# Overview

One of the greatest benefits of virtualization is the ability to rapidly deploy virtual machines using templates or master images. In this exercise, you will use the previously created base images and differencing disk to rapidly deploy virtual machines using the GUI and command line.

# Objectives

In this Guided Practice, you will create a VM in the Hyper-V management console and PowerShell using an existing differencing disk.

1.1 Define or explain common terms and concepts related to virtualization storage and networking.

1.2 Create VM using differencing disk.

## Skills Reviewed

* Connect to a VM.
* Complete setup of a computer system after imaging.
* Rename a computer.

## New Skills

* Creating a VM in Hyper-V with a differencing disk using the Hyper-V management console.
* Creating a VM in Hyper-V using PowerShell.
* Enabling PowerShell Remoting in Windows Server Core and Windows 10.
* Using PowerShell Direct to configure a virtual machine.

## References

* PowerShell Direct - <https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/user-guide/powershell-direct>

# Initial Conditions

* Hyper-V running and Hyper-V manager open.
* Differencing disk has been created.

# Final Conditions

* The following virtual machines have been created:
  + Server-01 with Windows Server Datacenter (Desktop Experience)
  + Server-02 with Windows Server Datacenter
  + Client-01 with Windows 10 Professional or Education
  + Client-02 with Windows 10 Professional or Education
* All computers have been renamed in accordance with the course naming standard.

# nstructions

In this step you will create a virtual machine using the differencing disks that you created in a previous exercise. The procedures below will guide you through this procedure using the GUI and PowerShell.

## Creating a VM using an Existing Disk With Hyper-V Manager

To create a virtual machine in Hyper-V management console, perform the following:

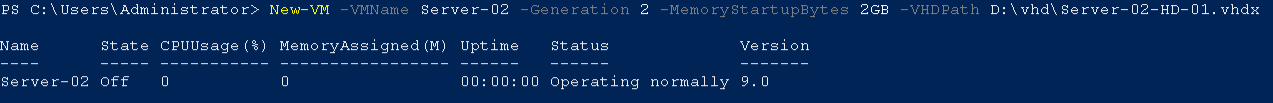
1. Login to the **LastName-VM-Host** virtual machine in VCastle.
2. Open the **Hyper-V Management** console.
3. In the **Actions** **pane**, click **New🡪 New Virtual Machine…** to start the **New Virtual Machine Wizard**.
4. On the **Before You Begin** page, click **Next**.
5. On the **Specify Name and Location** page, in the **Name**: textbox enter the name **Server-01** and then click **Next**.
6. On the **Specify Generation** page, select **Generation 2** and then click **Next**.
7. On the **Assign Memory** page, in the **Startup memory**: text box, enter **2048,** check the **Use** **Dynamic Memory** **for this virtual machine** check box and then click **Next**.
8. On the **Configure Networking** page, in the **Connection:** dropdown menu, select **Not** **Connected** and then click **Next**.
9. On the **Connect Virtual Hard Disk** page, select **Use an existing virtual hard disk** and in the Location: box browse and select the **differencing disk** (**Server-01-HD-01**) and then click **Next**.
10. On the **Summary page**, review the data in the Description textbox and then click **Finish**.

## Creating a VM using an Existing Disk Using powerShell

To create a virtual machine using PowerShell, perform the following:

1. Opena **PowerShell (Admin)** session.
2. To create **Server-02** VM, enterthe following command:

New-VM -VMName Server-02 -Generation 2 -MemoryStartupBytes 2GB -VHDPath D:\VHD\Server-02-HD-01.vhdx

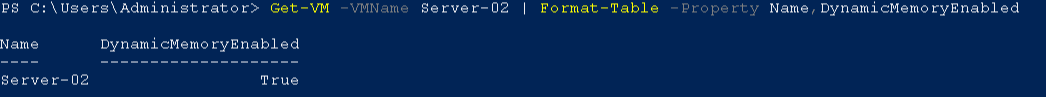
* 1. You should see the output shown below.

1. To configure the machine to use Dynamic Memory, enter the following command:

Set-VM -VMName Server-02 -DynamicMemory

1. The command above provides no output, so to verify the configuration, type the following:

Get-VM -VMName Server-02 | Format-Table -Property Name,DynamicMemoryEnabled

1. You should see the output shown below.

## Creating the Remaining Virtual Machines

Use either of the methods above, to create the remaining virtual machines shown in the table below.

|  |  |
| --- | --- |
| VM Name | VHD |
| Client-01 | Client-01-HD-01.vhdx |
| Client-02 | Client-02-HD-01.vhdx |

## Configuring Computer Names using PowerShell Direct

PowerShell Direct allows you to run commands from the host directly on virtual machines without going through the network.

1. Since sysprep was run on the machines, you will need to connect to the console and go through the initial setup. Once this has done you can proceed to the next step. On the Server virtual machines, set the **Administrator** password to **Password1**. You’ll need to start and log onto each Server virtual machine to set the password.

To use PowerShell Direct to configure the computer name for the **Server-01** virtual machine:

1. From the **VM-Host**, open a **PowerShell** session on the **host** machine and change the computer name of **Server-01** to **DC-01**.

If the VM isn’t started, you’ll need to do that first:

Start-VM -VMName Server-01

Then:

Invoke-Command -VMName Server-01 -ScriptBlock {Rename-Computer -NewName DC-01 -Restart } -Credential Administrator

1. Change the computer name of **Server-02** to **DC-02**

**Note**: For **Client-01** and **Client-02**, use the **student** user account as the administrative account.

1. Change the computer name of **Client-01** to **PC-01**
2. Change the computer name of **Client-02** to **PC-02**

In all cases, log onto each computer and verify the computer name.

1. You can verify that the computer name was changed by using PowerShell Direct to run a command on the virtual machine.

# 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:

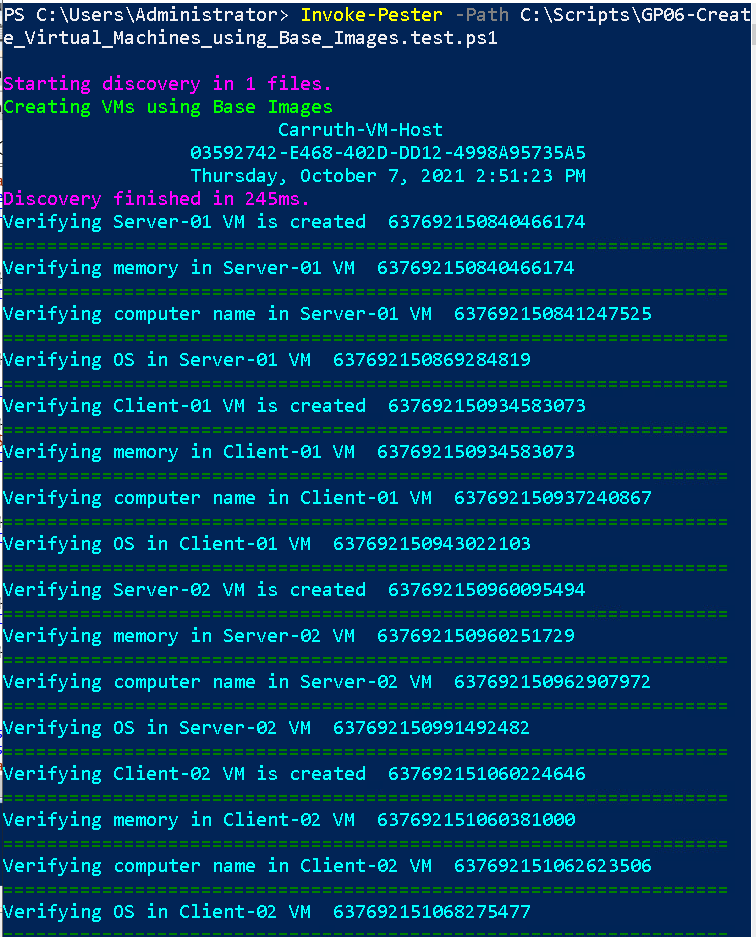
Invoke-Pester -Path C:\Scripts\GP06-Creating\_Virtual\_ Machines\_using\_Base\_Images.test.ps1

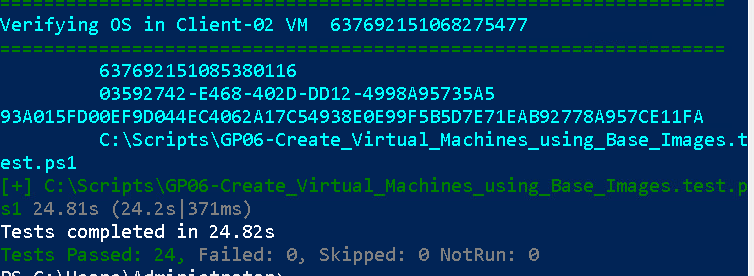
**Note**: You will see a security warning when running the script. Enter **R** to run the script.

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

Invoke-Pester -Path C:\Scripts\GP06-Creating\_Virtual\_ Machines\_using\_Base\_Images.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.