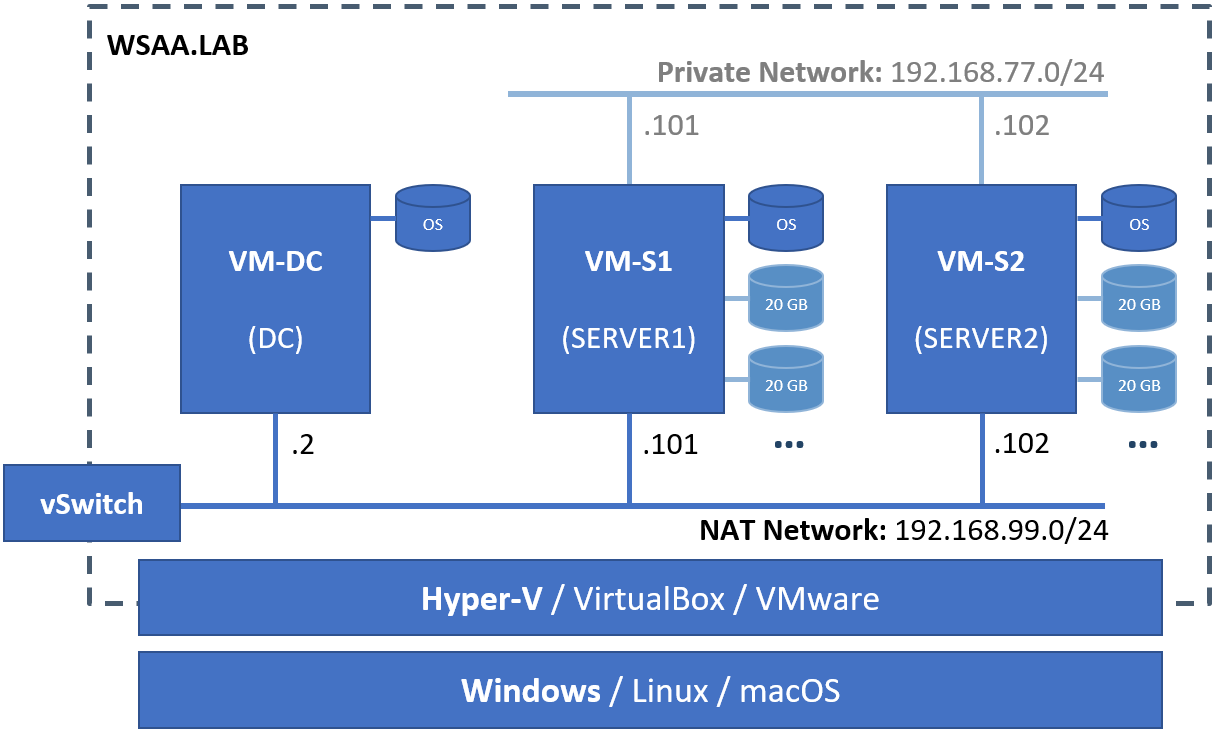
# Practice M1: Storage Solutions

For the purpose of this lab and the course, we will consider that we are working in a pure Windows environment either on-premise or in the cloud and using **Hyper-V** as a virtualization solution. All tasks can be achieved under different configuration (another host OS or virtualization solution) with the appropriate adjustments

The expected lab infrastructure consists of three machines – one domain controller (**DC**) and two member servers (**SERVER1** and **SERVER2**). Each machine will have one OS disk and 1 NIC for start. Windows Server 2019 Standard is enough, Desktop experience or core – it is up to you



Make sure that you have a **regular user** (member of just the **Domain Users** group) in your domain

If you are short on resources, you can shrink the infrastructure to two machines – domain controller and a member server and adjust the tasks accordingly

## Part 1: Storage 101

Now, let’s practice a few simple tasks like basic disk management with GUI and on the PowerShell terminal. Then we will continue with sharing

### Disk Management

#### The GUI way

Let’s first add **two dynamically expanding disks**, each **20GB** in size, to **WSAA-M1-VM-S1** (**SERVER1**)using the **Hyper-V Manager** (or the appropriate tool/technique in your case) on the host

Log onto **SERVER1** with an account with administrative privileges and start **Disk Management** tool

You should see both additional disks. They are in **offline** state

Click on the **first disk** (on its header, not the space part) and from the context menu select **Online** to change its state

Again, select the header and from the context menu select **Initialize Disk**

Select the **MBR (Master Boot Record)** partition style and click **OK**

Now, select the space representation of the disk and from the context menu chose **New Simple Volume**

Click on **Next**

Accept the maximum size offered and click **Next**

Make sure that the **Assign the following drive letter** option is selected and that the value is **D** and click **Next**

Explore the options of different file systems. Finally, select **NTFS**, leave the **Allocation unit size** to **Default** and in the **Volume label** field enter **DISK1**. Click **Next**

Check the summary screen and if all is okay, click on the **Finish** button

Click on the **second disk** (on its header, not the space part) and from the context menu select **Online** to change its state

Again, select the header and from the context menu select **Initialize Disk**

Select the **GPT (GUID Partition Table)** partition style and click **OK**

Now, select the space representation of the disk and from the context menu chose **New Simple Volume**

Click on **Next**

Accept the maximum size offered and click **Next**

Make sure that the **Assign the following drive letter** option is selected and that the value is **E** and click **Next**

Make sure that you select **NTFS**, leave the **Allocation unit size** to **Default** and in the **Volume label** field enter **DISK2**. Click **Next**

Check the summary screen and if all is okay, click on the **Finish** button

Both disks appear to have different amount of available space. Why?

#### The PowerShell way

*Please note, that the following lines are only applicable if you use Hyper-V as a virtualization solution. If not, you may execute them in a similar fashion to the one used in the previous paragraph*

Now, let’s repeat the same procedure on **WSAA-M1-VM-S2** (**SERVER2**) including the disk creation but this time with **PowerShell**

Return to the **Hyper-V** host and open a **PowerShell** session with **Run as Administrator**

To create the two disks, execute the following two commands

**New-VHD -Path C:\VM\WSAA-M1-VM-S2-D1.vhdx -SizeBytes 20gb -Dynamic**

**New-VHD -Path C:\VM\WSAA-M1-VM-S2-D2.vhdx -SizeBytes 20gb -Dynamic**

In order to attach them to **WSAA-M1-VM-S2** (**SERVER2**), we can execute the following two commands, again on the **Hyper-V** host

**Add-VMHardDiskDrive -VMName WSAA-M1-VM-S2 -Path C:\VM\WSAA-M1-VM-S2-D1.vhdx**

**Add-VMHardDiskDrive -VMName WSAA-M1-VM-S2 -Path C:\VM\WSAA-M1-VM-S2-D2.vhdx**

Now, we can log on the **SERVER2** machine with an account with administrative privileges and continue from there or execute all here on the **Hyper-V** host

Let’s follow the second approach - continue working on the Hyper-V host

First, we need to get some information about the disks

**Invoke-Command -VMName WSAA-M1-VM-S2 -ScriptBlock { Get-Disk | Select DiskNumber, PartitionStyle, IsOffline }**

When asked for credentials, enter some but for an account with administrative privileges (for example for **WSAA\Administrator**)

Now, we have what we need, and we can initialize the disks

**Invoke-Command -VMName WSAA-M1-VM-S2 -ScriptBlock { Initialize-Disk -Number 1 -PartitionStyle MBR }**

**Invoke-Command -VMName WSAA-M1-VM-S2 -ScriptBlock { Initialize-Disk -Number 2 -PartitionStyle GPT }**

Entering so many times the credentials is kind of annoying. Let’s fix this. Execute:

**$c = Get-Credential**

Enter the credentials once more

It is time to create partitions and format them a file system. To set the partitions, execute:

**Invoke-Command -VMName WSAA-M1-VM-S2 -ScriptBlock { New-Partition -DiskNumber 1 -AssignDriveLetter -UseMaximumSize } -Credential $c**

**Invoke-Command -VMName WSAA-M1-VM-S2 -ScriptBlock { New-Partition -DiskNumber 2 -AssignDriveLetter -UseMaximumSize } -Credential $c**

Finally, to prepare the file system, execute:

**Invoke-Command -VMName WSAA-M1-VM-S2 -ScriptBlock { Format-Volume -DriveLetter D -FileSystem NTFS -NewFileSystemLabel DISK1 } -Credential $c**

**Invoke-Command -VMName WSAA-M1-VM-S2 -ScriptBlock { Format-Volume -DriveLetter E -FileSystem NTFS -NewFileSystemLabel DISK2 } -Credential $c**

All the above commands can be executed directly in the VM but only the part that is in the **ScriptBlock**

Those can be combined or even automated especially if both partition schemes were the same. Try to come up with a solution

The same applies for the credentials, they can be saved upfront in a variable, instead entered for every single command. We did it but halfway through. Try to optimize it, for example ask for them once and use them in all the following commands

### SMB Sharing

#### Shares

Log on to **SERVER1** and open **File Explorer** with an account with administrative privileges

Navigate to **DISK1** and create a folder named **Accounting**

Enter the folder and create a file named **Budget.txt** (mind that the extension for known files is hidden by default) containing for example the number 100

Right-click the **Accounting** folder and select **Properties** from its context menu

Switch to the **Sharing** tab and click **Advanced Sharing**

Select the **Share this folder** option and in the **Share name** enter **Budgets**

Click on **Permissions** button

With **Everyone** selected remove all ticks under the **Allow** section

Click on the **Add** button

In the **Enter the object names to select** field enter **domain admins** and click on the **Check Names** button

Click **OK**

With **Domain Admins** highlighted turn on **Full Control** and click **Apply**

Repeat the same procedure for **Domain Users** but with **Allow Read** permissions

Click **OK** to close the **Permissions** window

Click **OK** to close **Advanced Sharing** window

Click **Close** to close the **Accounting Properties** window

#### Access test

Log on to **SERVER2** with a regular user account. For example, **regular.user**

Open **File Explorer** and navigate to [**\\SERVER1\Budgets**](file:///\\SERVER1\Budgets)

Open the **Budget.txt** file and change its content

Try to save the file. An error should appear

Close the file without saving and log out the server

Log on again but this time with an account part of the **Domain Admins** group

Repeat the procedure. You should be able to save the file

#### NTFS permissions

Return to **SERVER1** and log on with account part of the **Domain Admins**

Right-click on the **Accounting** folder created earlier

From the context menu select **Properties** and then switch to the **Security** tab

Click **Edit** and the **Permissions for Accounting** window will appear

Click the **Add** button to add the regular user used earlier. Find the user and click **OK**

Select the restricted account and grant him **Full Control** and click **OK**

Click **OK** to close the **Accounting Properties** window

Return to **SERVER2** and log on with the regular user

Try to edit and save the file again

Again, no success. Why? (*Share vs NTFS permissions, perhaps*)

Return to **SERVER1** but this time log on with the regular user

Try to edit and save the file

It works. Why? (*local NTFS permissions, perhaps*)

#### Server Manager sharing

Log on to **SERVER1** with a user with administrative privileges and open **Server Manager**

Navigate to **File and Storage Service** section

Click on **Shares**

Select **New Share** from the **Tasks** drop-down list

Select **SMB Share – Quick** and click **Next**

Make sure that the **Select by volume** is selected under the **Share location**

Select the **E:** drive and click **Next**

In the **Share name** field enter **Documents** and click **Next**

Select the **Enable access-based enumeration** option and click **Next**

On the permission window examine and accept the proposed configuration and click **Next**

Check the summary screen and click on **Create**

After the new share is created, click on **Close**

You can log on to the other server with a regular user and test the new share

It should work. Why? (*Different default set of Share permissions, perhaps*)

#### PowerShell sharing

Log on to **SERVER1** with a user with administrative privileges and open a **PowerShell** terminal

Create a folder

**New-Item -Name PSShare -Type Directory -Path D:\**

Share the folder with full access to domain users

**New-SmbShare -Name PSShare -Path D:\PSShare -FullAccess "WSAA\Domain Users"**

Get the list of existing shares on **SERVER1** with

**Get-SmbShare**

Note the number of shares including those with the $ symbol at the end

Examine the permissions of a share with

**Get-SmbShareAccess -Name PSShare**

Compare them with the other two shares

Log on to **SERVER2** with a regular user to check if you can see and access the share

It should work. Why? (*Perhaps, due to the explicitly set Share permissions*)

### NFS Sharing

#### Preparation

Log on to **SERVER1** with an account with administrative privileges

Start **Server Manager** and go to **Manage** and click on **Add Roles and Features**

On the first screen click **Next**

Make sure that **Role-based or feature-based installation** is selected and click **Next**

Ensure that **SERVER1** is selected and click **Next**

Expand the **File and Storage Services** node

Select the **Server for NFS** option

On the dialog screen for the required features click **Add Features**

Click on **Next**

On the features screen select the **Client for NFS** option as well and click **Next**

On the confirm installation screen click **Install**

When the installation is complete, click on **Close**

Return to **Server Manager** and click on **Services for Network File System** under **Tools**

Select the **Services for NFS** node and from the context menu chose **Properties**

Check the **Active Directory domain name** and in the text field enter the domain name (for example **WSAA.LAB**)

Click on **OK** to close the properties windows and then close the tool

#### NFS share

Open **File Explorer** and navigate to disk **D:**

Create a folder **NFSData**

Invoke the context menu of the folder and choose **Properties**

Switch to **NFS Sharing** tab

Click on **Manage NFS Sharing** button

Check the **Share this folder**

Click on the **Permissions** button

Examine the available permissions and click on **OK** to close the window

Click **OK** to close the **NFS Advanced Sharing** window

Click on **Close** to close the **NFSData Properties** window

#### PowerShell

Open a **PowerShell** terminal session

To get a list of available NFS shares execute

**Get-NfsShare**

Let’s create a folder and share it via NFS

**New-Item -Name NFSShare -Type Directory -Path D:\**

**New-NfsShare -Name NFSShare -Path D:\NFSShare**

To list detailed information for a share, execute

**Get-NfsShare -Name NFSShare | Select \***

Compare the permissions of both shares

Try to mount one of the two on **SERVER2** with command like (in CMD terminal session):

**mount SERVER1:/NFSData Z:**

Don’t forget first to install the **Client for NFS** feature

## Part 2: Storage 102

We can continue with the infrastructure from the first part or create a new one with same characteristics

### DFS Installation

Log on to **SERVER1** with an account with administrative privileges

Start **Server Manager** and go to **Manage** and click on **Add Roles and Features**

On the first screen click **Next**

Make sure that **Role-based or feature-based installation** is selected and click **Next**

Ensure that **SERVER1** is selected and click **Next**

Expand the **File and Storage Services** node

Select the **DFS Namespaces** and **DFS Replication** options

When asked, click on **Add Features**

Click **Next**

On the features screen click on **Next**

Start the installation by clicking on **Install**

When the installation is complete, click on **Close**

Repeat the same steps on **SERVER2** as well but this time using PowerShell, execute:

**Install-WindowsFeature FS-DFS-Namespace, FS-DFS-Replication -IncludeManagementTools**

### Preparation

Log on to **SERVER1** with an account with administrative privileges

Open **File Explorer** and create two folders – **D:\Share01** and **D:\Share02**

Invoke the context menu of **D:\Share01** and click on **Properties**

Switch to the **Sharing** tab and click on **Advanced Sharing** button

Check the **Share this folder** option

Click on **Permissions** button

Grant **Full Control** to **Everyone** and close the window with **OK**

Click on **OK** to close the **Advanced Sharing** window

Click on **Close** to close the **Share01 Properties** window

Do the same with folder **D:\Share02**

Repeat the same procedure for folders and sharing on **SERVER2**

### DFS Namespaces

Return on **SERVER1**

Open **Server Manager** and navigate to **DFS Management** under **Tools**

Select **Namespaces** in the left panel

Invoke the context menu and select **New Namespace**

In the **Server** text field enter **SERVER1** and click **Next**

In the **Name** field enter a name for the **DFS Namespace**, for example **Shares**

Click on **Edit Settings** button

Change the **Shared folder permission** to **All users have read and write permissions** and click **OK**

Click **Next**

Verify that **Domain-based namespace** option is selected and click **Next**

Check the summary information and click **Create**

Once the namespace is created click **Close**

Being in the **DFS Management** tool, expand the **Namespaces** node

Select the namespace we created earlier

Click on **New Folder** item either in the **Actions** menu (in the right section) or in the context menu

Enter **SERVER1 Share01** in the **Name** field

Click the **Add** button and enter [**\\SERVER1\Share01**](file:///\\SERVER1\Share01) in the **Path to folder target** field and click **OK**

Click **OK** to close the **New Folder** window

Repeat the procedure for [\\SERVER1\Share02](file:///\\SERVER1\Share02) , [\\SERVER2\Share01](file:///\\SERVER2\Share01) and [\\SERVER2\Share02](file:///\\SERVER2\Share02)

Open a **File Explorer** and in the address field enter [**\\WSAA.LAB\Shares**](file:///\\WSAA.LAB\Shares)

You should see all four shares in one place

Return to the **DFS Management** tool

Select the namespace and invoke its context menu

Click on **Properties** and then switch to the **Advanced** tab

Check the **Enable access-based enumeration for this namespace** and click **OK**

### DFS Replication

Log on to **SERVER1** with an account with administrative privileges

Open **File Explorer** and create a folder **D:\Share03**

Share the folder with **Full Control** share permissions granted to **Everyone**

Repeat the same procedure on **SERVER2**

Return to **SERVER1** and go to **DFS Management** console

Select the **Replication** option in the left panel

Click on **New Replication Group** item either in the **Actions** menu (in the right section) or in the context menu

Accept the default settings on the first screen and click **Next**

Enter **Replication1** in the **Name of the replication group** field and click **Next**

Click **Add** to add member servers

In the **Enter the object names to select** enter **SERVER1; SERVER2** and click on **Check Names** and then click **OK**

If everything went fine, we should see both servers added as members. Click **Next**

Accept the offered **Full mesh** topology and click **Next**

On the **Replication Group Schedule and Bandwidth** screen accept the defaults and click **Next**

Select **SERVER1** as primary server and click **Next**

On the **Folders to Replicate** screen click **Add** button

Enter **D:\Share03** in the **Local path of folder to replicate** and click **OK**

Click **Next**

Click **Edit** on the next screen

Change **Membership status** to **Enabled** and enter **D:\Share03** in the **Local path or folder** field. Click **OK**

Click **Next**

Review the settings and click on **Create**

Once the replication group is created, click on **Close**

If a **Replication Delay** message appears, click **OK**

Return to the **DFS Management** console

Expand the **Replication** node and select **Replication1** to see how the things are going

Open a **File Explorer** and navigate to folder **D:\Share03**

Create a file **Readme.txt**

Enter something there and save and close the file

Go to **SERVER2** and check if the file is there

If not, you can wait a while or go and force the replication from **SERVER1** by executing **Replicate Now**

From here, we can publish the replicated folder to a namespace

### Storage Spaces and Pools

Add two more **10GB** **dynamically expanding** disks to one of the member servers, for example **SERVER1**

#### Pool creation

Open **Server Manager** and navigate to **File and Storage Services** option in the left panel

Click on **Storage Pools**

Click on **New Storage Pool** in the **Tasks** drop-down

On the first screen click **Next**

Enter **Pool1** in the **Name** fieldand click **Next**

Select both newly added disks (10 GB in size) and click **Next**

On the **Confirm selections** screen click **Create**

Once the pool is created, click on **Close**

#### Virtual disks

The pool will appear in the list of available pools. Select it

Click on **New Virtual Disk** option under **Tasks** in the **Virtual Disks** section

Select the **Pool1** pool and click **OK**

Click **Next** on the **Before you begin** window

Enter **DataDisk1** in the **Name** field and click on **Next**

On the **Enclosure Awareness** screen click **Next**

Select **Parity** and click **Next**

A warning appears. Why? (*We need at least one more disk, perhaps*)

Change the selection to **Mirror** and click **Next**

On the **Provisioning** screen select **Fixed** and click **Next**

On the next screen we can enter a desired size. Let’s select the **Maximum size** option and click **Next**

Explore the information provided on the **Confirmation** screen and click **Create**

Click **Close** once the creation process finished

#### Volumes

**New Volume Wizard** appears immediately

Click **Next** on the welcome screen

On the **Select Server and Disk** window there is not much to be done as we have one server with one virtual disk

Click **Next**

For **Volume size** enter **5GB** and click **Next**

Accept the proposed **Drive letter** and click **Next**

Change **Volume label** to **DATA1** and click **Next**

Examine the information on the **Confirmation** screen and click **Create**

Once the process is done, click **Close**

Open **File Explorer** to see if the drive is there

Check again in the **Disk Management** console

Try to extend the volume to the maximum possible size

### iSCSI

We can continue with the same infrastructure or start fresh

Let’s continue with the same but this time we will switch to **SERVER2**

#### Preparation 1

Log on to **SERVER2** with an account with administrative privileges

Start **Server Manager** and go to **Manage** and click on **Add Roles and Features**

On the first screen click **Next**

Make sure that **Role-based or feature-based installation** is selected and click **Next**

Ensure that **SERVER2** is selected and click **Next**

Expand the **File and Storage Services** node

Select the **iSCSI Target Server** option and click **Next**

On the features screen click on **Next**

Start the installation by clicking on **Install**

When the installation is complete, click on **Close**

#### Preparation 2

Create a new **private switch** (or use any other isolated virtual network type in your virtualization solution) named for example **iSCSI vSwitch**

Add one more network card to both virtual machines **WSAA-M1-VM-S1** (**SERVER1**) and **WSAA-M1-VM-S2** (**SERVER2**) and put them in an isolated network (the private switch created earlier)

Assign static IP addresses to the newly added interfaces, for example **SERVER1** with **192.168.77.101/24** and **SERVER2** with **192.168.77.102/24**

*Please note, that you do not need to add neither default gateway, nor any DNS servers*

#### iSCSI Initiator (1-st part)

Return to **SERVER1** which will play the client role (also known as the initiator) in our case

Open **Server Manager** and click on **iSCSI Initiator** under **Tools**

Click **Yes** to confirm that you want to start the service

Switch to **Discovery** tab

Click on the **Discover Portal** button

In the **IP address or DNS name** field enter the IP address of our future target – **SERVER2 – 192.168.77.102**

Click on **Advanced** button

Change **Local adapter** to **Microsoft iSCSI Initiator**

Select **192.168.77.101** in the **Initiator IP** drop-down list

Click **OK** to close the **Advanced Settings** window

Click **OK** to close the **Discovery Target Portal** window

#### iSCSI Target

Switch to **SERVER2**

Open the **Server Manager** and navigate to **File and Storage Services**

Then select **iSCSI**

Click on **New iSCSI Virtual Disk** option under the **Tasks** drop-down list

Select one of the available disks, for example **E:** and click **Next**

Enter **iSCSIDisk1** in the **Name** field and click **Next**

Set **5GB** for size and make sure that the **Dynamically expanding** option is selected. Click **Next**

Because we do not have any existing iSCSI targets, the **New iSCSI target** option is selected. Click **Next**

Enter **RemoteDisk** in the **Name** field and click **Next**

On the **Access Servers** screen click on the **Add** button

Select the **Select from the initiator cache on the target server** option and mark the record there. Click **OK**

Click **Next**

Do not enable authentication, instead click **Next**

Examine the information on the **Confirmation** screen and click **Create**

Once the process is done, click on **Close**

#### iSCSI Initiator (2-nd part)

Return to **SERVER1**

If the **iSCSI Initiator** tool is closed, reopen it

Go to the **Targets** tab and click on the **Refresh** button

One **inactive** record will appear. Select it and click on **Connect**

Then click on **Advanced** and select the correct adapter (**Microsoft iSCSI Initiator**), initiator IP (**192.168.77.101**) and target IP (**192.168.77.102**)

Click on **OK** to close the **Advanced Settings** window

Click **OK** to close the **Connect To Target** window

The status will change to **Connected**

Switch to **Favorite Targets** tab, we should see our target there

Switch to **Volumes and Devices** tab and click on **Auto Configure**

Click **OK** to close the tool

#### Utilize the disk

Open **Disk Management** and follow the standard steps you usually do to make a disk usable

## Part 3: Storage 103

We can continue with the infrastructure from the previous part or create a new one with same characteristics

### Tiered Storage

We will use **SERVER1** again

First, we must add **two disks 10GB** each and another **two – 30 GB each**. All (four) must be **dynamic**

Log on to **SERVER1** with an account with administrative privileges

Open a **PowerShell** session with **Run as Administrator**

Check what disks are available for use in pool with

**Get-PhysicalDisk | Where CanPool -Eq True | Select UniqueId, MediaType, Size**

Two of the disks are 10GB in size, and we want them to be treated as SSD drives, so we will execute

**Set-PhysicalDisk -UniqueId <drive-id> -MediaType SSD**

This must be executed for both 10GB disks

We can mark the other two as regular HDDs with

**Set-PhysicalDisk -UniqueId <drive-id> -MediaType HDD**

Now, let’s return to the **Server Manager**

Navigate to **File and Storage Services** and then to **Storage Pools**

Select the **Primordial** pool and in the **Physical Disks** section you will see our four disks

*Please note, that you haven’t restarted the machine since the previous disk manipulations, you may see more disks here. So, you should either restart, or be very careful what you select*

Click on **New Storage Pool** in the **Task** menu

Click on **Next** on the welcome screen

Enter **TieredPool** in the **Name** field and click **Next**

Select the four disks (two SSDs and two HDDs) and click **Next**

Finally click **Create**

Once the creation process finishes, click **Close**

Select the newly created pool and right-click on it and select **New Virtual Disk**

Make sure that the **TieredPool** is still selected and click **OK**

Click on **Next** on the welcome screen

Enter **TieredDisk** in the **Name** field, select the **Create storage tiers on this virtual disk** option and click **Next**

On the **Enclosure Awareness** screen click **Next**

On the **Storage Layout** screen select for example **Mirror** and click **Next**

When using storage tiers, we must use **fixed** provisioning, so nothing to do here. Click **Next**

We can specify the proportions. Let’s set 3GB for the faster tier and 10GB for the standard. Click **Next**

Explore the information on the **Confirmation** screen and click **Create**

Once the process finishes, we can click the **Close** button

Next, the **New Volume Wizard** starts

Follow the steps to create a volume that utilizes all the available space

Once the process finishes, click the **Close** button

Now, you can use the disk as every other disk with the exception that you can pin files to either of the tiers

Check the available commands with

**Get-Command \*FileStorageTier\***

### Deduplication

We can use either of both servers. Let’s switch to **SERVER2** again

First, we must install the **Data Deduplication** role service

#### Preparation

Start **Server Manager** and go to **Manage** and click on **Add Roles and Features**

On the first screen click **Next**

Make sure that **Role-based or feature-based installation** is selected and click **Next**

Ensure that **SERVER2** is selected and click **Next**

Expand the **File and Storage Services** node

Select the **Data Deduplication** option and click **Next**

On the features screen click on **Next**

Start the installation by clicking on **Install**

When the installation is complete, click on **Close**

#### Evaluation and configuration

Pick up a file (something bigger than 32 KB, for example from the **C:\Windows\Web\Wallpaper** folder) and paste it several times in the **D:\** folder

Then, open command-line session with **Run as Administrator**

To check what will be the potential saving if data deduplication is enabled, execute

**ddpeval D: /V**

You will see a report

Let’s assume that it is worth it to enable deduplication

Open **Server Manager** and navigate to **File and Storage Services** and then **Volumes**

Select the **D:** volume and choose **Configure Data Deduplication** in its context menu

Set the **Data deduplication** mode to **General purpose file server**

Leave the other options as they are and click **OK** to save the changes

You can see that two values appeared against the selected disk – **deduplication rate** and **deduplication savings**

Absolutely the same could have been achieved under **PowerShell** with this command

**Enable-DedupVolume -Volume D: -UsageType Default**

We can check the deduplication status with

**Get-DedupStatus**

Or extend the command to receive more detailed information

**Get-DedupStatus | FL**

Check the deduplication schedule with

**Get-DedupSchedule**

You can force a deduplication job to take place with

**Start-DedupJob -Type Optimization -Volume D: -Priority High**

And then, if you are quick, you can check its status with

**Get-DedupJob -Type Optimization -Volume D:**

Check again for any savings with either

**Get-DedupStatus | FL**

Or

**Get-DedupVolume -Volume D: | FL**

In our case, we won’t see any difference as the data volume is too small and the timestamp of the files is recent

### Encrypt files with EFS

Again, we can choose either **SERVER1** or **SERVER2**

Let’s continue with **SERVER1**. Log on with an account with administrative privileges

Open **File Explorer** and create a folder **D:\Data**

Create a **Readme.txt** file there with some text, for example your name. Save and close the file

Invoke the context menu of the folder

On the **General** tab click **Advanced** button

Click the **Encrypt contents to secure data**. Try to select the **Compress contents to save disk space**. It is not possible

Make sure that the encryption option is selected and click **OK** to close the **Advanced Attributes** window

Click **OK** to close the folder’s properties dialog box

When asked to confirm, do it

Now the file appears in a different way

Log off and log back on with the regular user from the previous parts

Navigate to the file and try to open it. You shouldn’t succeed (*Perhaps, because the file was encrypted by and for another user*)

Go to the DC and create another account member of the **Domain Admins** group, if there isn’t one already, and try with it. Any success? Why? (*Perhaps, because the file was encrypted by and for another user*)

While still with the second administrator, try to disable the encryption. Still no luck? Why? (*Perhaps, because the file was encrypted by and for another user*)

Log on back with the initial user that created the file and navigate to it

Invoke the **Properties** screen and click on **Advanced** button

Then click on **Details** to see the certificate. It is **important** to keep a **backup copy** of it

You can back up the certificate with the help of **Certificates** (**certmgr.msc**)

Try to remove the encryption of the folder and its files

*Depending on how you did it (the folder itself or within the folder), you will see an error. Ignore it and at the end the files will become unencrypted, but the folder may remain with its encryption turned on*

*You must invoke the context menu and select* ***File ownership*** *and then* ***Personal***

### Encrypt volumes with BitLocker

Log on back to **SERVER1**

Start **Server Manager** and go to **Manage** and click on **Add Roles and Features**

On the first screen click **Next**

Make sure that **Role-based or feature-based installation** is selected and click **Next**

Ensure that **SERVER1** is selected and click **Next**

On the **Server Roles** screen click **Next**

On the **Features** screen select **BitLocker Drive Encryption**

Confirm that you agree additional features to be installed by clicking on **Add Features**

Then click on **Next**

Start the installation by clicking on **Install**

When the installation is complete, click on **Close**

You will see a notification that the server must be restarted. Restart it

Once the server is up, log on again

Select disk **D:** and invoke its context menu. There select the **Turn on BitLocker** option

On the **Choose how you want to unlock this drive** screen select **Use a password to unlock the drive**

Enter the desired password twice and click **Next**

On the next screen choose a preferred key backup method. Let’s select **Save to a file** and select another drive and confirm with **Save**

Click **Next**

Select the **Encrypt entire drive** option and click **Next**

Depending on the drive and its usage scenario (if it a removable one and used on different Windows versions), select an **encryption mode**. Let’s select the **New encryption mode** and click **Next**

Finally, click on **Start encrypting**

The encryption process should finish very quickly

Anyway, we can always use **PowerShell** to check what is going on. Execute this command

**Get-BitLockerVolume**

You will see all volumes with their encryption status