# Solution M1: Storage Solutions

There are multiple ways to achieve what was requested

As the visual approach is intuitive, the presented solution is implemented with PowerShell

Please note that the solution is far from being optimal. The aim is to illustrate all (or most) individual steps

## Conditions

Next steps are executed under the following conditions (adjust them according to your situation):

* Virtual machines are stored in **C:\VM**
* Source (template) virtual hard disk is **C:\BAK\WIN-SRV-2K19-ST-DE.vhdx**
* One existing machine (**domain controller**)
* Both machines will be connected to a common network defined by an existing NAT switch (**NAT vSwitch**)
* Second pair of network adapters will be connected to an existing private network switch (**vPrivate**)

You may use the **M1-Solution-Script-Storage-Solutions-1-Preparation.ps1** file to prepare the environment

## Possible Solution

Open a **PowerShell** session with **Run as Administrator** on the **Hyper-V** host

Clone an existing virtual hard disk by creating a differencing child disk

**New-VHD -Path ($TargetFolder + $VM1 + ".vhdx") -ParentPath $SourceVHD -Differencing**

Create a new virtual machine

**New-VM -Name $VM1 -MemoryStartupBytes 2048mb -VHDPath ($TargetFolder + $VM1 + ".vhdx") -Generation 2 -SwitchName "NAT vSwitch" | Set-VM -CheckpointType Production -AutomaticCheckpointsEnabled $false -PassThru | Set-VMMemory -DynamicMemoryEnabled $false**

Create second hard disk

**New-VHD -Path ($TargetFolder + $VM1 + "-DISK1.vhdx") -SizeBytes 10gb -Dynamic**

Attach the second hard disk to the VM

**Add-VMHardDiskDrive -VMName $VM1 -Path ($TargetFolder + $VM1 + "-DISK1.vhdx")**

Create third hard disk

**New-VHD -Path ($TargetFolder + $VM1 + "-DISK2.vhdx")-SizeBytes 10gb -Dynamic**

Attach the third hard disk to the VM

**Add-VMHardDiskDrive -VMName $VM1 -Path ($TargetFolder + $VM1 + "-DISK2.vhdx")**

Create fourth hard disk

**New-VHD -Path ($TargetFolder + $VM1 + "-DISK3.vhdx")-SizeBytes 20gb -Dynamic**

Attach the fourth hard disk to the VM

**Add-VMHardDiskDrive -VMName $VM1 -Path ($TargetFolder + $VM1 + "-DISK3.vhdx")**

Create fifth hard disk

**New-VHD -Path ($TargetFolder + $VM1 + "-DISK4.vhdx") -SizeBytes 20gb -Dynamic**

Attach the fifth hard disk to the VM

**Add-VMHardDiskDrive -VMName $VM1 -Path ($TargetFolder + $VM1 + "-DISK4.vhdx")**

Power on the VM

**Start-VM -VMName $VM1**

Wait the machine to boot and then change its IP address

**Invoke-Command -VMName $VM1 -Credential $LC -ScriptBlock { New-NetIPAddress -InterfaceAlias "Ethernet" -IPAddress "192.168.99.100" -PrefixLength 24 -DefaultGateway 192.168.99.1 ; Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses 192.168.99.2 }**

Then rename it

**Invoke-Command -VMName $VM1 -Credential $LC -ScriptBlock { Rename-Computer -NewName HW -Restart }**

Wait the machine to boot and join it to the existing domain

**Invoke-Command -VMName $VM1 -Credential $LC -ScriptBlock { Add-Computer -DomainName $args[0] -Credential $args[1] -Restart } -ArgumentList $Domain, $DC**

Wait the machine to boot and add second network adapter to it (HW-M1/HWM2)

**Add-VMNetworkAdapter -VMName $VM1 -SwitchName "vPrivate"**

Set the IP address of the second network adapter

**Invoke-Command -VMName $VM1 -Credential $DC -ScriptBlock { New-NetIPAddress -InterfaceAlias "Ethernet 2" -IPAddress "192.168.67.100" -PrefixLength 24 }**

Add second network adapter to the first (existing) virtual machine (HW-DC/EM1)

**Add-VMNetworkAdapter -VMName $VM0 -SwitchName "vPrivate"**

Set the IP address of the second network adapter

**Invoke-Command -VMName $VM0 -Credential $DC -ScriptBlock { New-NetIPAddress -InterfaceAlias "Ethernet 2" -IPAddress "192.168.67.2" -PrefixLength 24 }**

Switch the session to the second (homework) VM

**Enter-PSSession -VMName $VM1 -Credential $DC**

Set disk type of the two 10GB disks to SSD

**Get-PhysicalDisk | Where DeviceID -In -Value 1,2 | Set-PhysicalDisk -MediaType SSD**

Set disk type of the two 20GB disks to HDD

**Get-PhysicalDisk | Where DeviceID -In -Value 3,4 | Set-PhysicalDisk -MediaType HDD**

Create a storage pool

**$PD = (Get-PhysicalDisk -CanPool $true)**

**New-StoragePool -FriendlyName HomeworkPool -StorageSubsystemFriendlyName "Windows Storage\*" -PhysicalDisks $PD -Verbose**

Define SSD tier

**New-StorageTier -StoragePoolFriendlyName "HomeworkPool" -FriendlyName "HDDTier" -MediaType HDD -Verbose**

Define HDD tier

**New-StorageTier -StoragePoolFriendlyName "HomeworkPool" -FriendlyName "HDDTier" -MediaType HDD -Verbose**

Create a virtual hard drive

**New-Volume -StoragePoolFriendlyName "HomeworkPool" -FriendlyName "HomeworkDisk" -AccessPath "X:" -ResiliencySettingName "Mirror" -ProvisioningType "Fixed" -StorageTiers (Get-StorageTier -FriendlyName "\*SSD\*"), (Get-StorageTier -FriendlyName "\*HDD\*") -StorageTierSizes 6gb, 16gb -FileSystem NTFS -AllocationUnitSize 64KB**

Install the iSCSI Target role component

**Install-WindowsFeature FS-iSCSITarget-Server**

Create an iSCSI target

**New-IscsiServerTarget -TargetName "homework" -InitiatorId @("IPAddress:192.168.67.2")**

Create an iSCSI virtual hard disk

**New-IscsiVirtualDisk -Path "X:\homework-iscsi-disk.vhdx" -Size 10GB**

Attach the iSCSI virtual hard disk to the iSCSI target

**Add-IscsiVirtualDiskTargetMapping -TargetName "homework" -DevicePath "X:\homework-iscsi-disk.vhdx"**

Close the PowerShell session and return to the Hyper-V host

**exit**

Establish PowerShell session to the other machine (DC in our case)

**Enter-PSSession -VMName $VM0 -Credential $DC**

Start the iSCSI initiator service

**Start-Service msiscsi**

Set service start up type to automatic

**Set-Service msiscsi -StartupType Automatic**

Create a new iSCSI target portal

**New-IscsiTargetPortal -TargetPortalAddress "192.168.67.100" -InitiatorPortalAddress "192.168.67.2" -InitiatorInstanceName "ROOT\ISCSIPRT\0000\_0"**

Connect to an iSCSI target

**$TARGET=Get-IscsiTarget**

**Connect-IscsiTarget -NodeAddress $TARGET.NodeAddress -TargetPortalAddress "192.168.67.100" -InitiatorPortalAddress "192.168.67.2" -IsPersistent $true**

Initialize and format the disk

**Initialize-Disk -Number 1 -PartitionStyle GPT**

**New-Volume -DiskNumber 1 -FriendlyName "iSCSIDisk" -FileSystem NTFS -DriveLetter S**

Create new folder

**New-Item -ItemType Directory -Path "S:\Shared Data"**

Set NTFS permissions on the folder

**$ACL = Get-Acl -Path "S:\Shared Data"**

**$AR = New-Object System.Security.AccessControl.FileSystemAccessRule("WSAA\Domain Users", "Read", "Allow")**

**$ACL.SetAccessRule($AR)**

**$AR = New-Object System.Security.AccessControl.FileSystemAccessRule("WSAA\Domain Admins", "FullControl", "Allow")**

**$ACL.SetAccessRule($AR)**

**$ACL | Set-Acl -Path "S:\Shared Data"**

Share the folder

**New-SmbShare -Name "Shared" -Path "S:\Shared Data" -FullAccess Everyone**

We are done. The challenge has been solved

*Please note that the above steps are provided as a separate file (M1-Solution-Script-Storage-Solutions-2-Actual.ps1)*