# Overview

All virtualization products provide methods for the virtual machines to communicate with the host machine, other virtual machines, and the external network. Hyper-V does this using virtual switches. In this exercise, you will create and configure virtual switches to allow communication between different parts of your virtual infrastructure.

# Objectives

In this Guided Practice, you will create and test functionality of Hyper-V switches and build the network in Hyper-V.

* Manage VM resources.
* Create and manage switches in Hyper-V.
* Connect VMs to switches in Hyper-V.

## Skills Reviewed

* Configuring IP addressing
* Troubleshooting network connectivity

## New Skills

* Create virtual switches in Hyper-V.
* Configure virtual machine networking in Hyper-V.

# Initial Conditions

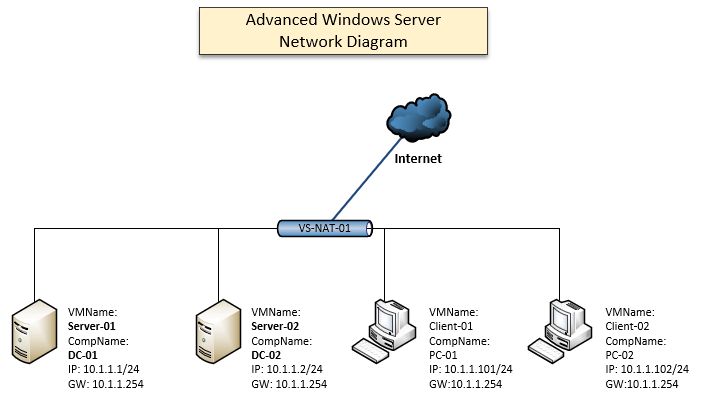
* Server-01, Server-02, Client-01, and Client-01 are running in Hyper-V. The network adapters on these systems are not connected.
* No switches have been manually created. One switch may have been created during Hyper-V role addition.

# Final Conditions

* The external virtual switch VS-EXT-01 has been created.
* The internal virtual switch VS-PRI-01 through VS-PRI-04 have been created.
* IP Addresses have been configured on all network adapters on the virtual machines.

# Lab Instructions

## Network diagram

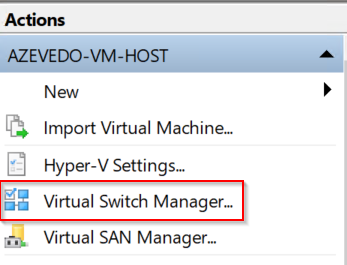
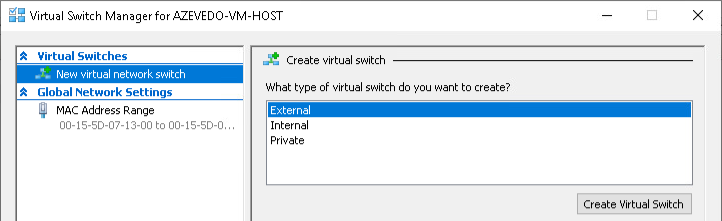
In this exercise, you will be configuring the virtual switches and the network adapter configurations shown in the diagram below.

## Verifying Network Setup

Before configuring your network, you should know the status of the existing network. This includes the existing switches and how the VMs are connected to the existing switches.

### Viewing virtual switches

**To view the existing switches in the Hyper-V management console, perform the following:**

1. Login to the **<lastname>-VM-Host** virtual machine.
2. Open the **Hyper-V Management** console.
3. In the **Actions** pane select the **Virtual Switch Manager** as shown in the figure.
4. You should see something like the screenshot below
5. Any existing switches will be shown in the Virtual Switches pane as shown. You may have a switch that was installed during installation. You will delete this switch in a later step.

**To view the virtual switches using PowerShell, perform the following:**

1. Open a PowerShell session and type the following command:

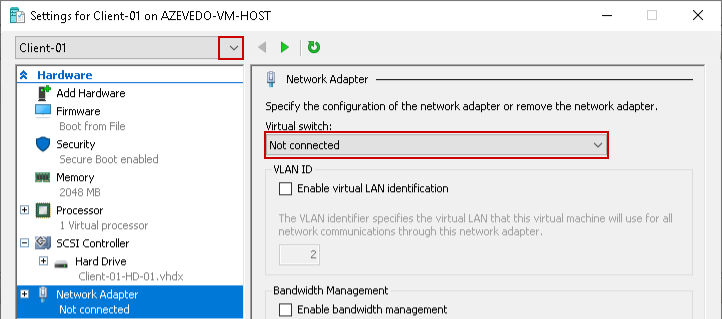
Get-VMSwitch

1. You should get no output.

## Check Network connections

In this step, you will check the status of the network adapters in your virtual machines.

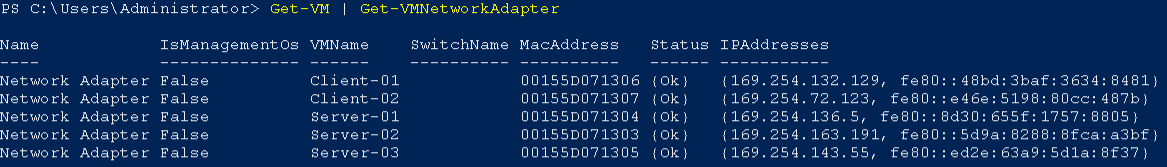
**To view the status of the network adapter in Hyper-V Management console, perform the following:**

1. Open the **settings** for the **Client-01** virtual machine.
2. Select the **Network Adapter** and verify that **Not connected** is shown in the **Virtual Switch:** dropdown box as shown below.
3. Note the little down arrow next to the **Client-01**. You can use this to move to the settings of the other virtual machines and verify their settings.

**To view the status of the network adapter using PowerShell, perform the following:**

1. Open a **PowerShell (Admin)** session and type the following command.

Get-VM | Get-VMNetworkAdapter

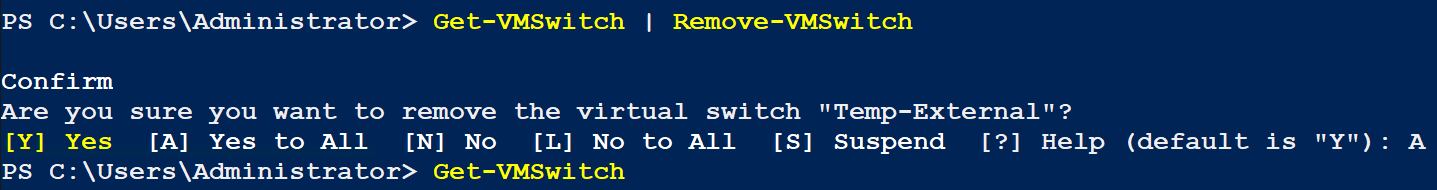
1. You should see output like the figure below.
2. Notice that the adapters are not connected to a switch and it shows the IP addresses. This command can be very useful in troubleshooting network connectivity issues.

## Removing existing switches

Before you start, you will need to remove any switches that exist. In this case, you will use the **Get-VMSwitch** and pipe it to the **Remove-VMSwitch** cmdlet to quickly remove all the switches. In real life you would just use the **Remove-VMSwitch** with the name of the specific switch to remove just that switch.

1. To remove all the virtual switches on the Lab-VM-Host virtual machine, type the following **command** in PowerShell:

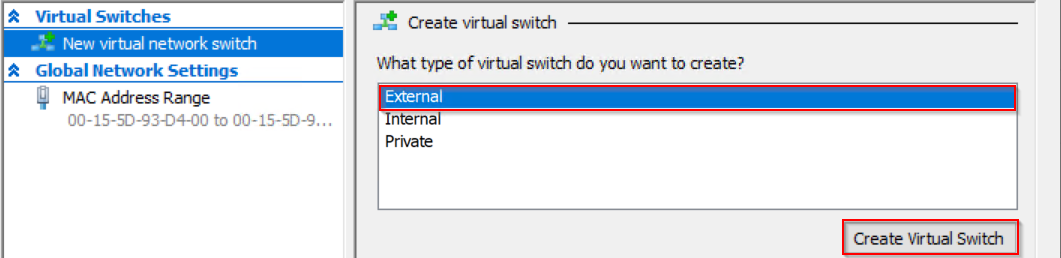
Get-VMSwitch | Remove-VMSwitch

1. You should verify that all the switches have been removed.
2. You should see the output shown below if you have a configured switch.

## Creating an external switch

An external switch is a switch that is bridged to a physical adapter. This will normally give any device connected to the switch access to the same physical network to which the adapter is attached.

To create an external virtual switch in Hyper-V, perform the following:

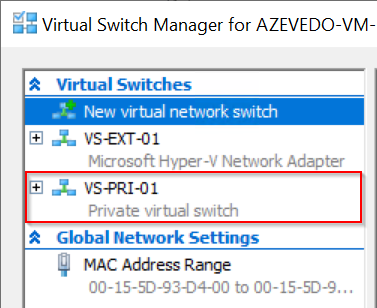
1. Open the **Hyper-V Management** console.
2. Open the **Virtual Switch Manager.**
3. Select **External** for the type of virtual switch to create as shown in the figure.
4. Click the **Create Virtual Switch** button.
5. In the **Virtual Switch Properties** page, type **VS-EXT-01** and then click the **Apply** button. You will get a message about disrupting network communications, just close the dialog box.
6. Click **OK** to close the **Virtual Switch Manager**.

## Creating Private Virtual Switches

Most communications between the virtual machines will take place using internal switches. There are two types of internal switches in Hyper-V. The **Internal** switch is used when you need communication between the host (hypervisor) and the virtual machines connected to that switch. It accomplishes this by creating an additional adapter on the host machine that is connected to the internal switch. The **Private** switch is used when you want only the virtual machines connected to the switch to communicate with each other.

In your network, only communication between virtual machine is necessary, so all your switches will be **Private**.

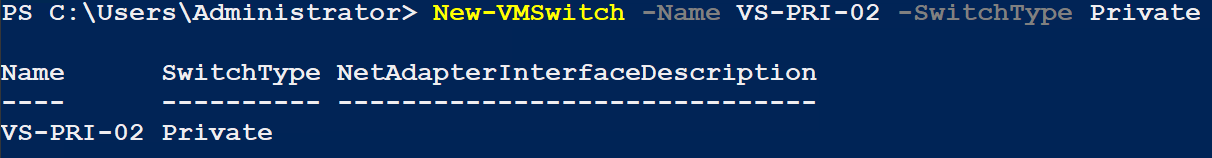
**To create a Private virtual switch using Hyper-V Management console, perform the following:**

1. Open **Hyper-V Management** console.
2. Open the **Virtual Switch Manager**.
3. In the “**What type of virtual switch do you want to create**?” section choose **Private** and click the **Create Virtual Switch** button.
4. On the **Virtual Switch Properties** page, type **VS-PRI-01** in the **Name**: textbox and then click the **Apply** button.
5. Verify that the switch shows up in the **Virtual Switches** pane as shown in the figure.

**To create a Private virtual switch using PowerShell, perform the following:**

1. Opena **PowerShell (Admin)** session.
2. Typethe following command:

New-VMSwitch -Name VS-PRI-02 -SwitchType Private

1. You should see the output below.

## Creating An Internal Virtual Switch

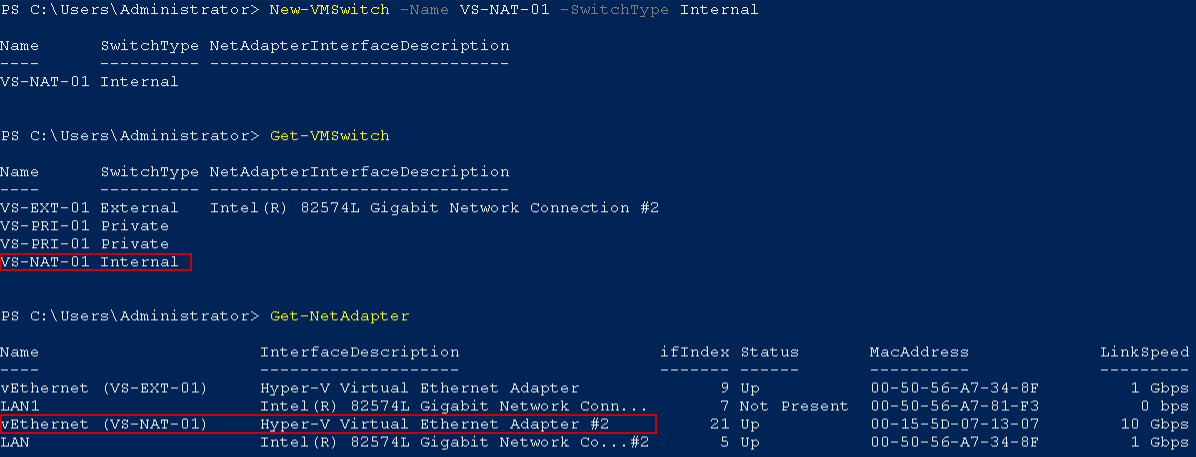
**Internal** virtual switches are created when you wish for the host machine to communicate with the virtual machines connected to the switch. It accomplishes this by creating a network adapter on the host that is connected to the internal virtual switch.

One particular use for this is when using the host machine as a NAT router. This allows the private IP addresses used in your internal networks to be replaced by the public address of the host machine and routed to the Internet, giving your virtual machines access to the Internet.

**To create an Internal virtual switch using PowerShell, perform the following:**

1. Opena **PowerShell (Admin)** session.
2. Typethe following command:

New-VMSwitch -Name VS-NAT-01 -SwitchType Internal

1. The output below shows the creation of the switch, verifying the switch was created, and the network adapter that was created to allow the host machine to connect to the switch.

## Configuring NAT in Hyper-V

In this section, you will create the necessary components that will allow the host machine act as a NAT router using the previously created internal switch and its associated adapter.

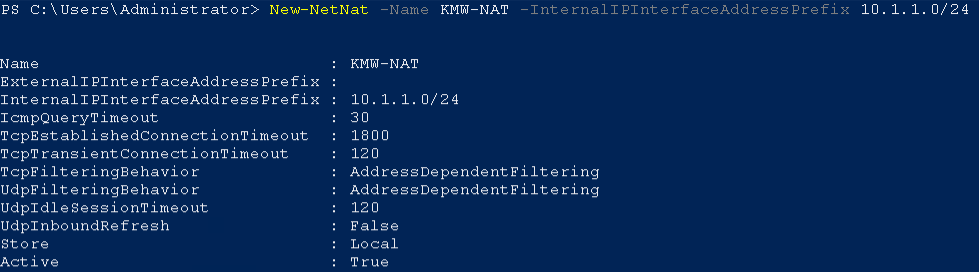
To configure a Hyper-V host to perform NAT routing, you need to create a NetNAT object, assign it an address range to be used by the VMs, and configure the associated adapter with an address within the address range that will be the router IP.

### Creating a NetNAT object

To create a NetNat object, perform the following:

1. Openan elevated PowerShell session.
2. Typethe following command:

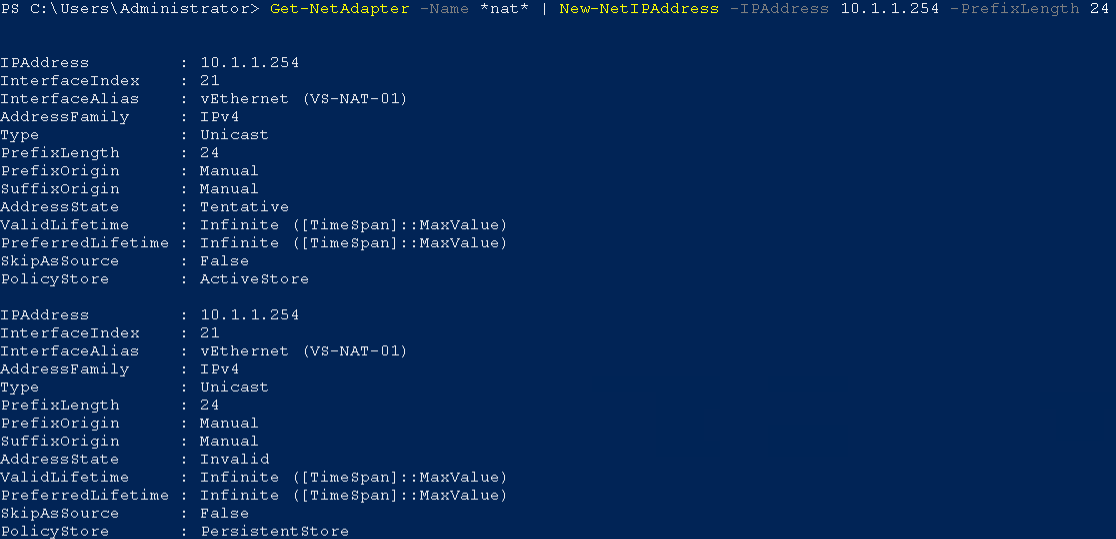
New-NetNat -Name KMW-NAT -InternalIPInterfaceAddressPrefix 10.1.1.0/24

1. You should see the output shown below.

### Configuring the Network Adapter for use as a Router

The last step in configuring the router is to assign an address to the associated network adapter. In this case, you will assign the 10.1.1.254 address to the adapter using the following command:

Get-NetAdapter -Name \*NAT\* | New-NetIPAddress -IPAddress 10.1.1.254 -PrefixLength 24

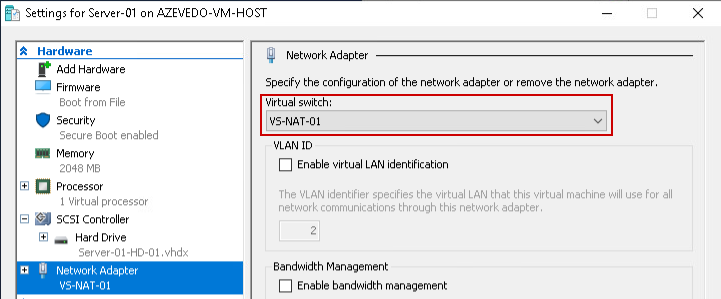
You should see the output shown below.

## Connecting a Network Adapter to a virtual switch

Now that the virtual switches have been configured, you can now connect the network adapters on the virtual machines to the appropriate switches.

To connect a network adapter to a switch using the Hyper-V Management console, perform the following:

1. Open **Hyper-V Management** console.
2. Open the settingsfor the **Server-01** virtual machine.
3. Selectthe **Network Adapter** in the **Hardware** pane and select the **VS-NAT-01** switch in the **Network Adapter** configuration page and then click the **Apply** button.

The page should look like the figure below.

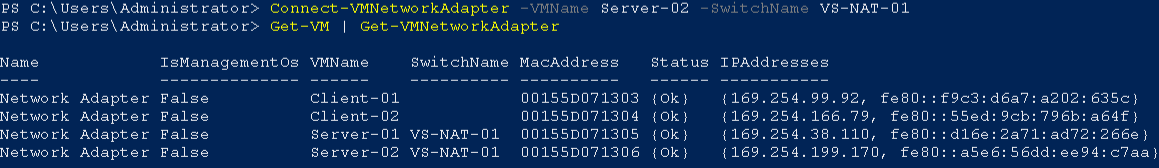
To connect a network adapter to a virtual switch using PowerShell, perform the following:

1. Openan elevated PowerShell session.
2. Typethe following command:

Connect-VMNetworkAdapter -VMName Server-02 -SwitchName VS-NAT-01

1. There will be no output. Use the following command to verify the adapter is connected to the correct switch.

Get-VM | Get-VMNetworkAdapter

1. You should see the output below.

## Connecting the Remaining Adapters

Use either of the methods above to connect the virtualnetworkadapters shown in the table below.

|  |  |
| --- | --- |
| Virtual Machine | Switch |
| Client-01 | VS-NAT-01 |
| Client-02 | VS-NAT-01 |

## Configuring TCP/IP Settings

In this step, you will rename your network adapters and configure the IP addresses to match the diagram. To do this you will need to identify which virtual switch each adapter is connected to, rename the adapter, and assign it the appropriate address.

### Renaming Network Adapters in PowerShell

To do this on **Server-01**, perform the following:

1. Login to the virtual machine using the **Administrator** account.
2. Open **PowerShell**.
3. Change the name of the adapter to **LAN**, using one of the following commands:

Get-NetAdapter (use the Name of the adapter as the InterfraceAlias in the next command)

Rename-NetAdapter -InterfaceAlias “Ethernet 2” -NewName LAN

**or**

Get-NetAdapter | Rename-NetAdapter -NewName LAN

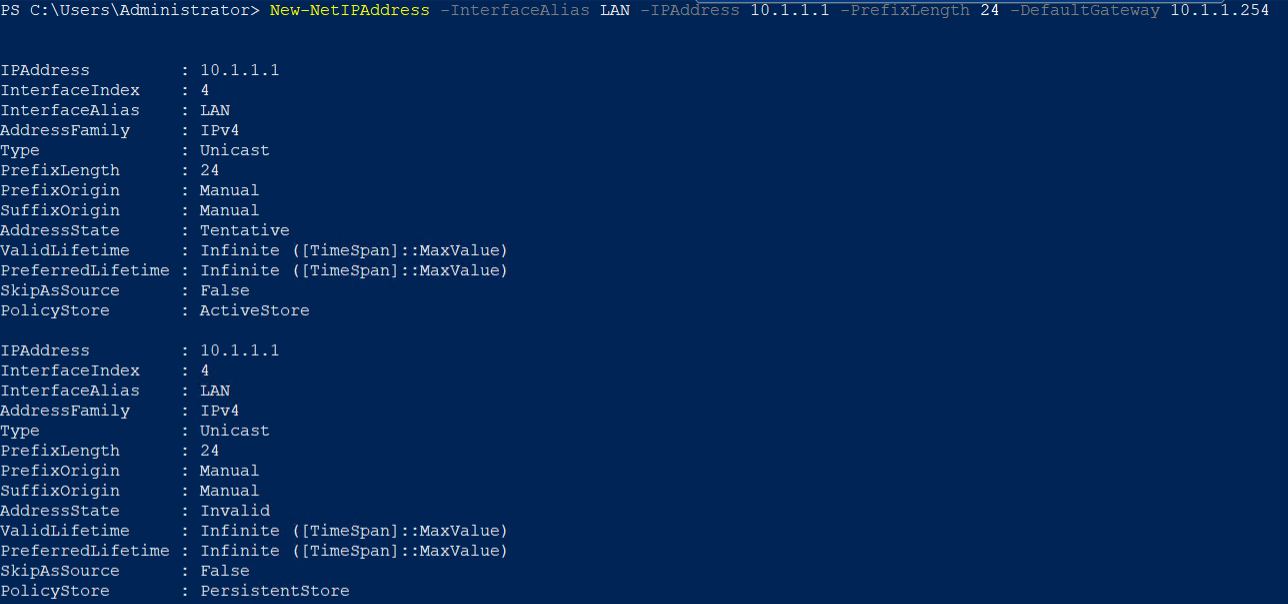
1. Repeat the process above to rename the adapters on the remaining virtual machines to **LAN**.

### Configuring TCP/IP Settings Using PowerShell

To configure the IP address of the LAN adapter using PowerShell, perform the following

1. Login to the **Server-01** virtual machine
2. Open **PowerShell**
3. In PowerShell type the following command:

New-NetIPAddress -InterfaceAlias LAN -IPAddress 10.1.1.1 -PrefixLength 24 -DefaultGateway 10.1.1.254

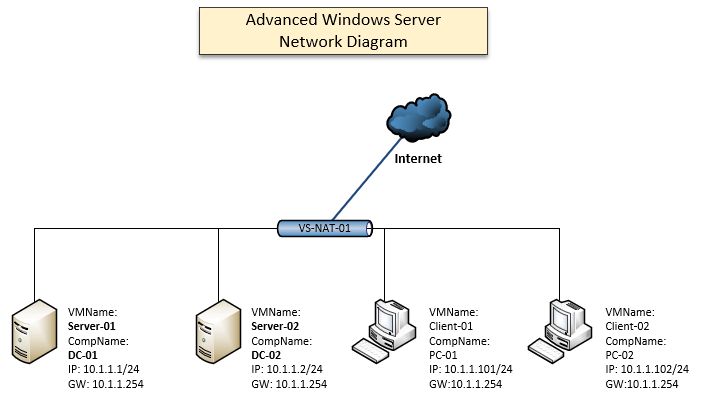
1. You should see the output below.
2. You should now be able to ping the Internet.
3. Test this by using ping command.

ping 1.1.1.1

1. To configure your DNS client using PowerShell, type the following command:

Set-DNSClientServerAddress -InterfaceAlias LAN -ServerAddresses 1.1.1.1

1. Verify your **TCP/IP** configuration using **ipconfig** command.
2. You should now be able to ping Internet sites using their DNS names.
3. Repeat the steps above to configure the **TCP/IP settings** on the remainingvirtualmachines using the IP addresses on the network diagram slmgr /skms 10.as a guide.



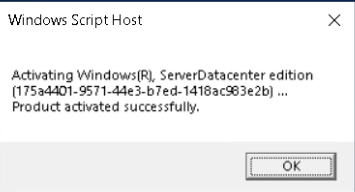
### ACTIVATE VMS

1. The KMS server for ECPI is located at **10.10.6.20**. In **Windows PowerShell (admin)**, issue the following commands to activate the Windows server and client systems.

slmgr /skms 10.10.6.20

slmgr /ato

The expected response is **Product activated successfully**.

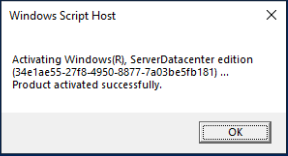


1. If the response from **slmgr /ato** is ***The activation server determined that the specified product key could not be used***, issue the following commands to add the correct product key to the VM so that it will activate.

slmgr /ipk WMDGN-G9PQG-XVVXX-R3X43-63DFG (Server)

slmgr /ipk 2WH4N-8QGBV-H22JP-CT43Q-MDWWJ (Windows 10)

slmgr /ato



# Submission Requirements

1. **Download** the **grading** **script** from the assignment page to the **C:\Scripts** folder.
2. Check your lab by running the following command:

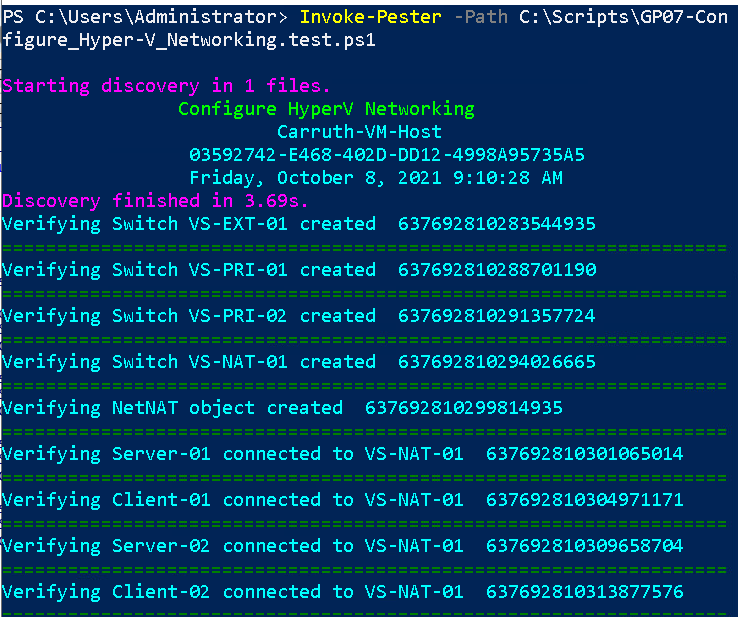
Invoke-Pester -Path C:\Scripts\GP07-Configure\_Hyper-V\_Networking.test.ps1

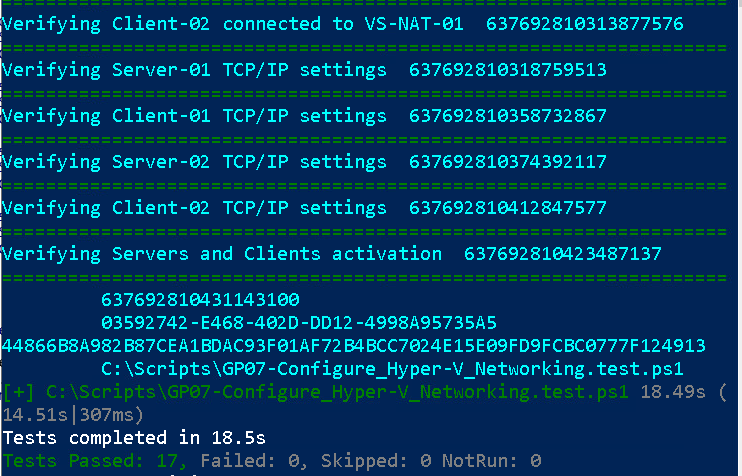
**Note**: You will see a security warning when running the script. Enter **R** to run the script.

If you want to see more detail, add **-Output Detailed** to the command. This may assist you with troubleshooting

Invoke-Pester -Path C:\Scripts\GP07-Configure\_Hyper-V\_Networking.test.ps1 -Output Detailed

1. You should not see any red in the output. Red in the PowerShell way of telling you that an error condition exists. Most of the time, the output will tell you what is wrong. If it is not obvious, contact your teacher and ask for assistance. You will be learning PowerShell during this term. **Correct** any **errors** you may have and run the script until all the output has no red. You should see the output like the images below.





1. Capture a snippet that shows the PowerShell Command and all its output. If you must use more than one snippet to capture the output, you must have at least **one line of overlap** in the snippets. The text in the snippets **must be legible** when pasted into the Word document. Paste the snippet(s) into a **new** **Word** **document**
2. **Fill** **in** the **information** in the following table. Copy the following table into the **Word** **document** and fill in the information about all the **new** commands used in this lab (the example provided is not a new command and should be deleted):

|  |  |  |
| --- | --- | --- |
| PowerShell Commands | | |
| Command | Example | Description |
| *Get-Childitem* | *Get-Childitem -Path C:\* | *Displays the files in the C:\ directory* |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Upload** the **document** in the submission area of the assignment.