtualization-Documentation/blob/live/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.md" \o "10 Contributors)

* + [+5](https://github.com/MicrosoftDocs/Virtualization-Documentation/blob/live/virtualization/hyper-v-on-windows/user-guide/setup-nat-network.md" \o "10 Contributors)

Windows 10 Hyper-V allows native network address translation (NAT) for a virtual network.

This guide will walk you through:

* creating a NAT network
* connecting an existing virtual machine to your new network
* confirming that the virtual machine is connected correctly

Requirements:

* Windows 10 Anniversary Update or later
* Hyper-V is enabled (instructions [here](https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v))

**Note:** Currently, you are limited to one NAT network per host. For additional details on the Windows NAT (WinNAT) implementation, capabilities, and limitations, please reference the [WinNAT capabilities and limitations blog](https://techcommunity.microsoft.com/t5/Virtualization/Windows-NAT-WinNAT-Capabilities-and-limitations/ba-p/382303)

**NAT Overview**

NAT gives a virtual machine access to network resources using the host computer's IP address and a port through an internal Hyper-V Virtual Switch.

Network Address Translation (NAT) is a networking mode designed to conserve IP addresses by mapping an external IP address and port to a much larger set of internal IP addresses. Basically, a NAT uses a flow table to route traffic from an external (host) IP Address and port number to the correct internal IP address associated with an endpoint on the network (virtual machine, computer, container, etc.)

Additionally, NAT allows multiple virtual machines to host applications that require identical (internal) communication ports by mapping these to unique external ports.

For all of these reasons, NAT networking is very common for container technology (see [Container Networking](https://docs.microsoft.com/en-us/virtualization/windowscontainers/container-networking/architecture)).

**Create a NAT virtual network**

Let's walk through setting up a new NAT network.

1. Open a PowerShell console as Administrator.
2. Create an internal switch.

PowerShellCopy

New-VMSwitch -SwitchName "SwitchName" -SwitchType Internal

1. Find the interface index of the virtual switch you just created.

You can find the interface index by running Get-NetAdapter

Your output should look something like this:

ConsoleCopy

PS C:\> Get-NetAdapter

Name InterfaceDescription ifIndex Status MacAddress LinkSpeed

---- -------------------- ------- ------ ---------- ---------

vEthernet (intSwitch) Hyper-V Virtual Ethernet Adapter 24 Up 00-15-5D-00-6A-01 10 Gbps

Wi-Fi Marvell AVASTAR Wireless-AC Net... 18 Up 98-5F-D3-34-0C-D3 300 Mbps

Bluetooth Network ... Bluetooth Device (Personal Area... 21 Disconnected 98-5F-D3-34-0C-D4 3 Mbps

The internal switch will have a name like vEthernet (SwitchName) and an Interface Description of Hyper-V Virtual Ethernet Adapter. Take note of its ifIndex to use in the next step.

1. Configure the NAT gateway using [New-NetIPAddress](https://docs.microsoft.com/en-us/powershell/module/nettcpip/New-NetIPAddress).

Here is the generic command:

PowerShellCopy

New-NetIPAddress -IPAddress <NAT Gateway IP> -PrefixLength <NAT Subnet Prefix Length> -InterfaceIndex <ifIndex>

In order to configure the gateway, you'll need a bit of information about your network:

* + **IPAddress** -- NAT Gateway IP specifies the IPv4 or IPv6 address to use as the NAT gateway IP.  
    The generic form will be a.b.c.1 (e.g. 172.16.0.1). While the final position doesn’t have to be .1, it usually is (based on prefix length)

A common gateway IP is 192.168.0.1

* + **PrefixLength** -- NAT Subnet Prefix Length defines the NAT local subnet size (subnet mask). The subnet prefix length will be an integer value between 0 and 32.

0 would map the entire internet, 32 would only allow one mapped IP. Common values range from 24 to 12 depending on how many IPs need to be attached to the NAT.

A common PrefixLength is 24 -- this is a subnet mask of 255.255.255.0

* + **InterfaceIndex** -- ifIndex is the interface index of the virtual switch, which you determined in the previous step.

Run the following to create the NAT Gateway:

PowerShellCopy

New-NetIPAddress -IPAddress 192.168.0.1 -PrefixLength 24 -InterfaceIndex 24

1. Configure the NAT network using [New-NetNat](https://docs.microsoft.com/en-us/powershell/module/netnat/New-NetNat).

Here is the generic command:

PowerShellCopy

New-NetNat -Name <NATOutsideName> -InternalIPInterfaceAddressPrefix <NAT subnet prefix>

In order to configure the gateway, you'll need to provide information about the network and NAT Gateway:

* + **Name** -- NATOutsideName describes the name of the NAT network. You'll use this to remove the NAT network.
  + **InternalIPInterfaceAddressPrefix** -- NAT subnet prefix describes both the NAT Gateway IP prefix from above as well as the NAT Subnet Prefix Length from above.

The generic form will be a.b.c.0/NAT Subnet Prefix Length

From the above, for this example, we'll use 192.168.0.0/24

For our example, run the following to setup the NAT network:

PowerShellCopy

New-NetNat -Name MyNATnetwork -InternalIPInterfaceAddressPrefix 192.168.0.0/24

Congratulations! You now have a virtual NAT network! To add a virtual machine, to the NAT network follow [these instructions](https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/user-guide/setup-nat-network#connect-a-virtual-machine).

**Connect a virtual machine**

To connect a virtual machine to your new NAT network, connect the internal switch you created in the first step of the [NAT Network Setup](https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/user-guide/setup-nat-network#create-a-nat-virtual-network) section to your virtual machine using the VM Settings menu.

Since WinNAT by itself does not allocate and assign IP addresses to an endpoint (e.g. VM), you will need to do this manually from within the VM itself - i.e. set IP address within range of NAT internal prefix, set default gateway IP address, set DNS server information. The only caveat to this is when the endpoint is attached to a container. In this case, the Host Network Service (HNS) allocates and uses the Host Compute Service (HCS) to assign the IP address, gateway IP, and DNS info to the container directly.

**Configuration Example: Attaching VMs and Containers to a NAT network**

*If you need to attach multiple VMs and containers to a single NAT, you will need to ensure that the NAT internal subnet prefix is large enough to encompass the IP ranges being assigned by different applications or services (e.g. Docker for Windows and Windows Container – HNS). This will require either application-level assignment of IPs and network configuration or manual configuration which must be done by an admin and guaranteed not to re-use existing IP assignments on the same host.*

**Docker for Windows (Linux VM) and Windows Containers**

The solution below will allow both Docker for Windows (Linux VM running Linux containers) and Windows Containers to share the same WinNAT instance using separate internal vSwitches. Connectivity between both Linux and Windows containers will work.

User has connected VMs to a NAT network through an internal vSwitch named “VMNAT” and now wants to install Windows Container feature with docker engine

Copy

PS C:\> Get-NetNat “VMNAT”| Remove-NetNat (this will remove the NAT but keep the internal vSwitch).

Install Windows Container Feature

DO NOT START Docker Service (daemon)

Edit the arguments passed to the docker daemon (dockerd) by adding –fixed-cidr=<container prefix> parameter. This tells docker to create a default nat network with the IP subnet <container prefix> (e.g. 192.168.1.0/24) so that HNS can allocate IPs from this prefix.

PS C:\> Start-Service Docker; Stop-Service Docker

PS C:\> Get-NetNat | Remove-NetNAT (again, this will remove the NAT but keep the internal vSwitch)

PS C:\> New-NetNat -Name SharedNAT -InternalIPInterfaceAddressPrefix <shared prefix>

PS C:\> Start-Service docker

Docker/HNS will assign IPs to Windows containers and Admin will assign IPs to VMs from the difference set of the two.

User has installed Windows Container feature with docker engine running and now wants to connect VMs to the NAT network

Copy

PS C:\> Stop-Service docker

PS C:\> Get-ContainerNetwork | Remove-ContainerNetwork -force

PS C:\> Get-NetNat | Remove-NetNat (this will remove the NAT but keep the internal vSwitch)

Edit the arguments passed to the docker daemon (dockerd) by adding -b “none” option to the end of docker daemon (dockerd) command to tell docker not to create a default NAT network.

PS C:\> New-ContainerNetwork –name nat –Mode NAT –subnetprefix <container prefix> (create a new NAT and internal vSwitch – HNS will allocate IPs to container endpoints attached to this network from the <container prefix>)

PS C:\> Get-Netnat | Remove-NetNAT (again, this will remove the NAT but keep the internal vSwitch)

PS C:\> New-NetNat -Name SharedNAT -InternalIPInterfaceAddressPrefix <shared prefix>

PS C:\> New-VirtualSwitch -Type internal (attach VMs to this new vSwitch)

PS C:\> Start-Service docker

Docker/HNS will assign IPs to Windows containers and Admin will assign IPs to VMs from the difference set of the two.

In the end, you should have two internal VM switches and one NetNat shared between them.

**Multiple Applications using the same NAT**

Some scenarios require multiple applications or services to use the same NAT. In this case, the following workflow must be followed so that multiple applications / services can use a larger NAT internal subnet prefix

***We will detail the Docker 4 Windows - Docker Beta - Linux VM co-existing with the Windows Container feature on the same host as an example. This workflow is subject to change***

1. C:> net stop docker
2. Stop Docker4Windows MobyLinux VM
3. PS C:> Get-ContainerNetwork | Remove-ContainerNetwork -force
4. PS C:> Get-NetNat | Remove-NetNat  
   *Removes any previously existing container networks (i.e. deletes vSwitch, deletes NetNat, cleans up)*
5. New-ContainerNetwork -Name nat -Mode NAT –subnetprefix 10.0.76.0/24 (this subnet will be used for Windows containers feature) *Creates internal vSwitch named nat*  
   *Creates NAT network named “nat” with IP prefix 10.0.76.0/24*
6. Remove-NetNAT  
   *Removes both DockerNAT and nat NAT networks (keeps internal vSwitches)*
7. New-NetNat -Name DockerNAT -InternalIPInterfaceAddressPrefix 10.0.0.0/17 (this will create a larger NAT network for both D4W and containers to share)  
   *Creates NAT network named DockerNAT with larger prefix 10.0.0.0/17*
8. Run Docker4Windows (MobyLinux.ps1)  
   *Creates internal vSwitch DockerNAT*  
   *Creates NAT network named “DockerNAT” with IP prefix 10.0.75.0/24*
9. Net start docker  
   *Docker will use the user-defined NAT network as the default to connect Windows containers*

In the end, you should have two internal vSwitches – one named DockerNAT and the other named nat. You will only have one NAT network (10.0.0.0/17) confirmed by running Get-NetNat. IP addresses for Windows containers will be assigned by the Windows Host Network Service (HNS) from the 10.0.76.0/24 subnet. Based on the existing MobyLinux.ps1 script, IP addresses for Docker 4 Windows will be assigned from the 10.0.75.0/24 subnet.

**Troubleshooting**

**Multiple NAT networks are not supported**

This guide assumes that there are no other NATs on the host. However, applications or services will require the use of a NAT and may create one as part of setup. Since Windows (WinNAT) only supports one internal NAT subnet prefix, trying to create multiple NATs will place the system into an unknown state.

To see if this may be the problem, make sure you only have one NAT:

PowerShellCopy

Get-NetNat

If a NAT already exists, delete it

PowerShellCopy

Get-NetNat | Remove-NetNat

Make sure you only have one “internal” vmSwitch for the application or feature (e.g. Windows containers). Record the name of the vSwitch

PowerShellCopy

Get-VMSwitch

Check to see if there are private IP addresses (e.g. NAT default Gateway IP Address – usually *x*.*y*.*z*.1) from the old NAT still assigned to an adapter

PowerShellCopy

Get-NetIPAddress -InterfaceAlias "vEthernet (<name of vSwitch>)"

If an old private IP address is in use, please delete it

PowerShellCopy

Remove-NetIPAddress -InterfaceAlias "vEthernet (<name of vSwitch>)" -IPAddress <IPAddress>

**Removing Multiple NATs**  
We have seen reports of multiple NAT networks created inadvertently. This is due to a bug in recent builds (including Windows Server 2016 Technical Preview 5 and Windows 10 Insider Preview builds). If you see multiple NAT networks, after running docker network ls or Get-ContainerNetwork, please perform the following from an elevated PowerShell:

PowerShellCopy

$keys = Get-ChildItem "HKLM:\SYSTEM\CurrentControlSet\Services\vmsmp\parameters\SwitchList"

foreach($key in $keys)

{

if ($key.GetValue("FriendlyName") -eq 'nat')

{

$newKeyPath = $KeyPath+"\"+$key.PSChildName

Remove-Item -Path $newKeyPath -Recurse

}

}

Remove-NetNat -Confirm:$false

Get-ContainerNetwork | Remove-ContainerNetwork

Get-VmSwitch -Name nat | Remove-VmSwitch # failure is expected

Stop-Service docker

Set-Service docker -StartupType Disabled

Reboot the operating system prior executing the subsequent commands (Restart-Computer)

PowerShellCopy

Get-NetNat | Remove-NetNat

Set-Service docker -StartupType Automatic

Start-Service docker

See this [setup guide for multiple applications using the same NAT](https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/user-guide/setup-nat-network#multiple-applications-using-the-same-nat) to rebuild your NAT environment, if necessary.

**References**

Read more about [NAT networks](https://en.wikipedia.org/wiki/Network_address_translation)