Practice M1: Introduction to Windows Server

In this module we will put the foundations of our course lab environment. As a main workstation you can use either a hosted environment, your laptop, or a home/office PC trough a remote connection. The only requirement is to support hardware virtualization.

Part 1: Lab Preparation

Firstly, we must pick up an appropriate virtualization option, then a solution, and finally install it

Please refer to the **M0-Guide-Introduction-to-Virtualization** document for information about three of the most popular options

For the purpose of the demonstration and depending on the available time the following combinations will be shown:

- **VirtualBox** running on Windows 10 host or Linux host
- **VMware Workstation** (Player) running on Windows 10 host or Linux host
- **Hyper-V** role installed on Windows 10 or Windows Server 2019 host

Once we have the virtualization solution installed and configured, we must obtain the install media. For this, we can use evaluation images:

- **Windows 10** - https://www.microsoft.com/en-us/evalcenter/download-windows-10-enterprise
- **Windows Server 2019** - https://www.microsoft.com/en-us/evalcenter/download-windows-server-2019
- **Windows Server 2022** - https://www.microsoft.com/en-us/evalcenter/download-windows-server-2022

Additionally, for the sake of doing additional experiments and studying, you can download as well:

- **Windows Server 2012 R2** - https://www.microsoft.com/en-us/evalcenter/download-windows-server-2012-r2
- **Windows Server 2016** - https://www.microsoft.com/en-us/evalcenter/download-windows-server-2016

You can download **Essentials edition** as well if you are curious. Please note that is not suitable for some of our exercises throughout the course

Now, we shall be ready to continue our journey

Part 2: Install Windows Server (by Next-Next method)

From now on all exercises will assume that we are working on a **Windows** host with **Hyper-V** role installed

We will be using **Windows Server 2019** installation media, except if something else explicitly stated

Even if you decided to use a different combination of a host OS and a virtualization solution, all steps will be applicable with small modifications or no modifications at all

Create Virtual Machine in Hyper-V

Open **Hyper-V Manager**

Initiate new **VM** creation by clicking on **Action > New > Virtual Machine** or by picking the same option, but from the **Actions** section on your right

Click **Next**

Specify a name for the **VM**. For example, **SU-SRV-1**. Click **Next**

Choose either **Generation 1** or **Generation 2** (*generally speaking, Generation 1 virtual machines are BIOS based, while Generation 2 are UEFI based*) and click **Next**

Set the amount of **Startup memory** to the desired value or accept its default value of **1024MB** and click **Next**

Leave the **Connection** parameter to **Not Connected** and click **Next.**

Accept the **Size** of the virtual hard disk to its default value (*in case of Hyper-V it is **127GB** dynamically expanding disk, which means that it will start small (around 4MB) and will grow up until the configured size*) and click **Next**

Click **Next**

On the summary screen, click **Finish.**

Edit Virtual Machine Settings

Now we should install an **OS** to our new **VM**. First, we must add a **DVD** drive and attach the appropriate **ISO** image

Select the **VM** in the **Virtual Machines** section

From the context menu, choose the **Settings** command. The same can be accomplished by using the machine-related commands in the bottom-right section of the screen

Select the **SCSI Controller** in the **Hardware** section

Select **DVD** and click **Add**

Select the newly added **DVD Drive**

For **Media** choose **Image file** and click **Browse**

Navigate to the desired image file and confirm with **Open**

Close the Settings window by clicking on **OK**

Connect to the VM and Power It On

With VM selected click Connect either in the machine's context menu or in the bottom-right section of the screen

Power on the machine by clicking on the Start button

Windows Server Installation

If the **VM** doesn't boot from the **DVD** we can press **Enter** to reset it and the once again to boot from the media

On the first screen of the **Windows Setup** we can adjust the values according to our liking or accept the default values and click **Next**

Click on the **Install now** button

On the next screen click **Next**

Select the **I accept the license terms** option and click **Next**

Click on the **Custom: Install Windows only (advanced) option**

Here we can choose a hard disk and repartition it, but for now we will just click on **Next**

Now we must wait until installation finishes. Then we will be prompted to reboot

Once installation is over and the machine is rebooted, we will be prompted to set a password for the **Administrator**

Now we have a fresh and fully functional installation of **Windows Server 2019 Core**

Do a Basic Server Core Configuration (The Easy Way)

Log on to the early installed **Windows Core** machine

Type the following command and hit **Enter**:

```
sconfig
```

If we see a warning that there aren't any **Network Adapters** configured, we must edit the **VM** settings and attach the **Network Adapter** to a **Virtual Switch**

Now we can change the machine name and / or the **Workgroup**

Then we can create additional account(s) with administrative rights

Control the remote connectivity to the machine

Change the way updates are installed

We can do few other tasks from this application (script) as well. Everything else we can do on the command line or through external management system

Install Windows Server (With GUI)

Now let's create a new **Generation 2 VM**, but this time set the memory to **2048MB** and link the **Network Adapter** to an existing **Virtual Switch**

Edit the settings and add a **DVD Drive** and point it to the **Windows Server ISO** file

Then change the **Boot order** and save the configuration

Connect to the machine and power it on

During the setup select **Windows Server 2019 Standard (Desktop Experience)**

Follow the steps to the end

Once the machine is ready, set the **Administrator** password

Log on in the system

With the **Server Manager** started select **Local Server** in the left section

Let's examine our system

Now we can change few settings like **Computer Name**, **Workgroup**, **Time Zone**, and etc.

Prepare a VM Template

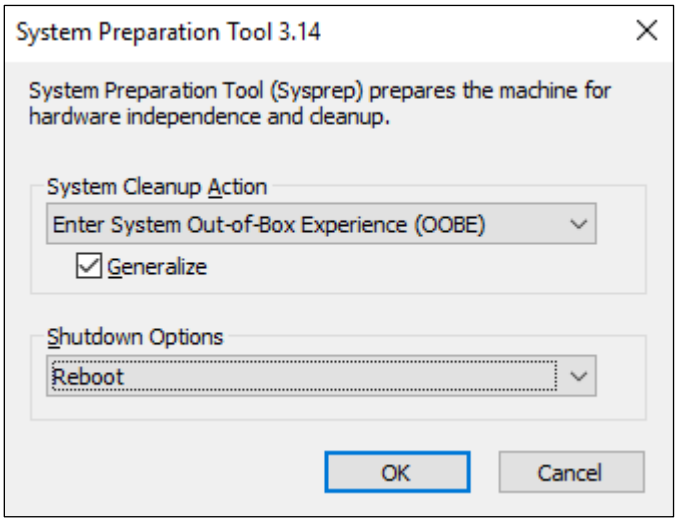
Typically, we do prepare the so-called template machines. This saves us time when we need multiple machines with similar characteristics. Depending on the virtualization platform in use, we can approach the task differently

It is good practice, before producing the actual template, to run set of tasks to prepare the machine. This is mostly applicable to **Windows** based virtual machines, especially when they will be part of an **Active Directory** setup. There is a special tool available for this purpose, it is called **SysPrep**

Let's use one of the machines created earlier. Open CMD window and type:

C:\Windows\System32\Sysprep\sysprep.exe

You can use the **Tab** key, instead of typing the whole string. Then hit **Enter**. You should see a window like this:



Typically, you would set the **Shutdown Options** to **Shutdown**, but now leave it to **Reboot**. Click on **OK**

After a while, the machine will reboot and will enter in initial configuration mode

Part 3: Work on the Command Line

Work on the CMD Shell

Log on to either of the machines. If you are logged on machine with **GUI**, then from the **Start menu** open **Command Prompt** application

Now, let's see what internal commands we have at our disposal by typing:

```
help
```

Let's get help about the **help** command itself:

```
help /?
```

Clear the screen with:

```
cls
```

Do you know what version of **Windows** we are running? We can check with:

```
ver
```

Okay, let's get more detailed information about the system with:

```
systeminfo
```

Because the information doesn't fit on the screen, we can combine (pipe) two commands:

```
systeminfo | more
```

We can scroll the information line-by-line with **Enter** key, or page-by-page with **Spacebar** key. We can quit by pressing the **Q** key

If we want to get some information about the license, we can do it by using the **slmgr.vbs** script

First, let's see what parameters are accepted:

```
slmgr.vbs
```

We can omit the extension. Now let's check current license:

```
slmgr /dlv
```

If we use a trial (time-based evaluation) version and our period (180 days for **Windows Server** or 90 days for **Windows 10 Enterprise**) is about to expire or expired already, we can reset (rearm) the counter with:

```
slmgr /rearm
```

Now let's examine what local users we have. This we will do by using one very old, but powerful command – **net**

If we just type:

```
net
```

We can see how many directions it covers. If we execute:

```
net help
```

We will get some helpful information. Now get information about sub-command:

```
net help user
```

So, in order to get information about the local users, we should type:

```
net user
```

Let's check the information about the built-in users Administrator and Guest:

```
net user administrator
```

```
net user guest
```

What about to create a user:

```
net user demo Password1 /add
```

Alternative syntaxis, in case you do not want to write the password in plain text, would be:

```
net user demo * /add
```

And the groups:

```
net localgroup
```

Let's create a new one:

```
net localgroup "Demo Group" /add /comment:"Demo group created during the practice"
```

And add our **demo** user to it and to the **Administrators** group:

```
net localgroup "Demo Group" demo /add
```

```
net localgroup Administrators demo /add
```

Again, check the info about our **demo** user:

```
net user demo
```

If we are working in an installation with **Desktop Experience**, we can check the graphical tool for managing users and groups. Open **Server Manager** if not opened already. Choose **Computer Management** from the **Tools** menu in the top-right corner. Then go to **Local Users and Groups**. Check if the user and group created earlier are seen here. Create one more user and add it to the same group. Now return to the console and check the members of the group

Now let's clear the artefacts that we created earlier:

```
net localgroup Administrators demo /delete
```

```
net localgroup "Demo Group" demo /delete
```

```
net user demo /delete
```

```
net user
```

Don't forget to delete the additional user, created via the GUI tool

There is a shutdown command. Let's check what are its parameters. This can be done by executing:

```
help shutdown
```

Or by:

```
shutdown /?
```

Restart the Core machine

```
shutdown /r /t 0
```

Or if we want to stop it, we can change the **/r** switch to **/s**

Work with PowerShell

Log on to either of the machines. If you are logged on machine with **GUI**, then from the **Start menu** open **Windows PowerShell** application. If on **Core** machine, then type:

```
powershell
```

Now, let's experience one interesting feature of **PowerShell** – command completion. Type:

```
Get-H
```

And press **Tab** key. The command will be extended to **Get-Help** first. Now press **Tab** key few more times until the command is again **Get-Help**. Now hit **Enter**

Let's check what **Update-Help** command does:

```
Get-Help Update-Help
```

In order to have appropriate help content, we must execute the following command:

```
Update-Help
```

There are so many commands, so we should be able to enumerate them in a way:

```
Get-Command
```

The list is too long, we can filter it by verb (action):

```
Get-Command -Verb Get
```

We can filter even further by module:

```
Get-Command -Module *LocalAccounts
```

Or by command name:

```
Get-Command -Name *User*
```

We can get all verbs as well with:

```
Get-Verb
```

But because the list is long, we can display it page by page:

```
Get-Verb | more
```

Now let's get list of local users:

```
Get-LocalUser
```

Let's examine the built-in **Administrator**:

```
Get-LocalUser Administrator
```

Okay, this output is not quite informative, so let's modify it:

```
Get-LocalUser Administrator | Select *
```

Don't worry about the above command chain, with the time we will get used to it

We can create one additional user with the following command:

```
New-LocalUser -Name Admin1
```

Enter the password and press **Enter** key

Now let's add a new group:

```
New-LocalGroup Demo
```

And local group list now looks like:

```
Get-LocalGroup
```

Let's add both users – **Administrator** and **Admin1** to the newly created group:

```
Add-LocalGroupMember -Group Demo -Member Admin1, Administrator
```

The list of **Demo** group members now can be visualized with:

```
Get-LocalGroupMember Demo
```

Now let's clean all artefacts that we created:

```
Remove-LocalGroupMember Demo -Member Administrator,Admin1
Remove-LocalGroup Demo
Remove-LocalUser Admin1
```