# The Big Ass Virtualisation help me daddy cheat sheet

## Inhoud

Voor enter-pssession eerst dit → “Enable-PSRemoting -Force”

## Setup Domain

Zorg eerst voor een static IP! Manueel of via sconfig

```
Install-WindowsFeature -Name AD-Domain-Services

$password = ConvertTo-SecureString "Pxl-2021" -AsPlainText -Force

Install-ADDSForest -CreateDnsDelegation:$false `
                   -DatabasePath 'C:\Windows\NTDS' `
                   -DomainMode WinThreshold `
                   -DomainName 'company.pri' `
                   -DomainNetbiosName 'COMPANY' `
                   -ForestMode WinThreshold `
                   -InstallDns:$true `
                   -LogPath 'C:\Windows\NTDS' `
                   -NoRebootOnCompletion:$false `
                   -SafeModeAdministratorPassword $password `
                   -SysvolPath 'C:\Windows\SYSVOL' `
                   -Force:$true

Add-WindowsCapability -Online -Name Rsat.ActiveDirectory.DS-LDS.Tools~~~~0.0.1.0
Add-WindowsFeature RSAT-Role-Tools
```

## Setup Hyper-V

Op server:

```
Add-WindowsFeature -Name Hyper-V, Hyper-V-Tools, Hyper-V-PowerShell

(reboot)

Op client: run als admin


Enable-WindowsOptionalFeature -Online -FeatureName:Microsoft-Hyper-V -All
```

## Set ip address

Met sconfig of

```
Get-NetAdapter #selecteer de interface waar je het ipadres wilt aanpassen
New-NetIPAddress -InterfaceIndex "5" -IPAddress 192.168.3.20 -PrefixLength 24 -
DefaultGateway 192.168.3.2
Set-DnsClientServerAddress -InterfaceIndex "5" -ServerAddresses "192.168.3.10 ,
192.168.3.2 , 8.8.8.8"
```

Ip address verwijderen

```
Remove-NetIPAddress -IPAddress 192.168.3.20
```

## Set hostname

Via sconfig of

```
Rename-Computer -NewName "Hyperv2019" -Restart
```

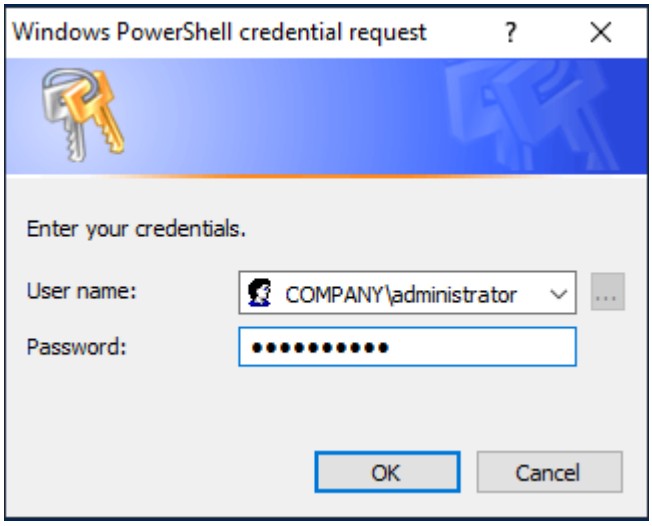
## Let computer join domain

Zorg er voor dat het domain bereikbaar is = je kan er naar pingen.

Doe dan op de computer die in het domain gezet moet worden:

```
Add-Computer -DomainName company.pri -Credential COMPANY\administrator -Restart -Force
```

Geef het wachtwoord in en druk enter




Meerdere host tergelijk joinen:

```
Add-Computer -ComputerName Hyperv2019, Mydesktop -DomainName company.pri –credential
COMPANY\administrator -restart –force
```

## Create VM with powershell

```
New-VM -Name TESTMACHINE `
        -MemoryStartupBytes 1GB `
        -NewVHDSizeBytes 25GB `
        -NewVHDPath C:\TESTMACHINE.VHDX


New-VM -Name TESTMACHINE02 `
        -MemoryStartupBytes 1GB `
        -NewVHDSizeBytes 20GB `
        -NewVHDPath C:\TESTMACHINE02.VHDX `
        -Generation 2

Set-VMMemory TESTMACHINE02 -DynamicMemoryEnabled $false
```

## Add processor to VM

```
Set-Vm -Name myfirstvm -ProcessorCount 2
```

#voor  live migratie van 1vm naar andere

```
Set-VMProcessor myfirstvm -CompatibilityForMigrationEnabled $true
```

## Integration services

Zet ze allemaal uit

```
Disable-VMIntegrationService * -VMName myfirstvm
```

```
Allemaal aanzetten
```

```
Enable-VMIntegrationService * -VMName myfirstvm
```

```
Check of ze nog aanstaan
```

```
Get-VMIntegrationService -VMName myfirstvm
```

```
1 service uitzetten
```

```
Disable-VMIntegrationService -Name 'Guest Service Interface' myfirstvm
```

## Copy-VMFile

Bestanden van host naar vm

```
Copy-VMFile myfirstvm -SourcePath C:\hey.txt -DestinationPath C:\temp\hey.txt -
CreateFullPath -FileSource host
```

## Check VM locatie via powershell

Host settings:

```
Get-VMHost | select VirtualHardDiskPath,VirtualMachinePath
Get-VMHost | select *path
```

```
VM settings:
```

```
Get-VM | select Name,ConfigurationLocation,SnapshotFileLocation,SmartPagingFilePath
Get-VM | select Name,ConfigurationLocation,SnapshotFileLocation,SmartPagingFilePath |
fl

Get-VM | Get-VMHardDiskDrive | Select VMName,Path
```

## Convert HVD ⇔ VHDX

```
New-VHD -Path C:\Test\ConversionTest.vhd -SizeBytes 2GB
Convert-VHD -Path C:\Test\ConversionTest.vhdx
            -DestinationPath C:\Test\ConversionTest.vhd -DeleteSource
```

## Create disk partition

```
Mount-VHD -Path C:\Test\ConversionTest.vhdx
Get-Disk -Number 1 | Initialize-Disk -PassThru
New-Partition -DiskNumber 1 -Size 4GB
```

## Attach disk to VM

```
New-VHD -SizeBytes 20G `
        -Path 'C:\Users\Public\Documents\Hyper-V\Virtual hard disks\seconddisk.vhdx'

Add-VMHardDiskDrive -VMName myfirstvm `
                    -Path 'C:\Users\Public\Documents\Hyper-V\Virtual hard
disks\seconddisk.vhdx' `
                    -ControllerType IDE -ControllerNumber 0 `
                    -ControllerLocation 1
```

## Expand disk

```
Resize-VHD -Path C:\Test\ConversionTest.vhdx -SizeBytes 10GB
```

## Shrink disk

```
Mount-VHD -Path C:\Test\ConversionTest.vhdx
Get-Disk -Number 1 | Initialize-Disk -PassThru
New-Partition -DiskNumber 1 -Size 4GB


Resize-VHD -Path C:\Test\ConversionTest.vhdx -ToMinimumSize
Get-VHD -Path C:\Test\ConversionTest.vhdx | select size
Dismount-VHD -Path C:\Test\ConversionTest.vhdx
```

## Optimize disk

```
Mount-VHD -Path C:\test\ConversionTest.vhdx -ReadOnly
Optimize-VHD -Path C:\test\ConversionTest.vhdx -Mode Full
Dismount-VHD -Path C:\test\ConversionTest.vhdx
```

## Checkpoints

```
#list commands for checkpoints
Get-Command -Module Hyper-V -Name *checkpoint*
Get-Command -Module Hyper-V -Name *snapshot*

#list vm's on host
```

```
Get-VM

#create checkpoint
Checkpoint-VM -Name myfirstvm -SnapshotName 'Pre-software install'

#list all checkpoints of specific vm
Get-VMSnapshot -VMName myfirstvm

#remove all checkpoints
Remove-VMSnapshot -Name * -VMName myfirstvm
```

## Networking

#external switch

```
New-VMSwitch -Name 'SwitchName' -NetAdapterName 'Ethernet0'

#internal / private switch
New-VMSwitch -Name 'SwitchName'-SwitchType Internal / Private

#connect vm to switch
Connect-VMNetworkAdapter -VMName myfirstvm -Name 'Network Adapter' -SwitchName
'SwitchName'
#nat switch maken op host
New-VMSwitch -ComputerName hyperv2019 -SwitchType Internal -SwitchName NAT

Get-NetAdapter # check interface of nat
New-NetIPAddress -InterfaceIndex 16 -IPAddress 192.168.4.1 -PrefixLength 24
New-NetNat -Name NAT -InternalIPInterfaceAddressPrefix "192.168.4.0/24"


#portforwarding nat
Add-NetNatStaticMapping -NatName NAT -Protocol TCP `
                        -ExternalIPAddress 0.0.0.0 -InternalIPAddress 192.168.4.10 `
                        -InternalPort 80 -ExternalPort 80
```

#delete vmSwitches

```
Remove-VMSwitch * -Force
```

### NIC teaming

```
New-NetLbfoTeam -Name "Team1" -TeamMembers "Ethernet0","Ethernet1"
#remove nic team
Remove-NetLbfoTeam -Name "Team1"


#create nic team vswitches
New-VMSwitch -Name "switch1" -NetAdapterName Ethernet0
New-VMSwitch -Name "switch2" -NetAdapterName Ethernet1

New-NetSwitchTeam -Name "Switchteam" -TeamMembers "vEthernet (switch1)","vEthernet
(switch2)"

#rename network
Get-NetAdapter -Name Ethernet0 | Rename-NetAdapter -NewName 'LAN'
Get-NetAdapter -Name Ethernet1 | Rename-NetAdapter -NewName 'Storage'
Get-NetAdapter -Name Ethernet2 | Rename-NetAdapter -NewName 'HeartBeat'

Enable ping

netsh advfirewall firewall add rule name="ICMP Allow incoming V4 echo
request" protocol="icmpv4:8,any" dir=in action=allow
```

## Storage

Vmx tool powershell

```
Install-Module vmxtoolkit


$VMX = Get-VMX -Path 'C:\vms\HyperV2019'
$VMX | Stop-VMX -Mode Soft
$VMX | New-VMXScsiDisk -NewDiskSize 100GB -NewDiskname DISV
$VMX | Set-VMXScsiController -SCSIController 3 -Type pvscsi
$VMX | Add-VMXScsiDisk -Diskname 'C:\vms\HyperV2019\DISV.VMDK' -LUN 0 -Controller 3


Create iscsi disk with controller

$VMX = get-vmx -Path 'C:\Virtualization les\VM\Windows 2019 W19HVC0X'
$VMX | Stop-VMX -Mode Soft
$VMX | New-VMXScsiDisk -NewDiskSize 1TB -NewDiskname DISK1
$VMX | Set-VMXScsiController -SCSIController 0 -Type PVSCSI
$VMX | ADD-VMXScsiDisk -Diskname DISK1.VMDK -LUN 0 -Controller 0
$VMX | New-VMXScsiDisk -NewDiskSize 1TB -NewDiskname DISK2
$VMX | Set-VMXScsiController -SCSIController 0 -Type PVSCSI
$VMX | ADD-VMXScsiDisk -Diskname DISK2.VMDK -LUN 1 -Controller 0
```

Pad opvragen van gestorede VM

```
Get-VM -Name myfirstvm | `
select Path,ConfigurationLocation,SnapshotFileLocation,SmartPaging
```

Move VM (live migration)

```
Move-VM LMTest TestServer02 -IncludeStorage -DestinationStoragePath D:\LMTest
# check if prosessor allows live migration
Set-VMProcessor TestVM -CompatibilityForMigrationEnabled $true
```

Install cluster & hyper v & fs-fileserver (reboot after install)

```
$vms = 'W19HVC01', 'W19HVC02'
foreach ($vm in $vms) {
$cred = Get-Credential "company\Administrator"
$s = New-PSSession -computername $vm -Credential $cred
$sb = {
$FeatureList = "Hyper-V", "Failover-Clustering", "FS-FileServer"
Install-WindowsFeature $FeatureList -IncludeManagementTools
}
Invoke-Command -Session $s -ScriptBlock $sb
## and clean up
Remove-PSSession -Session $s}
```

## Enable S2D

```
Enable-ClusterStorageSpacesDirect
```

Create volume in S2D

```
New-Volume -FriendlyName 'Vol1' -FileSystem CSVFS_NTFS `
-StoragePoolFriendlyName "S2D*" -Size 50GB
New-Volume -FriendlyName 'Vol2' -FileSystem CSVFS_NTFS `
-StoragePoolFriendlyName "S2D*" -Size 50GB
```

Aanmaken van image map op S2D

```
New-Volume -FriendlyName 'Images' -FileSystem CSVFS_NTFS `
-StoragePoolFriendlyName "S2D*" -Size 100GB
```

Create cluster witness (quorum)

Maak gedeelde map aan op dc share met everyone

Ga naar de failover cluster manager -> onder rechtse paneel “action” kies more actions & configure cluster quorum

Om virtual switch te hebben in de cluster manager moet je deze virtual switch aanmaken op één van de nodes in hyper v manager (best met script aanmaken)

Het script:

```
$vms = 'W19HVC01', 'W19HVC02'
foreach ($vm in $vms) {
$cred = Get-Credential "Administrator"
$s = New-PSSession -computername $vm -Credential $cred
$sb = {
New-VMSwitch -Name 'CLAN' -NetAdapterName 'LAN'
}
Invoke-Command -Session $s -ScriptBlock $sb
}
```

Connect vm met netadpter

```
Get-VM -Name CLTST01 | Get-VMNetworkAdapter | `
Connect-VMNetworkAdapter -SwitchName CLAN
```

Je kan wel een data disk toevoegen via cluster manager aan een vm, let wel op dat je deze disk opslaat in het S2D volume

Ititialize disk with powershell

```
Get-Disk |
where PartitionStyle -eq 'RAW' |
Initialize-Disk -PartitionStyle GPT -PassThru |
New-Partition -DriveLetter 'E' -UseMaximumSize |
Format-Volume -FileSystem NTFS -NewFileSystemLabel 'Data disk 01' `
-Confirm:$false
```

S2D uitbreiden? =>

Gewoon iscsi disk toevoegen via gui in vm ware (dan rebooten)

S2D zal de nieuwe disks mee opnemen in de storage pool

Script voor te kijken hoeveel vrije ruimte je hebt op je S2D disks:

```
Get-StoragePool -FriendlyName s2D* |
```

```
Get-PhysicalDisk |
foreach {
$node = $psitem | Get-StorageNode -PhysicallyConnected |
select -ExpandProperty Name
$size = [math]::Round( ($psitem.Size / 1GB), 2)
$free =
[math]::Round( ( ($psitem.Size - $psitem.VirtualDiskFootprint) / 1GB), 2)
$props = [ordered] @{
Node = ($node -split '\.')[0]
DeviceID = $psitem.DeviceId
Type = $psitem.MediaType
'Size(GB)' = $size
'Free(GB)' = $free
FreePerc = [math]::Round( ( ($free / $size) * 100 ), 2)
}
New-Object -TypeName PSobject -Property $props
} |
sort Node, DeviceID |
Format-Table
```

Syncen van S2D schijven (kost veel tijd!)

```
Optimize-StoragePool -FriendlyName "S2D on HVCL01"
```

Om het proces van de sync te volgen

```
Get-StorageJob | Format-List Name, ElapsedTime, `
JobState, PercentComplete, BytesProcessed, BytesTotal
```

Extend cluster shared volume

```
Get-VirtualDisk -FriendlyName Vol1 | Resize-VirtualDisk -Size 768GB
Get-VirtualDisk -FriendlyName Vol2 | Resize-VirtualDisk -Size 768GB
```

Je moet ook partitie vergroten!!

```
Get-Disk -FriendlyName Vol? |
foreach {
$part = Get-Disk -FriendlyName ($_.FriendlyName) |
Get-Partition | where Type -eq 'Basic'
$size = $part |
Get-PartitionSupportedSize |
select -ExpandProperty SizeMax
$part | Resize-Partition -Size $size
}
```

Change node that owns a disk

```
Move-ClusterSharedVolume -Name 'Cluster Virtual Disk (Vol1)' `
-Node W19HVC02
```

Create Migration network, zodat je het netwerk van clients niet overbelast wanneer je een vm gaat migreren

Voor elke node :Add network in vmware gui (vmnet 9 host only: 192.168.30.0/26= 255.255.255.192)

Op de cluster manger: rename network naar migration & set cluster only

Onder Network:

Select live migration settings > selecteer alleen het migration network.

Add node to the cluster manager

```
Add-ClusterNode -Name W19HVC03 -Type Node –NoStorage
```

Remove cluster node

```
Remove-ClusterNode -Name W19HVC03
```

## DEBUGGING stuff

Event viewer

```
Get-WinEvent -ListLog *Hyper-V*
Get-WinEvent -ListLog *failover*
```

Event bekijken:

```
Get-WinEvent -LogName Microsoft-Windows-Hyper-V-Hypervisor-Operational|
select -First 1 | Format-List *
```

performance monitor

de ingestelde waarde weergeven van de performance monitor in powershell$

```
Get-Counter
```

Show available space

```
Get-Volume
```

Via windows admin center > system insights

# MIGRATION

Upgraden naar nieuwere migration version is mogelijk!

Downgraden gaat niet! Dus bv van versie 9.0 naar 8.0 gaat niet

```
Get-VMHostSupportedVersion | sort Version –Descending
```

VM aanmaken in lager niveau

```
New-VM -Name 'TestConfigV' -Version 8.0
```

Update config version

```
Update-VMVersion -VMName <name of VM> -Confirm:$false
```

Update all vm's to latest version

```
Get-VM |`
where Version -lt (Get-VMHostSupportedVersion -Default).Version |`
Update-VMVersion -Confirm:$false -Passthru
```

## Cold migration:

Open hyper v manager > kies vm >sluit vm af> export > kies map om naar te exporteren

Als de vm geexporteerd is verplaats je de map naar de nieuwe server

In de hyper v manager > RMK op hyper v host > import virtual machine > register vm (1ste keuze)

IMPORT VM VIA PS

```
Import-VM -Path 'C:\VirtualMachines\CLTST01\Virtual Machines\F4BE1665-
50D0-47DE-8E46-3C08C16B2014.vmcx'
```

## Quick migration:

In cluster manager (mv mag aan staan) > selecteer vm> right panel > 'vm name' > move > quick mig > select node

## Live migration:

In cluster manager (vm moet aan staan) > selecteer vm> right panel > 'vm name' > move > live mig > select node

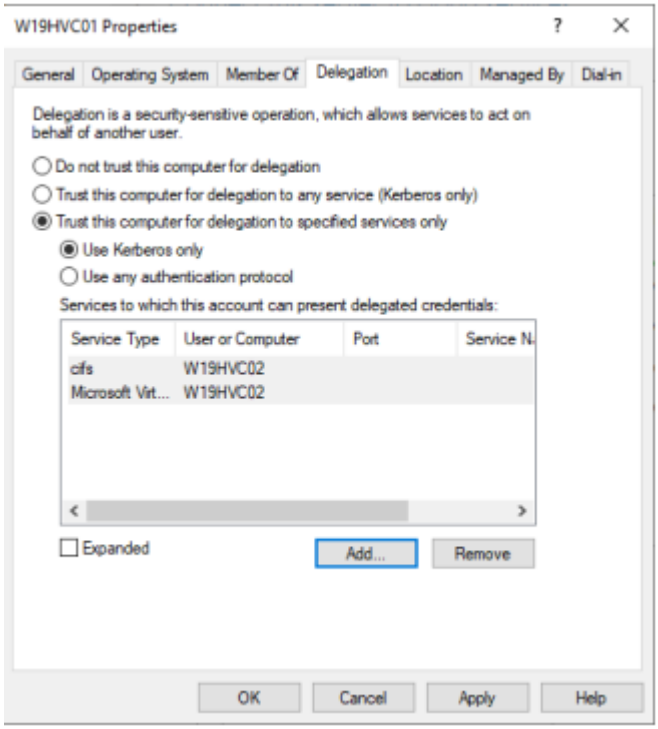
PS > `Move-VM -Name <name VM> -DestinationHost W19HVC02`

## Shared nothing Mig:

Nodes moeten zich niet in dezelfde cluster bevinden (best wel in hetzelfde AD bevinden)

Vink live migration aan settings

Delegation aanvkingen onder > AD users & computers > Computers (op beide hosts)




Cifs & virtual services…

Vm mag niet in een cluster zitten

Hyper v manager > select vm > move > virtual machine > select destination node > move data to single location (1ste)>  create folder op destination node & select deze map > finish

VM Storage migration:

Vm migregen naar andere plaats

Hyper v host > select vm > move > move vm storage > data to single location > kies map > next > finish

```
PS> Move-VMStorage -VMName <naam VM> -DestinationStoragePath `
'E:\Virtual Machines\W19HV01\'
```

Storage cluster migration:

Cluster manager > vm > move > vm storage

# HYPER V REPLICA

Export vm if located in S2D

Hyper v manager > settings > replication configuration > vink enable aan > kies Kerberos > “allow replication from any authenticated server” > browse > kies map waar replica komt te staan > OK

Firewall enable rule for replica:

Inbound rules > hyper-v replica http listener > enable

Inbound rules > hyper-v replica https listener > enable

Op andere hyper v manager:

Kies een vm > start vm > rmk > enable replication > replica server : w19hvc02> next > compress data > next > next > next > recovery point : maintain only the latest > next > send initial copy over network > next > finish

Test failover:

Select vm > replication > test failover

Replication > stop test failover

Failover:

Select vm> replication > failover

Failover back to production site:

Select vm > replication > reverse replication > next > configure server > enable > use Kerberos > allow replication from any server > OK

Firewall rules:

Inbound rules > hyper-v replica http listener > enable

Inbound rules > hyper-v replica https listener > enable

>ok >next> use kerberus > compress data > next > last recovery point > send over network > next > finish

Terug op production site :

Select Vm > replication > planned failover > reverse the replication direction after failover > failover

# Docker commands:

Docker pull **download docker image**

Docker run **download & run that docker image**

Docker ps **show running docker containers**

docker ps -a **show all containers that exist = running & not running**

docker run -d -it --hostname cont1 --name cont1 --network prodnet --ip 10.10.10.10 -p 80:80 mcr.microsoft.com/windows/servercore:ltsc2019

- Detached mode: run in background
- -it : interactive terminal : (keeps container alive)
- Hostname set hostname of the container
- Name : name the container
- Network: set custom network (if not set this will be nat)
- Ip : set custom ip
- P : port fist host port then container port
- mcr.microsoft.com/windows/servercore:ltsc2019 == image:version

docker logs 'container' : vieuw logs of container

docker exec -it 'container' powershell : open a powershell terminal in the running container

➔ bin/bash : for linux

docker stop 'container' : stops running container

docker stop $(docker ps -a -q) : stops a containers

docker rm 'container' : deletes the container

docker images : show all the local images

docker network ls : show all the networks available for docker

**NETWORKING**

**remove nat network**

geen bridge network aanmaken: \\server\c$\programdata\docker\config

maak hier daemon.json aan

daar in → { "bridge" : "none" }

get-hnsnetwork| remove-hnsnetwork

**Nat**

docker network create --driver nat --subnet 192.168.0.0/24 --gateway 192.168.0.1 mynat

create new nat interface:

- driver : choose nat, none, transparent
- in what subnet is this network located
- give gateway of this network
- name your created network

**Transparent**

docker network create --driver transparent --subnet 10.10.10.0/24 --gateway 10.10.10.1 `

-o com.docker.network.windowsshim.dnsservers=10.10.10.1 `

-o com.docker.network.windowsshim.interface="Ethernet0" `

prodnet

**DOCKER FILE create image**

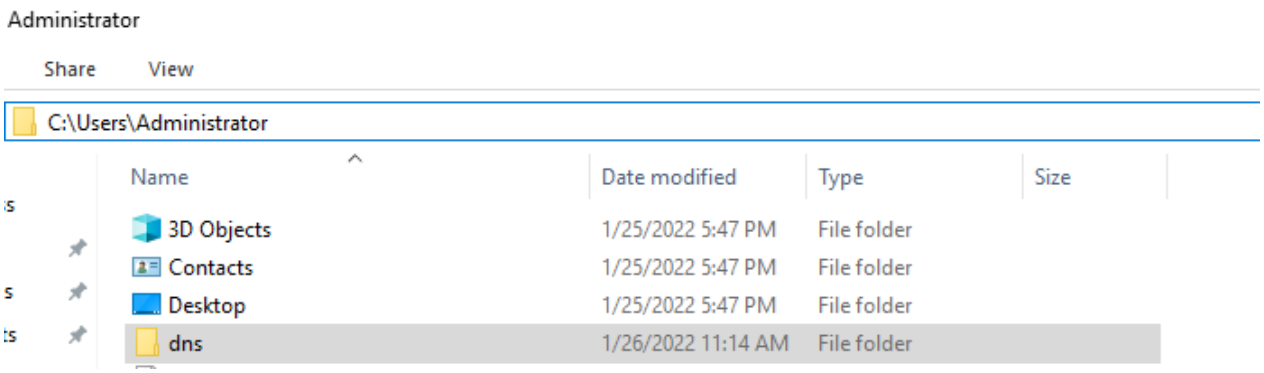
Create container with dockerfile

docker uses backslash '\' as escape char.

in the docker file we will say that the backtic '`' is the escape char.

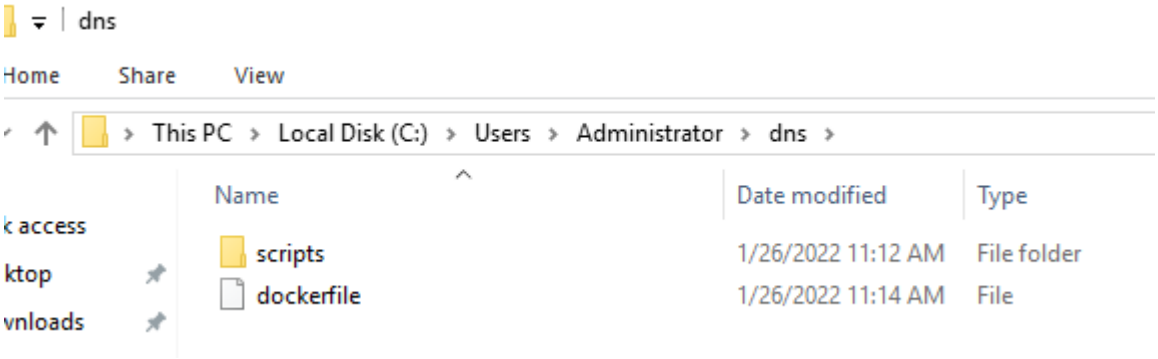
so we start to build our file.

first create a folder for the container under the user that runes PowerShell

i have created the folder dns & within this folder there is another folder called ‘scripts’

In the dns folder you will create a docker file called ‘dockerfile’ this file does not have an extension




now we will edit the dockerfile with notepad

content:

_____

```
# escape=`
FROM mcr.microsoft.com/windows/servercore:ltsc2019
COPY scripts/installdhcp.ps1 c:\installdhcp.ps1
RUN powershell.exe -ExecutionPolicy Bypass c:\installdhcp.ps1
```

_____

now create the image

cd dns

docker image build --tag dnsimage .

extended docker file that keeps host alive  and other finetuneing

____

```
# escape=`
FROM mcr.microsoft.com/windows/servercore:ltsc2019
RUN powershell -Command " ` Install-WindowsFeature DNS,RSAT-DNS-Server -Verbose `
"
ENTRYPOINT powershell -Command " `
```

```
Add-DnsServerPrimaryZone -Name company.pri -ZoneFile company.pri.dns -Verbose ; `
$localip = Get-NetIPAddress -AddressFamily ipv4 -InterfaceAlias 'vEthernet (Ethernet)*' ; `
Add-DnsServerResourceRecord -ZoneName company.pri -A -Name (Get-Content env:computername) -IPv4Address $Localip.ipaddress -Verbose ; `
ping -t localhost > NULL `

EXPOSE 80

HEALTHCHECK -- interval=2s
        CMD powershell -Command `
        try { `
        $result = get-service dns; `
        if ($result.status -eq 'Running') { exit 0 } `
        else { exit 1 } `
        } catch { exit 1 }
```

create docker hub repo

docker login : login with your credentials

docker image tag dns vermesen/dns

docker push vermees/dns

**link local storage to container storage**

op server1 onder de "c:\" maak map aan contdata : mkdir contdata

maak in deze map een file aan data.txt : typ hier in "dit is data van server1"

link dit volume aan de container via volgend commando:

docker run -d -it --name dns --hostname dns --network prodnet --ip 10.10.10.10 --volume c:\contdata:c:\contdata

**remote volumes**

op mydesktop maak map aan op bureaublad ‘moredata’ met daarin data2.txt : ‘more data’

op deze map rechtermuisklik > properties>sharing> advanced sharing…

vink share this folder aan > klik permissions > klik add> type “Everyone” > full control> ok

Onder security tab van de folder > edit > add> everyone> Full control

link remote volume to container:

vanop my desktop:

enter-pssession server1

New-smbglobalmapping -remotepath \\mydesktop\moredata\ -localpath -e

nu in powershell op server1 :

docker run -d -it --name dns --hostname dns --network prodnet --ip 10.10.10.10 --volume e:\:c:\contdata

# Openfiler

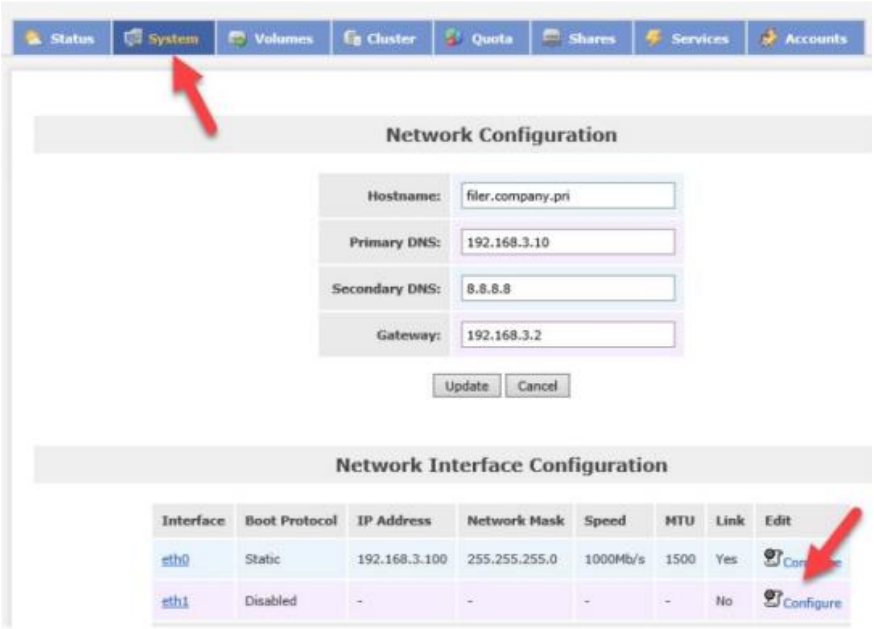
add iscsi disk

add storage network (moet niet naar internet kunnen)

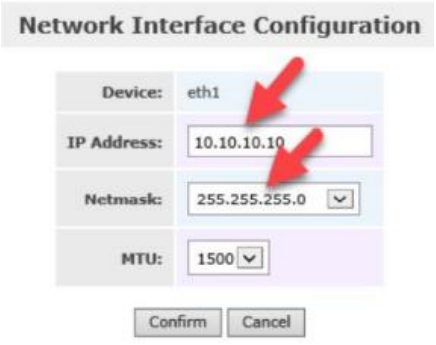
boot openfiler

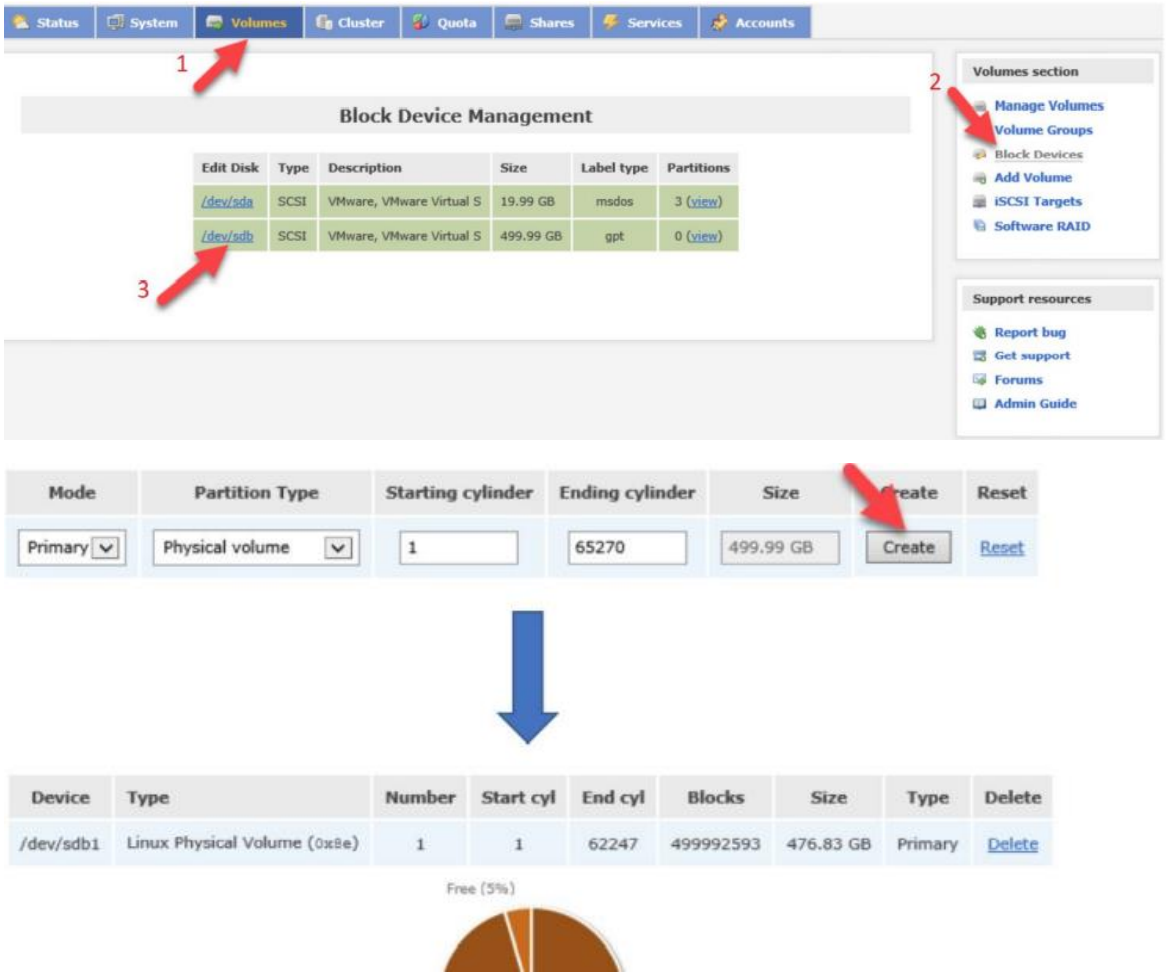
ga naar https://openfiler-Ip:446/

Configureer storage network



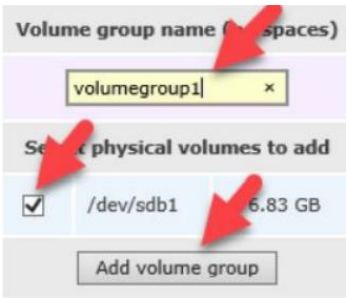






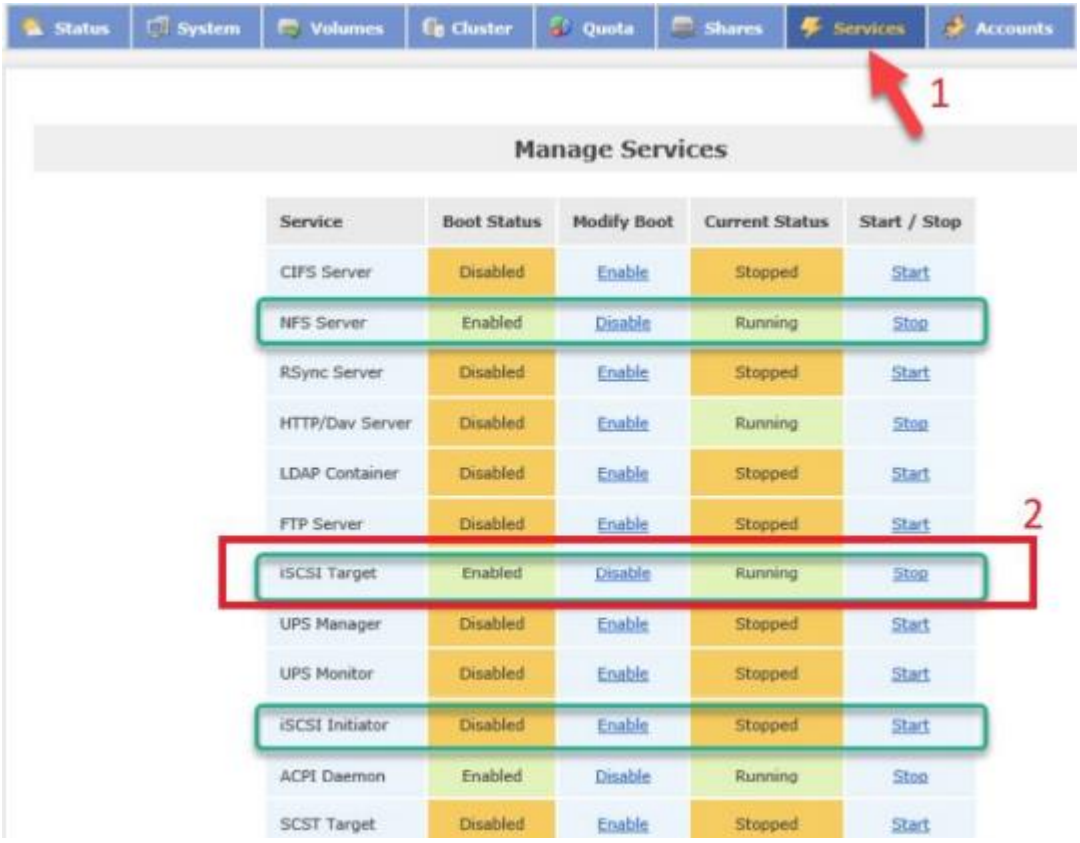

create volume groups


geef naam aan volume group en selecteer disk





zet iscsi target, nfs server, iscsi initiator op enable en start

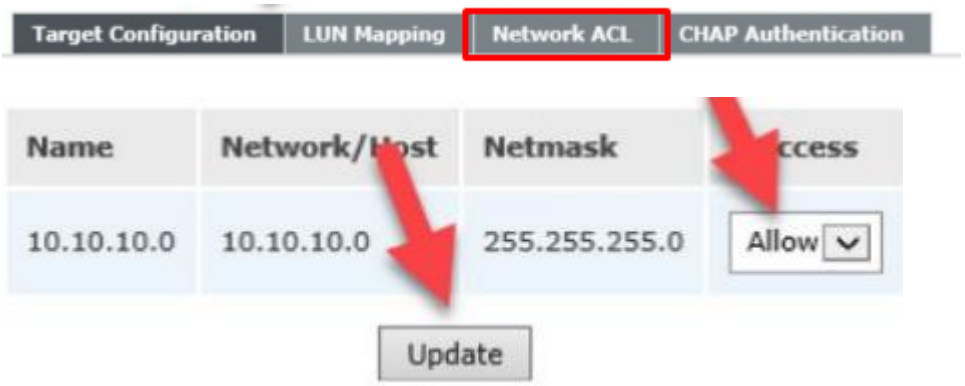



- Klik Add to make this as new iSCSI Target

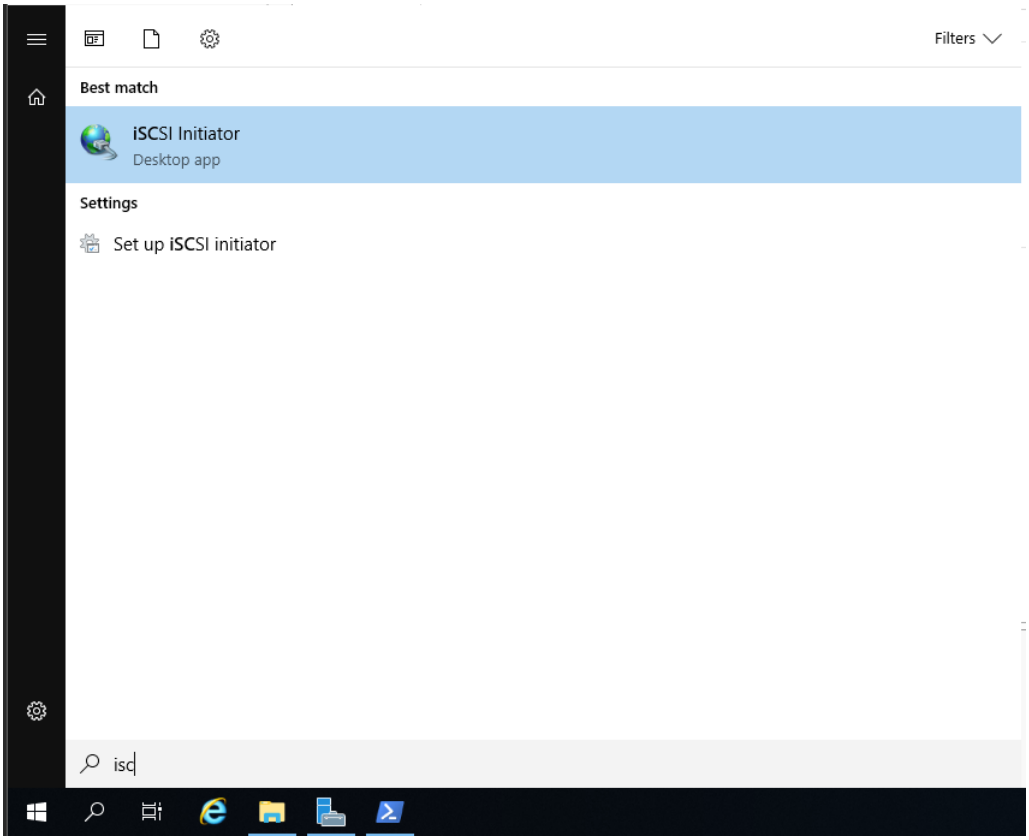




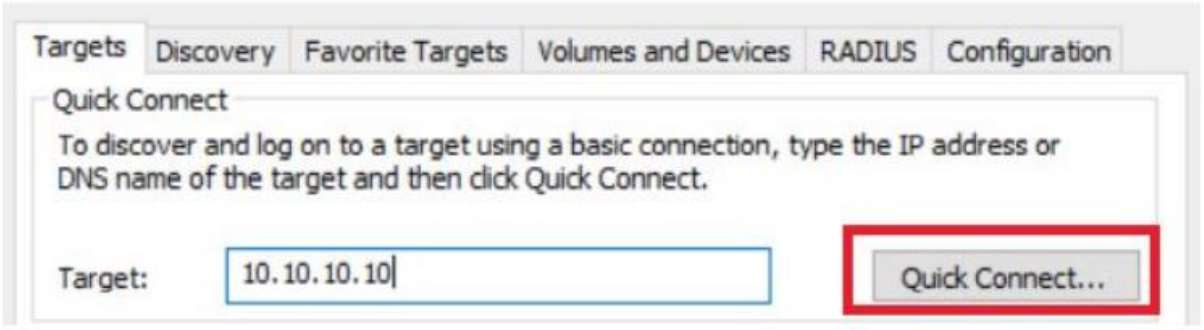
- Indien geen lijst van netwerken:
  - Klik op Local Networks

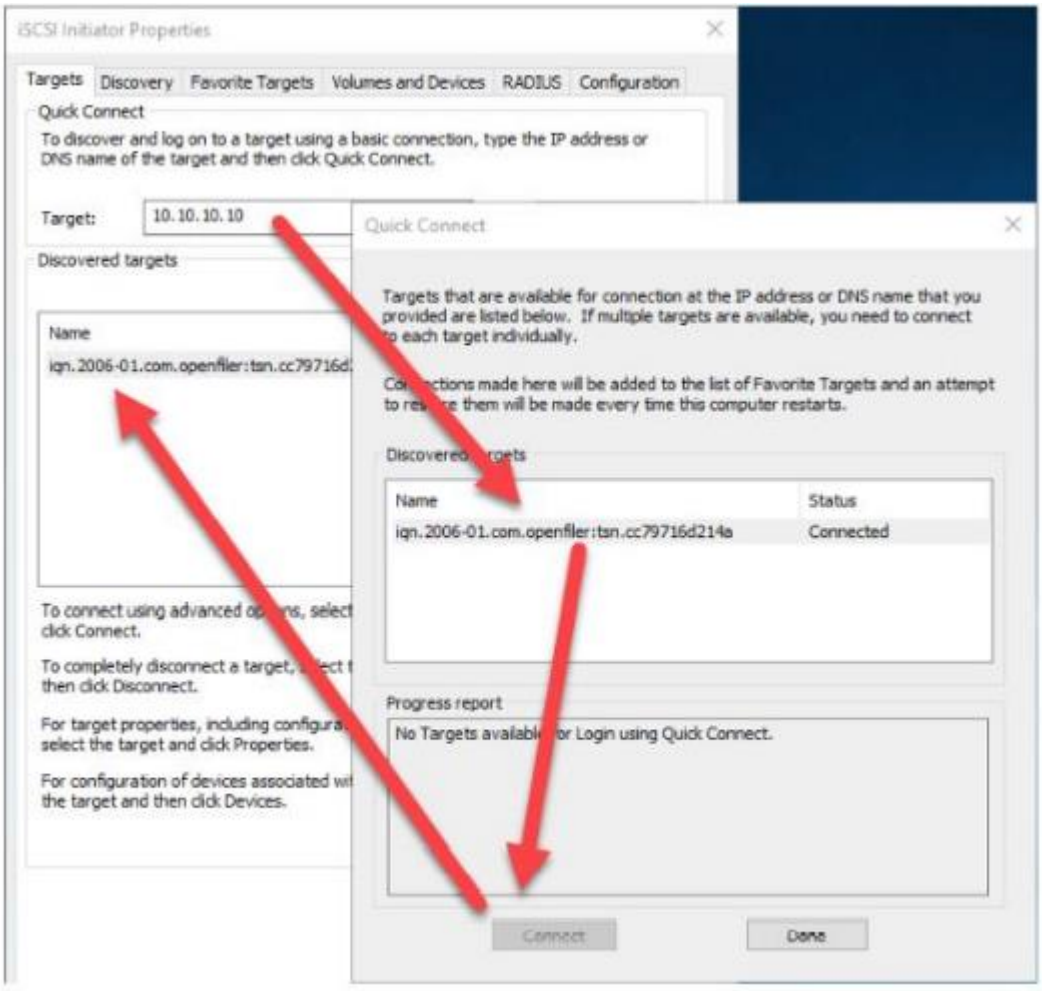




connect nu op de hyper-v host naar de iscsi initiator.

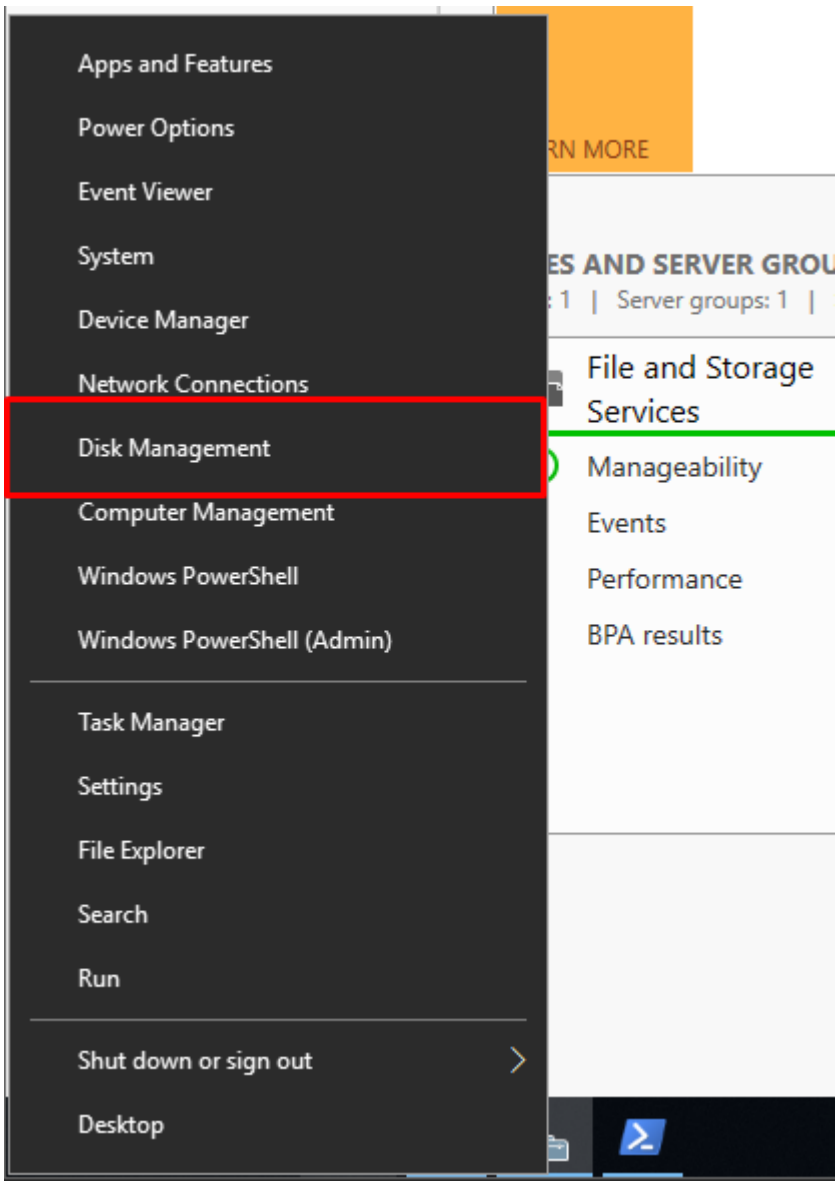



geef het ip van de openfiler op



ga nu naar diskmanagement windowstoets+X




- Rechtsklik op disk.
- Online.
- Initialize Disk (MBR)
- Rechtsklik op grote rechthoek.
- New Simple Volume...
- Drive-letter E
  Toekennen.
- Format.
- Volume label: “openfiler” als naam.