

### ### Technische documentatie

#### ## Creatie windows server 2019

1. Download een Windows Server 2019 ISO van deze website:  
<https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2019?filetype=ISO>
2. Open virtual box en maak een nieuwe server aan en noem deze Lima2
3. Adapter 1 aanpassen naar een host-only adapter
4. De server opstarten
5. Account naam: Administrator met wachtwoord: Admin2019
6. Sluit de server
7. Verwijder de iso drive
8. Voeg een tweede harde schijf toe (vdi) met 50 GB
9. Voeg een lege optische schijf toe voor de guest additions
10. Start de server weer op en installeer de guest additions
11. Na de installatie sluit de server
12. Verwijder de optische schijf voor de guest additions
13. Start de server op en voeg een gedeelde map toe. Dit is het pad naar de scripts: "p3ops-1920-red\Servers\Lima2\LIMA2\provisioning"
14. Start een domein controller op vb. Alpha2 en configureer deze volledig aan de hand van de Alpha2 scripts op github
15. Voer het script "startScript" uit op Lima2
16. Geef de credentials mee: Administrator en Admin2019

#### ## De file server configureren

1. Voordat het eerste script kan uitgevoerd worden moet er een domein controller geconfigureerd zijn en aan staan
2. Voer het "startScript" uit. Dit geeft de server de naam Lima2, joint het domein en configureert de ip instellingen
3. Na de reboot wordt het Lima2 script automatisch uitgevoerd
4. Na de uitvoering van het script Lima2 is de server geconfigureerd.
5. Powershell code van het script "startScript":

...

```
# Parameters username en password
$Username = "Administrator"
$Password = "Admin2019"
```

```
#Zorgt ervoor dat de server automatisch weer inlogt met de gebruiker
Administrator met wachtwoord Admin2019 na de reboot
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon" -Name DefaultUserName -Value $Username
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon" -Name DefaultPassword -Value $Password
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon" -Name AutoAdminLogon -Value 1
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon" -Name ForceAutoLogon -Value 1
```

```
#Een sheduled task toevoegen die het script Lima2 uitvoert na de reboot.
Import-Module -Name "ScheduledTasks"
$Sta = New-ScheduledTaskAction -Execute "powershell" -Argument "-
ExecutionPolicy Bypass Z:\Lima2.ps1"
$Stt = New-ScheduledTaskTrigger -AtLogOn
$Stp = New-ScheduledTaskPrincipal -UserId "Administrator" -RunLevel
Highest
```

```

$StTaskName = "ConfigLima2"
Register-ScheduledTask -TaskName $StTaskName -Action $Sta -Trigger $Stt -
Principal $Stp -Force

# Parameters voor de netwerk adapter
$local_ip = "172.18.1.67"
$lan_prefix = "27"
$default_gateway = ""
$preferred_dns_ip = "172.18.1.66"

# De naam van de huidige adapter veranderen naar LAN
$stemp = (Get-NetAdapter).Name
if ($stemp -ne "LAN") { Get-NetAdapter | Rename-NetAdapter -NewName "LAN"
}
else { Write-Host("Nothing to do") }
# De LAN adapter configureren met static ip en subnet masker
$stemp = (Get-NetAdapter -Name "LAN" | Get-NetIPAddress -AddressFamily
IPv4).IPAddress
if ($stemp -ne $local_ip) { New-NetIPAddress -InterfaceAlias "LAN" -
IPAddress $local_ip -PrefixLength $lan_prefix }
else { Write-Host("Nothing to do") }
# DNS server instellen
Set-DnsClientServerAddress -InterfaceAlias "LAN" -
ServerAddresses($preferred_dns_ip)

$stest = Get-Credential
# De host hernoemen naar Lima2 en toevoegen aan het domein red.local
$stemp = (Get-WmiObject Win32_ComputerSystem).Domain
if ($stemp -ne "red.local") { Add-Computer -DomainName red.local -NewName
Lima2 -Credential $stest -Restart -Force }
else { Write-Host("Nothing to do") }

```

```

Restart-Computer
```

```

1. Het tweede script "Lima2" wordt automatisch na het startScript uitgevoerd
2. Dit deel van het script installeert de nodige services voor een file server.
3. Het maakt nieuwe partities aan en formateert de volumes.
4. Vervolgens worden de nieuwe shares aangemaakt.
5. Deel 1 van het Lima2 script:

```

```
#install windows fileservices
if (!(Get-WindowsFeature -Name File-Services).Installed)
{
    Install-WindowsFeature File-Services
}

Set-ExecutionPolicy -ExecutionPolicy Unrestricted

if (!(Get-WindowsFeature -Name File-Services).Installed)
{
    Install-WindowsFeature -Name FS-Resource-Manager -
IncludeManagementTools
}

```

```
Resize-Partition -DiskNumber 0 -PartitionNumber 2 -Size 30GB

#Disk 2 initiliseren. Anders kunnen er geen partities aangemaakt worden.
Get-Disk 0 | Initialize-Disk

#Alle partities aanmaken op disk 0 en deze direct formateren indien deze
nog niet bestaan.

if (!(Test-Path D:))
{
    New-Partition -DiskNumber 0 -Size 5GB -DriveLetter D
    Format-Volume -DriveLetter D -FileSystem NTFS -NewFileSystemLabel
VerkoopData -Confirm:$False
}

if (!(Test-Path E:))
{
    New-Partition -DiskNumber 0 -Size 5GB -DriveLetter E
    Format-Volume -DriveLetter E -FileSystem NTFS -NewFileSystemLabel
OntwikkelingData -Confirm:$False
}

#Disk 1 initiliseren. Anders kunnen er geen partities aangemaakt worden.
Get-Disk 1 | Initialize-Disk

#Alle partities aanmaken op disk 1 en deze direct formateren indien deze
nog niet bestaan.
if (!(Test-Path F:))
{
    New-Partition -DiskNumber 1 -Size 5GB -DriveLetter F
    Format-Volume -DriveLetter F -FileSystem NTFS -NewFileSystemLabel
ITData -Confirm:$False
}

if (!(Test-Path G:))
{
    New-Partition -DiskNumber 1 -Size 5GB -DriveLetter G
    Format-Volume -DriveLetter G -FileSystem NTFS -NewFileSystemLabel
DirData -Confirm:$False
}

if (!(Test-Path H:))
{
    New-Partition -DiskNumber 1 -Size 5GB -DriveLetter H
    Format-Volume -DriveLetter H -FileSystem NTFS -NewFileSystemLabel
AdminData -Confirm:$False
}

if (!(Test-Path Y:))
{
    New-Partition -DiskNumber 1 -Size 5GB -DriveLetter Y
    Format-Volume -DriveLetter Y -FileSystem NTFS -NewFileSystemLabel
HomeDirs -Confirm:$False
}

if (!(Test-Path P:))
{
    New-Partition -DiskNumber 1 -Size 5GB -DriveLetter P
```

```

        Format-Volume -DriveLetter P -FileSystem NTFS -NewFileSystemLabel
ProfileDirs -Confirm:$False
    }

    if (!(Test-Path Q:))
    {
        New-Partition -DiskNumber 1 -Size 5GB -DriveLetter Q
        Format-Volume -DriveLetter Q -FileSystem NTFS -NewFileSystemLabel
ShareVerkoop -Confirm:$False
    }

New-Item -ItemType directory -Path D:\Shares\VerkoopData
New-Item -ItemType directory -Path E:\Shares\OntwikkelingData
New-Item -ItemType directory -Path F:\Shares\ITData
New-Item -ItemType directory -Path G:\Shares\DirData
New-Item -ItemType directory -Path H:\Shares\AdminData
New-Item -ItemType directory -Path Q:\Shares\ShareVerkoop
New-Item -ItemType directory -Path Y:\Shares\HomeDirs
New-Item -ItemType directory -Path P:\Shares\ProfileDirs

#Alle shares aanmaken indien deze nog niet bestaan.
if(!(Get-SmbShare -Name VerkoopData -ea 0))
{
    New-SmbShare -Name "VerkoopData" -Path "D:\Shares\VerkoopData" -
ChangeAccess "red\IT_Administratie" -FullAccess "red\Verkoop"
}
if(!(Get-SmbShare -Name OntwikkelingData -ea 0))
{
    New-SmbShare -Name "OntwikkelingData" -Path
"E:\Shares\OntwikkelingData" -ChangeAccess "red\IT_Administratie" -
FullAccess "red\Ontwikkeling"
}
if(!(Get-SmbShare -Name ItData -ea 0))
{
    New-SmbShare -Name "ITData" -Path "F:\Shares\ITData" -ChangeAccess
"red\IT_Administratie"
}
if(!(Get-SmbShare -Name DirData -ea 0))
{
    New-SmbShare -Name "DirData" -Path "G:\Shares\DirData" -
ChangeAccess "red\IT_Administratie" -FullAccess "red\Directie"
}
if(!(Get-SmbShare -Name AdminData -ea 0))
{
    New-SmbShare -Name "AdminData" -Path "H:\Shares\AdminData" -
ChangeAccess "red\IT_Administratie"
}
if(!(Get-SmbShare -Name HomeDirs -ea 0))
{
    New-SmbShare -Name "HomeDirs" -Path "Y:\Shares\HomeDirs" -
ChangeAccess "red\IT_Administratie" -FullAccess "everyone"
}
if(!(Get-SmbShare -Name ProfileDirs -ea 0))
{
    New-SmbShare -Name "ProfileDirs" -Path "P:\Shares\ProfileDirs" -
ChangeAccess "red\IT_Administratie" -FullAccess "everyone"
}
if(!(Get-SmbShare -Name ShareVerkoop -ea 0))

```

```
{
    New-SmbShare -Name "ShareVerkoop" -Path "Q:\Shares\ShareVerkoop" -
ChangeAccess "red\IT_Administratie" -ReadAccess "red\Ontwikkeling"
}
...
```

6. Enkele afbeelding hoe de server eruit ziet na de uitvoering van het script.

7. Volumes:

![Volumes] (<https://github.com/HoGentTIN/p3ops-1920-logboek-RobbyDaelman/blob/master/images/Volumes.PNG>)

8. Shares:

![Shares] (<https://github.com/HoGentTIN/p3ops-1920-logboek-RobbyDaelman/blob/master/images/Shares.PNG>)

9. Om na te kijken of de shares toegankelijk zijn heb ik tijdelijk een host-only adapter toegevoegd. In de verkenner "\\naamServer" uitvoeren. Dit is het resultaat:

![SharesOpHostSysteem] (<https://github.com/HoGentTIN/p3ops-1920-logboek-RobbyDaelman/blob/master/images/SharesOpHostSysteem.PNG>)

## Configuratie dagelijkse shadow copy

1. De server moet dagelijks een schaduw kopie maken van adminData. Om dit te automatiseren wordt er gebruik gemaakt van de scheduler.

2. Powershell code:

...

```
#Configure shadow storage voor adminData
vssadmin add shadowstorage /for=h: /on=h: /maxsize=2000mb
#Hier komt ps code voor dagelijkse shadow copy te maken
Import-Module -Name "ScheduledTasks"
$Sta = New-ScheduledTaskAction -Execute "powershell" -Argument "-
ExecutionPolicy Bypass Z:\ShadowCopy.ps1"
$Stt = New-ScheduledTaskTrigger -Daily -At 5pm
#Zorgt ervoor dat de taak met "highest privileges" wordt gexecuted.
$Stp = New-ScheduledTaskPrincipal -UserId "Administrator" -RunLevel
Highest
$StTaskName="ShadowCopy"
$StDescript="Maakt een dagelijkse schaduw kopie van AdminData"
#Registreer de taak in de task scheduler
Register-ScheduledTask -TaskName $StTaskName -Action $Sta -Description
$StDescript -Trigger $Stt -Principal $Stp
...
```

3. Deze code in het script maakt een nieuwe taak aan en voegt deze toe aan de scheduler. Deze taak runt dagelijks om 5pm het script "ShadowCopy.ps1".

4. Powershell code van ShadowCopy.ps1:

...

```
vssadmin create shadow /for=h:
...
```

5. De code in dit script is simpel. Het maakt rechtstreeks een schaduw kopie van adminData.

6. Als het eerste script is uitgevoerd, dan ziet de Task Scheduler er zo uit:

![TaskScheduler] (<https://github.com/HoGentTIN/p3ops-1920-logboek-RobbyDaelman/blob/master/images/TaskScheduler.PNG>)  
7. Vanaf 5pm wordt de toegevoegde taak een eerste keer uitgevoerd. het resultaat hiervan is te zien op de volgende afbeelding:  
![SchaduwKopie] (<https://github.com/HoGentTIN/p3ops-1920-logboek-RobbyDaelman/blob/master/images/SchaduwKopie.PNG>)

## Fileshare permissies

...

```
$permissiesVerkoopDeny = "red\Verkoop","FullControl","Deny"  
$permissiesOntwikkelingDeny = "red\Ontwikkeling","FullControl","Deny"  
#$permissiesITDeny = "red\IT_Administratie","FullControl","Deny"  
$permissiesDirectieDeny = "red\Directie","FullControl","Deny"
```

```
$permissiesVerkoopAllow = "red\Verkoop","Write","Allow"  
$permissiesOntwikkelingAllow = "red\Ontwikkeling","Write","Allow"  
$permissiesITAllow = "red\IT_Administratie","FullControl","Allow"  
$permissiesDirectieAllow = "red\Directie","FullControl","Allow"  
$permissiesOntwikkelingRead = "red\Directie","FullControl","Allow"
```

```
$acl1 = Get-Acl D:/Shares/VerkoopData  
$newRule1 = New-Object -TypeName  
System.Security.AccessControl.FileSystemAccessRule($permissiesVerkoopAllo  
w)  
$newRule2 = New-Object -TypeName  
System.Security.AccessControl.FileSystemAccessRule($permissiesITAllow)  
$newRule3 = New-Object -TypeName  
System.Security.AccessControl.FileSystemAccessRule($permissiesDirectieDen  
y)  
$newRule4 = New-Object -TypeName  
System.Security.AccessControl.FileSystemAccessRule($permissiesOntwikkelin  
gDeny)  
$acl1.SetAccessRule($newRule1)  
Set-Acl D:/Shares/VerkoopData -AclObject $acl1  
$acl1.SetAccessRule($newRule2)  
Set-Acl D:/Shares/VerkoopData -AclObject $acl1  
$acl1.SetAccessRule($newRule3)  
Set-Acl D:/Shares/VerkoopData -AclObject $acl1  
$acl1.SetAccessRule($newRule4)  
Set-Acl D:/Shares/VerkoopData -AclObject $acl1  
Start-sleep 5  
$acl1 = Get-Acl E:/Shares/OntwikkelingData  
$newRule1 = New-Object -TypeName  
System.Security.AccessControl.FileSystemAccessRule($permissiesVerkoopDeny  
)  
$newRule2 = New-Object -TypeName  
System.Security.AccessControl.FileSystemAccessRule($permissiesITAllow)  
$newRule3 = New-Object -TypeName  
System.Security.AccessControl.FileSystemAccessRule($permissiesDirectieDen  
y)  
$newRule4 = New-Object -TypeName  
System.Security.AccessControl.FileSystemAccessRule($permissiesOntwikkelin  
gAllow)  
$acl1.SetAccessRule($newRule1)  
Set-Acl E:/Shares/VerkoopData -AclObject $acl1  
$acl1.SetAccessRule($newRule2)
```

```

Set-Acl E:/Shares/VerkoopData -AclObject $acl1
$acl1.SetAccessRule($newRule3)
Set-Acl E:/Shares/VerkoopData -AclObject $acl1
$acl1.SetAccessRule($newRule4)
Set-Acl E:/Shares/VerkoopData -AclObject $acl1
Start-sleep 5
$acl1 = Get-Acl F:/Shares/ItData
$newRule1 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesVerkoopDeny
)
$newRule2 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesITAllow)
$newRule3 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesDirectieDen
y)
$newRule4 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesOntwikkelin
gDeny)
$acl1.SetAccessRule($newRule1)
Set-Acl F:/Shares/VerkoopData -AclObject $acl1
$acl1.SetAccessRule($newRule2)
Set-Acl F:/Shares/VerkoopData -AclObject $acl1
$acl1.SetAccessRule($newRule3)
Set-Acl F:/Shares/VerkoopData -AclObject $acl1
$acl1.SetAccessRule($newRule4)
Set-Acl F:/Shares/VerkoopData -AclObject $acl1
Start-sleep 5
$acl1 = Get-Acl G:/Shares/directory
$newRule1 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesVerkoopDeny
)
$newRule2 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesITAllow)
$newRule3 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesDirectieAll
ow)
$newRule4 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesOntwikkelin
gDeny)
$acl1.SetAccessRule($newRule1)
Set-Acl G:/Shares/VerkoopData -AclObject $acl1
$acl1.SetAccessRule($newRule2)
Set-Acl G:/Shares/VerkoopData -AclObject $acl1
$acl1.SetAccessRule($newRule3)
Set-Acl G:/Shares/VerkoopData -AclObject $acl1
$acl1.SetAccessRule($newRule4)
Set-Acl G:/Shares/VerkoopData -AclObject $acl1
Start-sleep 5
$acl1 = Get-Acl H:/Shares/AdminData
$newRule1 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesVerkoopDeny
)
$newRule2 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesITAllow)
$newRule3 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesDirectieDen
y)

```

```

$newRule4 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesOntwikkelin
gDeny)
$ac11.SetAccessRule($newRule1)
Set-Acl H:/Shares/VerkoopData -AclObject $ac11
$ac11.SetAccessRule($newRule2)
Set-Acl H:/Shares/VerkoopData -AclObject $ac11
$ac11.SetAccessRule($newRule3)
Set-Acl H:/Shares/VerkoopData -AclObject $ac11
$ac11.SetAccessRule($newRule4)
Set-Acl H:/Shares/VerkoopData -AclObject $ac11
Start-sleep 5
$ac11 = Get-Acl Y:/Shares/HomeDirs
$newRule1 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesVerkoopAllo
w)
$newRule2 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesITAllow)
$newRule3 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesDirectieAll
ow)
$newRule4 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesOntwikkelin
gAllow)
$ac11.SetAccessRule($newRule1)
Set-Acl Y:/Shares/VerkoopData -AclObject $ac11
$ac11.SetAccessRule($newRule2)
Set-Acl Y:/Shares/VerkoopData -AclObject $ac11
$ac11.SetAccessRule($newRule3)
Set-Acl Y:/Shares/VerkoopData -AclObject $ac11
$ac11.SetAccessRule($newRule4)
Set-Acl Y:/Shares/VerkoopData -AclObject $ac11
Start-sleep 5
$ac11 = Get-Acl Z:/Shares/ProfileDirs
$newRule1 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesVerkoopAllo
w)
$newRule2 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesITAllow)
$newRule3 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesDirectieAll
ow)
$newRule4 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesOntwikkelin
gAllow)
$ac11.SetAccessRule($newRule1)
Set-Acl Z:/Shares/VerkoopData -AclObject $ac11
$ac11.SetAccessRule($newRule2)
Set-Acl Z:/Shares/VerkoopData -AclObject $ac11
$ac11.SetAccessRule($newRule3)
Set-Acl Z:/Shares/VerkoopData -AclObject $ac11
$ac11.SetAccessRule($newRule4)
Set-Acl Z:/Shares/VerkoopData -AclObject $ac11
Start-sleep 5
$ac18 = Get-Acl Q:/Shares/ShareVerkoop
$newRule1 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesVerkoopAllo
w)

```



```

$newRule2 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesITAllow)
$newRule3 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesDirectieDen
y)
$newRule4 = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule($permessiesOntwikkelin
gRead)
$acl1.SetAccessRule($newRule1)
Set-Acl Y:/Shares/VerkoopData -AclObject $acl1
$acl1.SetAccessRule($newRule2)
Set-Acl Y:/Shares/VerkoopData -AclObject $acl1
$acl1.SetAccessRule($newRule3)
Set-Acl Y:/Shares/VerkoopData -AclObject $acl1
$acl1.SetAccessRule($newRule4)
Set-Acl Y:/Shares/VerkoopData -AclObject $acl1
Start-sleep 5

```

...

8. Elke blok in deze code configureerd de ACL toegangs permissies voor elke share.

9. alle permissie mogelijkheden zijn bonevaan gedefinieerd voor een vlot verloop van dit proces

## Maximum capaciteit shares

...

```

New-FSRMQuotaTemplate -Name "AdminData Quota" -Size 100MB
New-FSRMQuotaTemplate -Name "VerkoopData Quota" -Size 100MB
New-FSRMQuotaTemplate -Name "DirDATA Quota" -Size 100MB
New-FSRMQuotaTemplate -Name "OntwikkelingData Quota" -Size 200MB
New-FSRMQuotaTemplate -Name "ITData Quota" -Size 200MB

```

...

10. deze quota's stellen de maximum capaciteiten in voor de shares die een maximum moesten ingesteld hebben.

## Probleem op de maximum capaciteit op de drive H: of AdminData

Door een quota in te stellen op de AdminData is de maximum capaciteit 100MB. Met als gevolg dat de BuiltIn\Administrator geen shadow copy's kan maken, omdat 100MB niet voldoende ruimte is.

1. Verwijder de quota op AdminData -> Server Manager -> Tools -> File Server Resource Manager -> Quota Management -> Quotas -> In deze lijst verwijder de quota met path H:
2. Op de Windows verkenner en ga naar This PC
3. Onder Devices and drives zoeken naar de AdminData (H:) drive
4. Selecteer properties
5. Ga naar quota en Enable quota mangement
6. Limiteer de disk space to 100MB en warning level naar keuze
7. Ga naar Quota Entries en controleer dat de BUILTIN\Administrator zijn quota limit 'No Limit' is