# Practice M2: Basic Services and Components

Note: All tasks are executed on a **Windows Server 2019 Desktop Experience** VM with one LAN adapter and three disks (one for the OS and two empty – for example 10GB and 50GB, added after the OS installation). An additional machine can be used to test the network connectivity between them.

**WARNING**: Please be very careful where you do execute commands, as some of them could cause data loss if executed not in the VM, but on the host.

# Part 0: PowerShell Tips and tricks

In order to get all commands related to a specific module, we can execute:

```
PS C:\> Get-Command -Module ServerManager
```

```
PS C:\> Get-Command -Module PackageManagement
```

Should we want to see the list of all aliases, we can execute:

```
PS C:\> Get-Alias
```

Sometimes the output returned by a command doesn't fit on the screen. We can force the information to be shown one screen at a time:

```
PS C:\> Get-Alias | more
```

Usually, we would like to filter the information returned based on a criterion

```
PS C:\> Get-Alias | Where-Object -Property Source -Like ServerManager
```

Alternatively, we can shorten the above command to

```
PS C:\> Get-Alias | where -Property Source -Like ServerManager
```

We can modify how the information is returned. For example, we can format it as a list

```
PS C:\> Get-Alias Add-WindowsFeature | Format-List *
```

Again, we can have a little bit shorter version

```
PS C:\> Get-Alias Add-WindowsFeature | fl *
```

Please note that **PowerShell** is case **insensitive** for the most part (for command names, options, component names, etc.) but it treats some symbols that may look similar in a different way. For example, the dash symbol (-). Some word-processing solutions may turn it into similarly looking (–) but unacceptable to PowerShell symbol. Be cautious when you do copy-paste 😉

# Part 1: Network and Firewall

## Network Management (GUI)

Graphically, network related tasks can be executed via the **Server Manager** or the **ncpa.cpl** control panel applet

## Network Management (PowerShell)

List network adapters

```
PS C:\> Get-NetAdapter
```

Rename network adapter *(the name of yours may be different)*

```
PS C:\> Rename-NetAdapter -Name "Ethernet 2" -NewName "LAN-Internal"
```

Disable network adapter

```
PS C:\> Disable-NetAdapter -Name "Lan-Internal"
```

Enable network adapter

```
PS C:\> Enable-NetAdapter -Name "lan-internal"
```

Disable and enable network adapter

```
PS C:\> Restart-NetAdapter "LAN-INTERNAL"
```

As you can see, commands are case-insensitive

(Skip it) Change network adapter VLAN ID

```
PS C:\> Set-NetAdapter -Name "Ethernet" -VlanID 10
```

Change network adapter MAC address

```
PS C:\> Set-NetAdapter -Name "Ethernet" -MacAddress "00-10-18-57-1B-0D"
```

Restore network adapter MAC address to its factory value

```
PS C:\> Set-NetAdapter -Name "Ethernet" -MacAddress ""
```

List connection profiles

```
PS C:\> Get-NetConnectionProfile
```

Display custom list of connection profile properties

```
PS C:\> Get-NetConnectionProfile | Format-Table -Property InterfaceAlias, Name,
NetworkCategory
```

Change connection profile of an interface by alias

```
PS C:\> Set-NetConnectionProfile -InterfaceAlias "Lan-Internal" -NetworkCategory Private
```

Change connection profile of an interface by index

```
PS C:\> Set-NetConnectionProfile -InterfaceIndex 9 -NetworkCategory Public
```

List all network addresses

```
PS C:\> Get-NetIPAddress
```

List only IPv4 addresses

```
PS C:\> Get-NetIPAddress | Where-Object -Property AddressFamily -Eq IPv4 | Format-Table -
Property InterfaceIndex, InterfaceAlias, IPAddress -AutoSize
```

List all properties of a network address

```
PS C:\> Get-NetIPAddress -InterfaceIndex 9 | Format-List *
```

Creates and configures an IP address

```
PS C:\> New-NetIPAddress -InterfaceIndex 9 -IPAddress 192.168.200.1 -PrefixLength 24 -
DefaultGateway 192.168.200.10
```

Removes an IP address and its configuration

```
PS C:\> Remove-NetIPAddress -InterfaceIndex 9 -IPAddress 192.168.200.1
```

Modifies the configuration of an IP address

```
PS C:\> Set-NetIPAddress -InterfaceIndex 9 -IPAddress 192.168.200.1 -PrefixLength 24
```

## Firewall Management (GUI)

We can manage it via the **Windows Defender Firewall with Advanced Security** management console which can be started from **Server Manager** or directly via **WF.msc**

## Firewall Management (CMD Shell)

List all dynamic inbound firewall rules

```
C:\> netsh advfirewall firewall show rule name=all dir=in type=dynamic
```

Show details about particular firewall rule

```
C:\> netsh advfirewall firewall show rule name="File and Printer Sharing (Echo Request -
ICMPv4-In)"
```

Enable firewall rule

```
C:\> netsh advfirewall firewall set rule name="File and Printer Sharing (Echo Request -
ICMPv4-In)" new enable=yes
```

Disable firewall rule

```
C:\> netsh advfirewall firewall set rule name="File and Printer Sharing (Echo Request -
ICMPv4-In)" new enable=no
```

## Firewall Management (PowerShell)

List configured profiles

```
PS C:\> Get-NetFirewallProfile
```

List all firewall rules

```
PS C:\> Get-NetFirewallRule
```

Search for firewall rules matching a pattern

```
PS C:\> Get-NetFirewallRule *icmp4*
```

Enable predefined firewall rule

```
PS C:\> Enable-NetFirewallRule -DisplayName "File and Printer Sharing (Echo Request -
ICMPv4-In)"
```

Disable predefined firewall rule

```
PS C:\> Disable-NetFirewallRule -DisplayName "File and Printer Sharing (Echo Request -
ICMPv4-In)"
```

# Part 2: Software and Services

For this part we will need just a **Windows Server** machine. We won't need any additional hardware. For some of the exercises we will need an internet connection

## Software (GUI)

Under **Windows Server** the graphical approach to install and manage roles, services, and features is through **Server Manager**

## Software (PowerShell)

To list all features both installed and available under **Windows Server** we can execute

```
PS C:\> Get-WindowsFeature
```

Let's install a specific role or (in this case) feature

```
PS C:\> Add-WindowsFeature XPS-Viewer
```

Alternatively, we can use the command itself

```
PS C:\> Install-WindowsFeature XPS-Viewer
```

Respectively, should we want a role or (this case) feature removed, we can go with:

```
PS C:\> Remove-WindowsFeature XPS-Viewer
```

Or with the command itself:

```
PS C:\> Uninstall-WindowsFeature XPS-Viewer
```

Please note, that the **Uninstall** command in addition to feature removal, deletes also all the corresponding files from the machine, while the **Remove** command just "deactivates" the feature

## Package Management

Package management under **Windows Server** resembles in a way package management under Unix-like operating systems. We can start with receiving information about loaded providers:

```
PS C:\> Get-PackageProvider
```

Then, we can list all available providers on the local machine:

```
PS C:\> Get-PackageProvider -ListAvailable
```

We can list all available package providers that can be installed with:

```
PS C:\> Find-PackageProvider
```

If asked to install the **NuGet** provider, confirm by typing **Y** and press **Enter**

*If not, we can install it manually, by executing the following command*

```
PS C:\> Install-PackageProvider -Name NuGet -RequiredVersion 2.8.5.201 -Force
```

Then, list all versions of a package provider

```
PS C:\> Find-PackageProvider -Name "Nuget" -AllVersions
```

Or, install a package provider automatically with:

```
PS C:\> Get-PackageProvider -Name "Chocolatey" -ForceBootstrap
```

To list all available packages from all providers, execute:

```
PS C:\> Find-Package
```

To list all available packages from a specific provider, execute

```
PS C:\> Find-Package -ProviderName Chocolatey
```

We can list all available packages that match the given pattern

```
PS C:\> Find-Package *notepad*
```

*If we see an error that says:*

***WARNING: NuGet: The request was aborted: Could not create SSL/TLS secure channel.***

*Then, we can execute the following commands to deal with it*

***reg add HKLM\SOFTWARE\Microsoft\.NETFramework\v4.0.30319 /v SystemDefaultTlsVersions /t REG_DWORD /d 1 /f /reg:64***

***reg add HKLM\SOFTWARE\Microsoft\.NETFramework\v4.0.30319 /v SystemDefaultTlsVersions /t REG_DWORD /d 1 /f /reg:32***

*More information, here: https://devblogs.microsoft.com/nuget/deprecating-tls-1-0-and-1-1-on-nuget-org/*

Or search for a particular package

```
PS C:\> Find-Package zoomit
```

Finally, we can install package

```
PS C:\> Install-Package XmlNotepad
```

Or, install it from a particular provider

```
PS C:\> Install-Package zoomit -ProviderName "Chocolatey"
```

As usual, we can combine commands to automate the steps involved in the installation process

```
PS C:\> Find-Package zoomit | Install-Package
```

The removal of a package is as simple as this:

```
PS C:\> Uninstall-Package zoomit
```

## Service Management (GUI)

Under **Windows Server** the graphical approach to manage services is through the **Services** console

## Service Management (CMD Shell)

To list all started services, we can execute:

```
C:\> net start
```

Let's try to stop a service

```
C:\> net stop "Windows Firewall"
```

*Some services may refuse to stop. If this is the case, pickup another one. For example, the **SNMPTRAP** one*

And then start it back again

```
C:\> net start "Windows Firewall"
```

We can query the service controller for all active services with:

```
C:\> sc query type= service
```

Should we want information for inactive services only, we can execute:

```
C:\> sc query type= service state= inactive
```

Services have display and internal name. Let's ask for a service by its display name:

```
C:\> sc GetKeyName "Windows Firewall"
```

And now, ask by its internal name:

```
C:\> sc GetDisplayName MpsSvc
```

General information about a service can be received with:

```
C:\> sc query MpsSvc
```

To stop a service, we can execute:

```
C:\> sc stop MpsSvc
```

To change for example, the service start mode, we can execute:

```
C:\> sc config MpsSvc start= disabled
```

Configuration information for a service, can be returned with:

```
C:\> sc qc MpsSvc
```

It is time to start the service again with:

```
C:\> sc start MpsSvc
```

## Service Management (PowerShell)

To list all services, execute:

```
PS C:\> Get-Service
```

In order to get properties of a service, execute:

```
PS C:\> Get-Service MpsSvc | Format-List -Property *
```

A list of all running services can be received with:

```
PS C:\> Get-Service | Where-Object -Property Status -Eq "Running"
```

A list of all services that can be suspended, can be received with:

```
PS C:\> Get-Service | Where-Object -Property CanPauseAndContinue -Eq "True"
```

We can force a service to restart:

```
PS C:\> Restart-Service -Name Winmgmt -Force
```

Or pause a service:

```
PS C:\> Suspend-Service Winmgmt
```

And then resume it:

```
PS C:\> Resume-Service Winmgmt
```

Change a service startup mode

```
PS C:\> Set-Service MpsSvc -StartupType Disabled | Automatic
```

Change a service description

```
PS C:\> Set-Service -Name Winmgmt -Description "System management."
```

Stop a service

```
PS C:\> Stop-Service Winmgmt
```

Start a service

```
PS C:\> Start-Service Winmgmt
```

# Part 3: Disks

Take a moment to ensure that the virtual machine you are working with has two empty additional hard drives – 10GB and 50GB

We will work with both disks to exercise disk, partition, and volume management activities

## Disk Management (GUI)

Under Windows Server the graphical approach to manage services is through the **Disk Management** console. It can be started via the **Server Manager** or from the command line with **diskmgmt.msc**

## Disk Management (CMD Shell)

# List all disks

```
DISKPART> LIST DISK
```

# List all volumes

```
DISKPART> LIST VOLUME
```

# Select disk

```
DISKPART> SELECT DISK 1
```

# List all partitions on a selected disk

```
DISKPART> LIST PARTITION
```

Initialize (or bring online) a disk

```
DISKPART> ONLINE DISK
```

Clear the read-only attribute of a disk

```
DISKPART> ATTRIBUTE DISK CREAL READONLY
```

In order to create partition on a selected disk that consumes all available space, execute:

```
DISKPART> CREATE PARTITION PRIMARY
```

Instead of the above, we will create a partition on the selected disk with a specific size in MB

```
DISKPART> CREATE PARTITION PRIMARY SIZE=1000
```

Let's select the partition

```
DISKPART> SELECT PARTITION 1
```

And delete it

```
DISKPART> DELETE PARTITION
```

We can ask for disk details with

```
DISKPART> DETAIL DISK
```

Okay. Let's create again the partition we deleted earlier:

```
DISKPART> CREATE PARTITION PRIMARY SIZE=1000
```

Ask for details about the partition

```
DISKPART> DETAIL PARTITION
```

Let's show list of supported file systems for a selected volume or partition

```
DISKPART> FILESYSTEMS
```

We can extend a selected volume or partition to the maximum

```
DISKPART> EXTEND
```

Instead, let's extend the selected volume or partition with a specific size in MB

```
DISKPART> EXTEND SIZE=500
```

In order to be able to store data, we must format the volume. Let's do it for the selected one

```
DISKPART> FORMAT FS=NTFS LABEL="New Volume" QUICK
```

Let's assign a drive letter to the selected volume

```
DISKPART> ASSIGN LETTER=H
```

Should we want to remove any and all partition or volume formatting from the disk with focus, we can execute this command (skip it for now):

```
DISKPART> CLEAN
```

Let's exit the **DISKPART** utility with

```
DISKPART> EXIT
```

We have several other commands available to manipulate volumes

We can re-format the **H** drive, we created earlier with:

```
C:\> FORMAT H: /Q
```

Or extend the command to add label, file system and compression settings:

```
C:\> FORMAT H: /V:DATA /FS:NTFS /C
```

Should we want, we can change label of a partition or volume

```
C:\> LABEL H: NEW
```

## Disk Management (PowerShell)

To list all disks, execute

```
PS C:\> Get-Disk
```

Let's initialize a disk

```
PS C:\> Initialize-Disk -Number 2 -PartitionStyle MBR
```

Convert disk partition scheme

```
PS C:\> Set-Disk -Number 2 -PartitionStyle GPT
```

(Skip it) Remove partition and volume information from disk

```
PS C:\> Clear-Disk -Number 2
```

(Skip it) To remove partition and volume information from disk with data partition present

```
PS C:\> Clear-Disk -Number 2 -RemoveData
```

Get partitions of all disks

```
PS C:\> Get-Partition
```

Let's ask for all partitions of a disk

```
PS C:\> Get-Partition -DiskNumber 2
```

(Skip it) In order to create a partition that consumes the whole disk, execute

```
PS C:\> New-Partition -DiskNumber 2 -UseMaximumSize -AssignDriveLetter
```

Let's create new partition with a specific size

```
PS C:\> New-Partition -DiskNumber 2 -AssignDriveLetter -Size 20GB
```

Create new partition with a specific size and drive letter

```
PS C:\> New-Partition -DiskNumber 2 -DriveLetter Z -Size 2GB
```

(Skip it) If we used the MBR partition scheme, we would like to make it active

```
PS C:\> Set-Partition -DriveLetter Z -IsActive $True
```

Let's change the drive letter

```
PS C:\> Set-Partition -DriveLetter Z -NewDriveLetter Y
```

Resize partition and underlying file system

```
PS C:\> Resize-Partition -DiskNumber 2 -PartitionNumber 3 -Size 20GB
```

Remove partition by drive letter

```
PS C:\> Remove-Partition -DriveLetter Y
```

Remove drive partition by disk and partition index

```
PS C:\> Remove-Partition -DiskNumber 2 -PartitionNumber 2
```

Get list of volumes

```
PS C:\> Get-Volume
```

Now, let's create again one new partition with a specific size and drive letter

```
PS C:\> New-Partition -DiskNumber 2 -DriveLetter Z -Size 2GB
```

Get details about a volume

```
PS C:\> Get-Volume -DriveLetter Z | Format-List *
```

Get supported file systems

```
PS C:\> Get-SupportedFileSystems -DriveLetter Z
```

Quick format (NTFS) of a volume

```
PS C:\> Format-Volume -DriveLetter Z
```

Quick format with explicit file system

```
PS C:\> Format-Volume -DriveLetter Z -FileSystem FAT32
```

Full format with explicit file system

```
PS C:\> Format-Volume -DriveLetter Z -FileSystem FAT32 -Full -Force
```

(Skip it) Create new volume out of an empty GPT disk

```
PS C:\> New-Volume -DiskNumber 2 -FriendlyName "Test" -DriveLetter Y -FileSystem NTFS
```

Set volume label

```
PS C:\> Set-Volume -NewFileSystemLabel "Data" -DriveLetter Z
```

(Skip it) Change volume label

```
PS C:\> Set-Volume -FileSystemLabel "Test" -NewFileSystemLabel "Test Data"
```