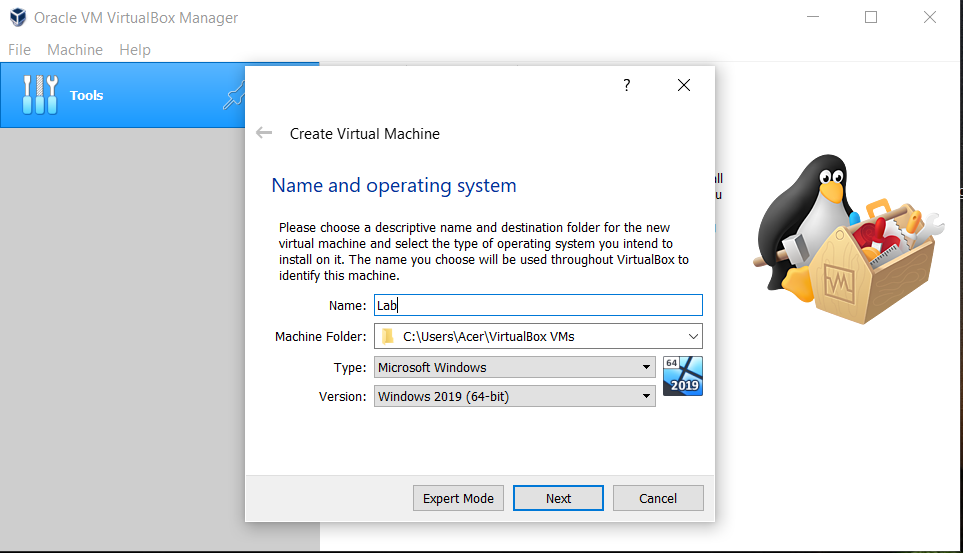
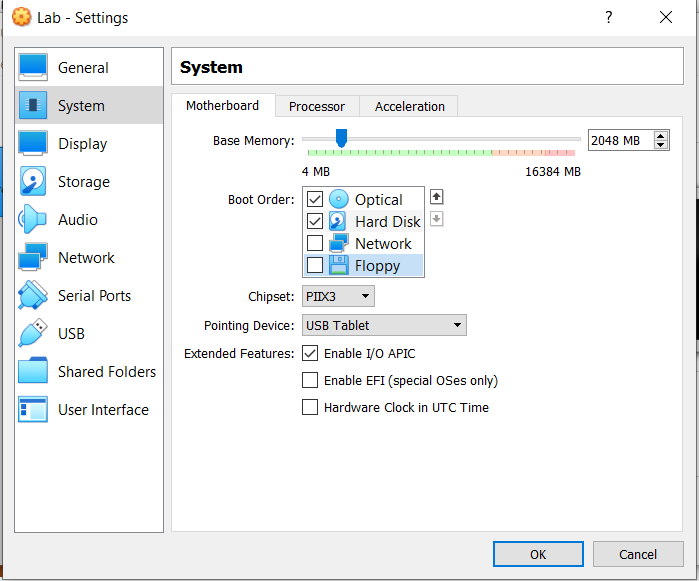
1. Creating a VM with Windows Server 2019

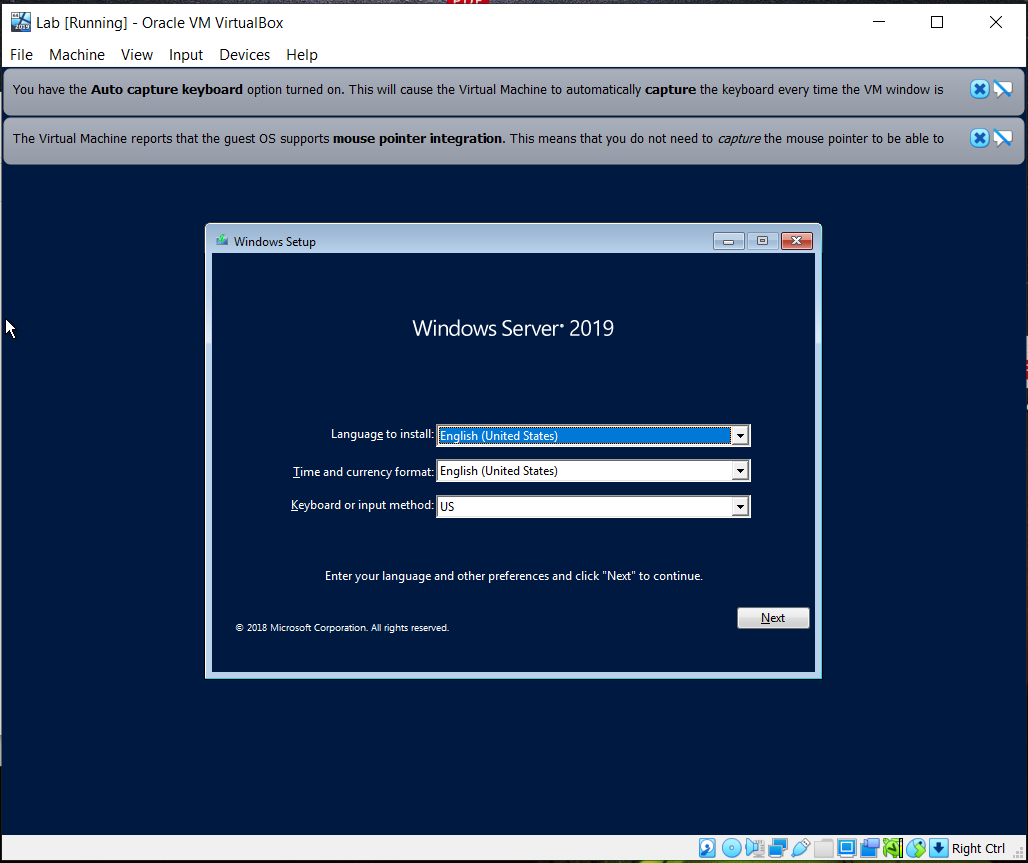
Set up the VM:



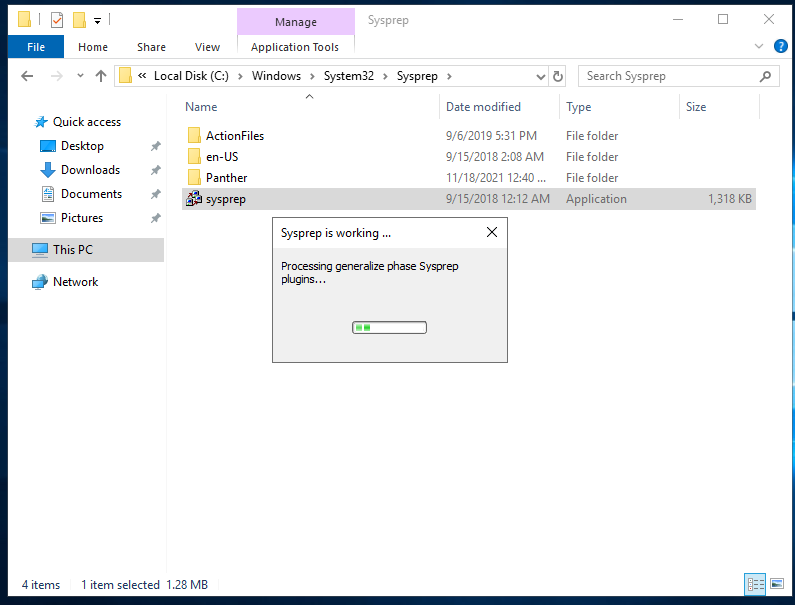
Set up the boot order:



Install Windows Server 2019 on the VM:



Create a template with sysprep:



Introduction to CMD and Powershell:

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**