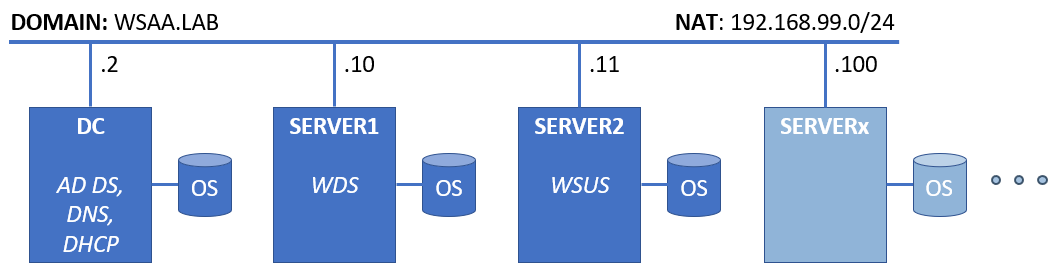
# Practice M6: Extended Services

The lab environment has the following structure:



Of course, it could be limited to just the DC and one other station (workstation or server) but all tasks should be adjusted accordingly

The following tasks are executed on different machines. It is stated clearly on which machine a task is being executed

## Part 1: Windows Deployment Services 101

### WDS (On SERVER1)

Let’s first prepare the server:

* Create a folder **C:\Deployment**
* Open **Server Manager** if not already open
* In **Manage** choose **Add Roles and Features**
* Then click **Next**
* Then do not make any changes and click **Next**
* Select the domain controller and click **Next**
* In **Server Roles** select **Windows Deployment Services** and click **Next**
* Confirm with **Add Features** and click **Next** again
* On the **Features** leave everything as is, and click **Next**
* Read the notes about the role and click **Next**
* Leave default selection of the two role services and click **Next**
* On the summary screen click **Install**
* Once the feature is installed click **Close**

Now we are ready to try a simple deployment:

* From **Server Manager** > **Tools**, select **Windows Deployment Services**
* Select our WDS server in left section
* It seems that we have to configure it
* Choose **Configure Server** from the context menu
* Read the notes and click **Next**
* Leave the defaults for installation options (**Integrated with Active Directory**) and click **Next**
* Navigate to **C:\Deployment** folder and click **OK**. Then click **Next**
* Click **Yes** on the warning message. We acknowledge that we are going to use our system volume for deployments as well
* Select **Do not respond to any client computers** and click **Next**
* Deselect the **Add images to the server now** option
* Click **Finish**
* If the service did not start, then we can start it manually

Let’s explore the nodes below our server

Once, we are done with the exploration, let’s attach the install DVD of Windows Server to the virtual machine

Next step is to add an **install image**:

* Select **Install Images**
* From the context menu click **Add Install Image**
* Enter image group name, for example **Demo**
* Click **Next**
* Click **Browse**
* Navigate to DVD drive (it should contain a **Windows Server 2016/2019** install media), then **sources**, and finally select **install.wim**
* Click **Open**
* Click **Next**
* Select only one of the images, for example **Standard Core**
* Uncheck the **Use the default name and description** option and click **Next**
* For **Image name** and **Image description** enter **Windows Server 2019 Std. Core** and click **Next**
* On the summary screen click **Next** and wait for the image to be copied
* Once, done click **Finish**

Now that we have an install image, we must add a **boot image** as well:

* Select **Boot Images**
* From the context menu choose **Add Boot Image**
* Click **Browse**
* Navigate to DVD drive (it should contain a **Windows Server 2016/2019** install media), then **sources**, and finally select **boot.wim**
* Click **Open**
* Click **Next**
* Accept the default values and click **Next**
* On the **Summary** screen click **Next** and wait for the image to be copied
* Click **Finish**

There are a few final adjustments that we can do:

* Select the server
* From the context menu choose **Properties**
* Check and modify, if needed (when DHCP and WDS are on the same machine), the settings on the **DHCP** tab
* Check and modify, if needed, the settings on the **Boot** tab
* Switch to the **PXE Response** tab
* Select **Respond to all client computers (known and unknow)**
* Select **Require administrator approval** option
* Explore the rest of the tabs as well
* Once ready, click **OK**

Let’s test the setup:

* Create a new VM
* Change its settings to be able to do PXE boot
* Don’t forget to connect its network adapter to the network of our lab
* Start the VM
* Once the WDS information is displayed, press the **Enter** key
* Then a message showing that an approval is needed will be shown
* Return on the **SERVER1** machine
* Open the **Windows Deployment Services** console
* Select the **Pending Devices** node and refresh if needed
* Select the only pending request and from the context menu select **Name and Approve**
* Enter **VM1** in the **Device Nam**e field and click **Next**
* On the **Boot** screen click again **Next**
* On the **Client Unattend** screen click **Next**
* On the **Join Rights** screen click on the **Configure User** button
* Then click **Select**
* Enter **Administrator** and click **Check Names**
* Then click **OK**
* Select **Full Rights** and click **OK**
* Finally, click **Finish**
* Watch if the deployment will start
* You can initiate the setup process and while it is going, continue with the next section

Now, we can try with a pre-staged machine:

* Create a new VM
* Change its settings to be able to do PXE boot
* Don’t forget to connect its network adapter to the network of our lab
* Start the VM
* Write down its **MAC** address
* Stop the VM
* Return on the **SERVER1** machine
* Open the **Windows Deployment Services** console
* Select the **Active Directory Prestaged Devices** node
* Invoke the context menu and select **Add Device**
* Enter **VM2** in the **Name** field
* Enter the MAC address of the VM without the dashes in the **Device ID** field
* Click **Next**
* On the **Boot** screen click again **Next**
* On the **Client Unattend** screen click **Next**
* On the **Join Rights** screen click on the **Configure User** button
* Then click **Select**
* Enter **Administrator** and click **Check Names**
* Then click **OK**
* Select **Full Rights** and click **OK**
* Finally, click **Finish**
* Start the VM
* Press Enter when required
* Watch if the deployment will start automatically
* You can initiate the setup process

You can switch to the **DC** machine and check in the **Active Directory Users and Computers** if the two machines are there

## Part 2: Windows Deployment Services 102

### Custom Image with DISM (On SERVER1)

Create two folders – **C:\custom** and **C:\mount**

Copy the **install.wim** file to the **C:\custom** folder

Open a **CMD** session (with Run as Administrator):

* Explore the file content with

**dism /get-wiminfo /wimfile:c:\custom\install.wim**

* Delete extra image #1 with

**dism /delete-image /imagefile:c:\custom\install.wim /index:1**

* Check again the remaining images

**dism /get-wiminfo /wimfile:c:\custom\install.wim**

* All indexes have been changed
* Delete images with #2 and #3

**dism /delete-image /imagefile:c:\custom\install.wim /index:2**

**dism /delete-image /imagefile:c:\custom\install.wim /index:2**

* The second command is not an error. When we delete the second image, the third becomes the second
* Mount the image

**dism /mount-image /imagefile:c:\custom\install.wim /index:1 /mountdir:c:\mount**

* Explore the **C:\mount** directory content
* To add a package, for example an update, we must download it first
* Navigate to <https://www.catalog.update.microsoft.com/Home.aspx>
* Search for example for **KB4586875**
* Explore package information
* Download the package for **Windows Server 2019** (**windows10.0-kb4586875-x64\_a14cffeb8033542a5379d4c5a9c0c6fa17c3f939.msu**)
* Create a folder **C:\packages** and copy it there
* Add the package to the image

**dism /image:c:\mount /add-package /packagepath:"c:\packages\windows10.0-kb4586875-x64\_a14cffeb8033542a5379d4c5a9c0c6fa17c3f939.msu"**

* In a similar fashion we can add other packages or drivers (with **/add-driver** and **/driver** switches)
* Check what roles and features are available in the image

**dism /image:c:\mount /get-features**

* It is a long list, but we can either scroll the screen or output it to a text file
* Let’s activate the IIS server role

**dism /image:c:\mount /enable-feature /featurename:iis-webserverrole**

* We can discard all our changes with (skip it)

**dism /unmount-wim /mountdir:c:\mount /discard**

* Or commit them with

**dism /unmount-wim /mountdir:c:\mount /commit**

Now, we can register this new image in **WDS** using the management console

Register it under the **Windows Server 2019 IIS** name

Repeat the procedure of creating a VM and testing the new image

This time we will be presented with a selection

### Custom Image with SysPrep (On Host)

Either use one of the previously created machines or add a new one (for example from template to save time)

Let’s go with the second option and create a new machine by using a template

Once the machine is running, log on to it

Let’s install a role by using the **Server Manager**

Again, install the **IIS** role but this time with **ASP.NET** support

Then, download and install an application

For example, the **ZoomIt** application. Download it from here: <https://download.sysinternals.com/files/ZoomIt.zip>

Extract it, for example, to **C:\BIN**

Let’s assume that we are done customizing our machine

Now, we must execute the system preparation step:

* Open a **CMD** session (with Run as Administrator)
* Navigate to **C:\Windows\System32\Sysprep**
* Execute either the **sysprep** command with no options
* Or with the following set of options (skip it) if preparing the image for a VM:

**sysprep /generalize /oobe /shutdown /mode:vm**

* In GUI select **Enter System Out-of-Box Experience (OOBE)** for **System Cleanup Action**
* Select the **Generalize** option
* Select **Shutdown** in the **Shutdown Options** list
* Confirm with **OK**

### Capture a Custom Image (On SERVER1)

Once the machine is powered off, return on **SERVER1**

Open the **Windows Deployment Services** console

Select the **Boot Images** node

Select the only item there and invoke the context menu

Select the **Create Capture Image** option

Enter **Capture Image** in the Image name field

Click the **Browse** button

Navigate to **C:\Custom**

Enter **Capture.wim** in the **File name** field and click **Open**

Click **Next**

Once the image creation is done, select the **Add image to the Windows Deployment Server** now option

Click **Finish**

A wizard to add the image will start

On the first screen click **Next**

Click **Next** on the second and on the third screen as well

Finally, click **Finish**

Now, make sure that the first boot device of the VM that we prepared earlier is set to Network Adapter

Then, power on the VM

A PXE boot will be initiated

Press **Enter** when requested

Return in the WDS console and approve the pending request by selecting **Approve** from the context menu

Return in the VM and in the **Boot Manager** select the **Capture Image** option and confirm with **Enter**

A capture image wizard will be launched

On the first screen click **Next**

Select **C:\** in the **Volume to capture** list

Enter **Win2k19IIS** in the **Image name** field

Enter **Windows Server 2019 with IIS** in the **Image description** field

Click **Next**

Click the **Browse** button

Navigate to **C:\**

Enter **Win2k19IIS.wim** in the **File name** field and click **Save**

Select the **Upload image to a Windows Deployment Services server item**

Enter **SERVER1** in the **Server name** field and click **Connect**

When asked, enter credentials, for example for user **WSAA\Administrator**

If authenticated successfully, select the **Demo** group in the **Image Group** **name** list

Click **Next**

Once the process is complete, click **Finish**

Turn off the VM

Return on **SERVER1** and check if the captured image is there

Now, we can create a new VM and test the new image

### Unattended Installation (On SERVER1)

Log on to the server

Download and install the **Windows Assessment and Deployment Kit** (**Windows ADK**) from this URL:

<https://docs.microsoft.com/en-us/windows-hardware/get-started/adk-install>

Install only the **Deployment Tools**

Start the **Windows System Image Manager** (**Windows SIM**) tool

In the **Windows Image** section invoke the context menu and select the **Select Windows Image** option

Navigate to **C:\Custom** and select the **install.wim** image

Confirm with **Yes** when asked if we want to create a new catalog

In the **Answer File** section invoke the context menu and select the **New Answer File** option

Find in the **Windows Image** section the **amd64\_Microsoft-Windows-International-Core-WinPE\SetupUILanguage** component and add it in **pass 1**

Set the **InputLocale**, **SystemLocale**, **UILanguage**, and **UserLocale** to **en-US**

Find in the **Windows Image** section the **amd64\_Microsoft-Windows-Setup\DiskConfiguration\Disk\Create Partitions** and **Modify Partitions** components and add them in **pass 1**

In **Create Partitions** add three partitions:

1. Order **1**, Size **500**, Type **Primary**
2. Order **2**, Size **100**, Type **EFI**
3. Order **3**, Extend **true**, Type **Primary**

In **Modify Partitions** add two partitions:

1. Format **NTFS**, Order **1**, PartitionID **1**
2. Format **NTFS**, Letter **C**, Order **2**, PartitionID **3**

Set in **Disk** node **DiskId** to **0** and **WillWipeDisk** to **true**

Add the **UserData** node to **pass 1** and set the **AcceptEula** to **true**

Find and add all nodes of **WindowsDeploymentServices**

Set the **InstallImage** node to

1. **Filename** to **install.wim** (make sure that the filename matches the actual filename of the install image)
2. **ImageGroup** to **Demo**
3. **ImageName** to **Windows Server 2019 SERVERSTANDARD**

Set the **InstallTo** node to

1. **DiskId** to **0**
2. **PartitionID** to **3**

Modify **Login\Credentials** to

1. **Domain** to **WSAA.LAB**
2. **Password** to **Password1**
3. **Username** to **Administrator**

Validate the file and save it as **C:\Windows-1.xml**

Create new **Answer File**

Add the **amd64\_Microsoft-Windows-Shell-Setup** option to **pass 4**

Set **TimeZone** to **FLE Standard Time**

Add **amd64\_Microsoft-Windows-International-Core** to **pass 7**

Sett all language and locale settings to **en-US**

Add **amd64\_Microsoft-Windows-Shell-Setup\OOBE** to **pass 7**

Set all **HideXXXX** settings to **true**

Set **ProtectYourPC** to **3**

Add **amd64\_Microsoft-Windows-Shell-Setup\UserAccounts\AdministratorPassword** to **pass 7**

Set the **Value** field to **Password1**

Validate the file and save it as **C:\Windows-2.xml**

Copy the **C:\Windows-1.xml** file to the **C:\Deployment\WdsClientUnattend** folder

Return to the **Windows Deployment Services** console

Select the server node and in the context menu click on **Properties**

Switch to the **Client** tab

Select the **Enable unattended installation** option

Click on the **Browse** button next to the **x64 (UEFI) architecture** option

Navigate to the file copied earlier and confirm

Close the properties window by clicking the **OK** button

Navigate to the install image (its name must match the one entered in the unattended file) and click **Properties** in its context menu

Select the **Allow image to install in unattended mode** option

Click the **Select File** button and navigate to **C:\Windows-2.xml**

Confirm and close the properties window by clicking on the **OK** button

Now, we can create a new VM and test the new setup

## Part 3: Windows Server Update Services 101

### WSUS (On SERVER2)

Let’s install the **WSUS** role:

* Create a folder **C:\Updates**. The name and the place are arbitrary, and it is up to you to choose the values. This will be the folder where all downloaded updates will be stored
* Open **Server Manager** if not already open
* In **Manage** choose **Add Roles and Features**
* Then click **Next**
* Then do not make any changes and click **Next**
* Select the domain controller and click **Next**
* In **Server Roles** select **Windows Server Update Services** and click **Next**
* Confirm with **Add Features** and click **Next** again
* On the **Features** leave everything as is, and click **Next**
* Read the notes and click **Next**
* Leave defaults and click **Next**
* Enter the path to the folder created earlier – **C:\Updates** and click **Next**
* On the summary screen click **Install**
* Once the feature is installed click **Close**

Alternative way to install WSUS is, of course, to use PowerShell:

* To install it with WID option:
  + Installation: **Install-WindowsFeature -Name UpdateServices -IncludeManagementTools**
  + Configuration: **wsusutil.exe postinstall content\_dir=D:\WSUS**
* To install it and prepare it for connection to an external SQL Server:
  + Installation: **Install-WindowsFeature -Name UpdateServices-Services, UpdateServices-DB -IncludeManagementTools**
  + Configuration: **wsusutil.exe postinstall sql\_instance="SQLSERVER\WSUS" content\_dir=D:\WSUS**

Of course, you must adjust the settings according to your environment

We must launch the post-installation wizard and finalize the installation:

* Click on the **Flag** icon in the **Server Manager**
* Select **Launch Post-Installation** **tasks**
* Run the **Windows Server Update Services** tool
* On the title screen click **Next**
* On the next screen if you want then opt-out from the **Microsoft Update Improvement Program** and click **Next**
* Choose an upstream server. For the purpose of the tests, you can choose either of both:
  + Use Microsoft Update servers (so, leave the default selection and click **Next**)
  + Enter an upstream server – **wsus.tuionui.com** (offers **Windows 10** and **Windows Server 2019**) and a port **8530**, and then click **Next**
* On the **Proxy** screen, if you do not have proxy, then just click **Next**, otherwise enter the proxy settings and continue to the next screen with **Next**
* We should download information about the available products and classifications from our upstream server, so must click **Start Connecting**
* After a while (it can take 20 or so minutes if the upstream server is Microsoft Update public server), when the job is done, click **Next**
* Select languages to download, for example select only **English** and click **Next**. If we selected upstream server other than the official one, our language selection will be limited to the languages offered by the server
* Narrow the product selection to just what you need. For the purpose of the lab, we will select only the options for **Windows Server 2019** in section **Windows**. Click **Next**. If we selected upstream server other than the official one, we won’t see dialog for products selection, we will be limited to the products offered by the server
* Leave the options with their default values and click **Next**. If we selected upstream server other than the official one, we won’t see dialog for classification options, we will be limited to what the server offers
* Leave the synchronization to manual and click **Next** or we can set a synchronization schedule
* Select **Begin initial synchronization** (this can take a long time, depending on the selected products and classifications) or leave it unselected and click **Next**. We can initiate synchronization later as well
* Read the notes and click **Finish**

Don’t forget to download and install both **Microsoft System CLR Types for Microsoft SQL Server 2012** and **Microsoft Report Viewer 2012 Runtime**

Then you can explore the **Reports** section in the WSUS console

Now we can set a global policy to control the updates in our environment. Basically, there are two options – to change the **Default Domain Policy** or to create a new **GPO**. For the purpose of the lab, we will choose the first option:

* While on the domain controller, open **Server Manager** if not already open
* Click on **Group Policy Management** in the **Tools** menu
* Select the **Default Domain Policy** (or create e new one, liked on domain level) and from the context menu click **Edit**
* Change the following settings:
  + **Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Update\Configure Automatic Updates** set it to **Enabled** and for **automatic updating** set **3 – Auto download and notify for install**. Click **OK**
  + **Computer Configuration \Policies\Administrative Templates\Windows Components\Windows Update\Specify intranet Microsoft Update service location** set it to **Enabled** and fill-in first two options with the **FQDN** of the **WSUS** server, which in our case is the **SERVER2**, so we should enter [**http://server2.wsaa.lab:8530**](http://server2.wsaa.lab:8530) . Leave the last (the third option) blank and click **OK**
  + Optionally, depending on our **WSUS** configuration, we may need to set also **Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Update\Enable client-side targeting**
* Close the editor

We can force the settings by executing set of commands in a **CMD** shell as **Administrator** on a chosen machine:

* In order to force the update of the GPOs, we can execute: **gpupdate /force**
* Ask Windows Update to check if there are new updates available: **wuauclt /resetauthorization /detectnow**
* Show the scan results: **wuauclt /reportnow**

Alternatively, you can check the Windows Update logs

* This used to be done by checking the contents of the file **%WINDIR%\WindowsUpdate.log**
* Nowadays, we can use the **Get-WindowsUpdateLog** command in PowerShell session