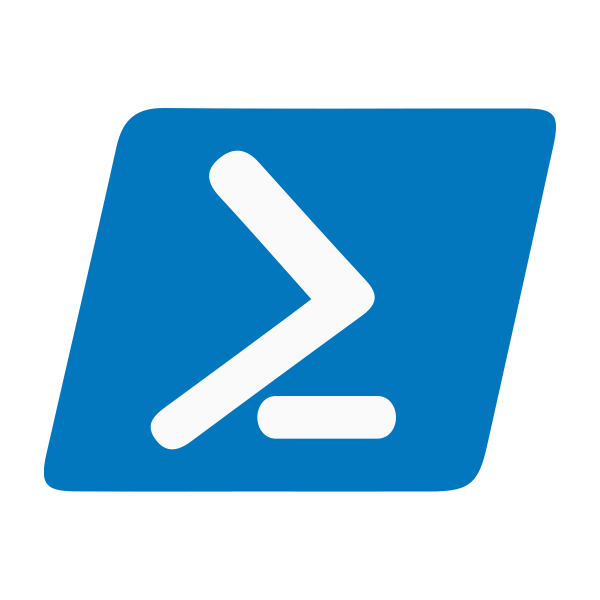
**Powershell Script Commando Documentatie**



# **Inhoudsopgave**

Contents

[**Inhoudsopgave** 2](#_Toc181386930)

[**Inleiding** 3](#_Toc181386931)

[**Power3-Install-ADDS-Unattended** 4](#_Toc181386932)

[**“In het begin was er geen administrator”** 4](#_Toc181386933)

[**#Start Transcript for logging – Start-Transcript** 5](#_Toc181386934)

[**# Check if running administrator** 5](#_Toc181386935)

[**# Get the current script path - Variable** 5](#_Toc181386936)

[**# Start a new PowerShell process with elevated privileges – Start-Process** 6](#_Toc181386937)

[**# Exit the current (non-elevated) instance of the script** 6](#_Toc181386938)

[**“De koning van basic”** 7](#_Toc181386939)

[**# Set variables** 8](#_Toc181386940)

[**# Set local administrator password** 9](#_Toc181386941)

[**# Configure network settings** 9](#_Toc181386942)

[**“Wild ADDS Approaching!”** 10](#_Toc181386943)

[**# Install the required Windows features** 11](#_Toc181386944)

[**# Import the AD DS deployment module** 11](#_Toc181386945)

[**# Install AD DS, create the forest, and promote the server to a domain controller** 12](#_Toc181386946)

[**“DNS hoe werkt dat?”** 14](#_Toc181386947)

[**# Configure DNS after AD DS promotion** 15](#_Toc181386948)

[**# Set DNS forwarders – Set-DnsServerForwarder** 15](#_Toc181386949)

[**# Create reverse lookup zone** 15](#_Toc181386950)

[**# Enable DNS scavenging** 15](#_Toc181386951)

[**“De wachtrij van het volgende script”** 16](#_Toc181386952)

[**# Create a scheduled task to run the post-restart script with administrator privileges** 16](#_Toc181386953)

[**Power3-Post-Restart-Config** 18](#_Toc181386954)

[**“Eeeeeen we zijn terug! Bijna”** 18](#_Toc181386955)

[**# Wait for AD Web Services to be available** 18](#_Toc181386956)

[**# Import the Active Directory module** 19](#_Toc181386957)

[**# Set variables** 19](#_Toc181386958)

[**“We hebben structuur nodig!”** 20](#_Toc181386959)

[**# Function to create OU structure** 20](#_Toc181386960)

[**# Create main OUs** 20](#_Toc181386961)

[**# Create sub-OUs under Departments** 21](#_Toc181386962)

[**“Een stapje terug”** 21](#_Toc181386963)

[**# Change the hostname** 21](#_Toc181386964)

[**“Wie wilt er allemaal een IP?”** 22](#_Toc181386965)

[**# Install DHCP role** 22](#_Toc181386966)

[**# Authorize DHCP server in Active Directory** 22](#_Toc181386967)

# **Inleiding**

Deze documentatie is gebaseerd op de Power3. Power3 is het script die het allergrootste is en ook alle andere code bevat. Power3 bestaat uit “power3-Install-ADDS-Unattended.ps1” en de post script “poewr3-Post-Restart-Config.ps1”.

In dit document wordt de individuele processen niet zelf uitgelegd maar juist de individuele commando’s binnen deze processen.

Het eerste script wordt als eerste behandeld en uitgelegd en dan wordt de tweede script behandeld en uitgelegd.

# **Power3-Install-ADDS-Unattended**

## **“In het begin was er geen administrator”**

**# Start transcript for logging**

**Start-Transcript -Path "C:\power3-Install-ADDS-Unattended.log" -Append**

**# Check if running as administrator**

**if (-NOT ([Security.Principal.WindowsPrincipal][Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole] "Administrator")) {**

**Write-Warning "This script requires administrator privileges. Attempting to elevate..."**

**# Get the current script path**

**$scriptPath = $MyInvocation.MyCommand.Path**

**# Start a new PowerShell process with elevated privileges**

**Start-Process powershell.exe -ArgumentList "-NoProfile -ExecutionPolicy Bypass -File `"$scriptPath`"" -Verb RunAs**

**# Exit the current (non-elevated) instance of the script**

**Exit**

**}**

### **#Start Transcript for logging – Start-Transcript**

Start-Transcript wordt gebruikt als een logging commando, dit commando wordt gebruikt aan het begin van het script zodat alles wat het script doet wordt gelogd. End-Transcript wordt gebruikt aan het eind voor de herstart.

-path: specificeert het pad waar het log bestand moet worden gestuurd/aangemaakt.

-Append: Zorgt ervoor dat het nieuwe transcript word aan het einde van een al bestaand bestand wordt geplakt.

### **# Check if running administrator**

**if (-NOT ([Security.Principal.WindowsPrincipal][Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole] "Administrator")) {**

Deze commando lijn zorgt ervoor dat er eerst gekeken wordt of Powershell met Administrator runt.

De “-NOT” betekent als het juist NIET zo is betekent dat de statement waar is.

### **# Get the current script path - Variable**

**$scriptPath = $MyInvocation.MyCommand.Path**

Dit commando zet een variable neer met $scriptPath en $MyInvocation.MyCommand.Path zorgt er juist voor dat de script locatie automatisch wordt opgehaald zodat wanneer deze van locatie veranderd dit niet handmatig hoeft worden aangepast.

### **# Start a new PowerShell process with elevated privileges – Start-Process**

**Start-Process powershell.exe -ArgumentList "-NoProfile -ExecutionPolicy Bypass -File `"$scriptPath`"" -Verb RunAs**

Start-Process powershell.exe: Start een process/executable in dit geval PowerShell

-ArgumentList: Een argumentenlijst wat PowerShell precies moet uitvoeren.

-NoProfile: Er wordt geen specifiek profiel gebruikt.

-ExecutionPolicy Bypass: De policy die executie scripts blokkeert liggend aan het script. Deze staat op Bypass zodat hij dit script doorlaat en het daarna weer op “limited” zet.

=File’”$scriptPath’”: Het bestand dat hij moet gebruiken in dit geval de variabel $scriptPath

-Verb RunAs: De verb is een argument dat hij moet gebruiken wanneer PowerShell opstart. In dit geval RunAs start PowerShell op met Administrator

### **# Exit the current (non-elevated) instance of the script**

**Exit**

Het enige commando hier is het non-elevated PowerShell Window te sluiten.

## **“De koning van basic”**

Om te beginnen hebben we een basis nodig voor de server, terwijl dit deels ook al gedaan wordt via het XML bestaat dit begin stuk uit conformaties om zeker te weten dat het allemaal correct is ingesteld.

**# Set variables**

**$DomainName = "groepjjvg.local"**

**$NetbiosName = "GROEPJJVG"**

**$AdminPassword = "P@ssw0rd123!"**

**$SafeModeAdminPassword = ConvertTo-SecureString $AdminPassword -AsPlainText -Force**

**$IPAddress = "10.7.220.10"**

**$PrefixLength = 24**

**$DefaultGateway = "10.7.220.3"**

**$DNSServer = "10.7.220.10"  # This will be the IP of the DC itself**

**$NewHostName = "DC01-JJVG"**

**$ForwarderIP = $DefaultGateway  # Using the default gateway as DNS forwarder**

**# Set local administrator password**

**$AdminUser = [ADSI]"WinNT://./Administrator,User"**

**$AdminUser.SetPassword($AdminPassword)**

**$AdminUser.SetInfo()**

**Write-Host "Local administrator password has been set."**

**# Configure network settings**

**$InterfaceIndex = (Get-NetAdapter | Where-Object {$\_.Status -eq "Up"}).InterfaceIndex**

**New-NetIPAddress -InterfaceIndex $InterfaceIndex -IPAddress $IPAddress -PrefixLength $PrefixLength -DefaultGateway $DefaultGateway**

**Set-DnsClientServerAddress -InterfaceIndex $InterfaceIndex -ServerAddresses $DNSServer**

### **# Set variables**

**$DomainName = "groepjjvg.local"**

**$NetbiosName = "GROEPJJVG"**

**$AdminPassword = "P@ssw0rd123!"**

**$SafeModeAdminPassword = ConvertTo-SecureString $AdminPassword -AsPlainText -Force**

**$IPAddress = "10.7.220.10"**

**$PrefixLength = 24**

**$DefaultGateway = "10.7.220.3"**

**$DNSServer = "10.7.220.10"  # This will be the IP of the DC itself**

**$NewHostName = "DC01-JJVG"**

**$ForwarderIP = $DefaultGateway  # Using the default gateway as DNS forwarder**

Deze variabelen hierboven mogen allemaal veranderd worden naar wens behalve $SafeModeAdminPassword.

$DomainName: Het naam van het domein die de Active Directory moet gebruiken/krijgen

$NetbiosName: De Netbios naam van de Active Directory voor naamgeving.

$AdminPassword: Het wachtwoord dat de ingebouwde Administrator moet krijgen. Zorg hiervoor dat het wachtwoord bij het aanpassen aan de Active Directory Eisen voldoet!

$SafeModeAdminPassword: Gebruikt $AdminPassword en transferred het naar een secure string en forceert het.

$IPAddress: Het ip adres van de machine.

$PrefixLength: De subnet lengte.

$DefaultGateway: De Default Gateway, hier wordt dan het IP van de Firewall, Router, Modem gebruikt.

$DNSServer= IP van de DNS server. Vul hier of de DC IP in als deze ook als DNS server wordt gebruikt. Anders gebruik die van Google of die van de Firewall/Router/Modem.

$NewHostName= Veranderd de hostnaam van de machine.

$ForwarderIP: Gebruikt de $DefaultGateway verander deze als de DNS forwarder anders is

### **# Set local administrator password**

**$AdminUser = [ADSI]"WinNT://./Administrator,User"**

**$AdminUser.SetPassword($AdminPassword)**

**$AdminUser.SetInfo()**

**Write-Host "Local administrator password has been set."**

Zet het admin wachtwoord.

$AdminUser = [ADSI]"WinNT://./Administrator,User": Maakt een variabelen aan die de AdminUser referenced

$AdminUser.SetPassword($AdminPassword): Zet het $AdminPassword voor de Administrator

$AdminUser.SetInfo(): Zet het Admin wachtwoord in voor de Administrator.

### **# Configure network settings**

**$InterfaceIndex = (Get-NetAdapter | Where-Object {$\_.Status -eq "Up"}).InterfaceIndex**

**New-NetIPAddress -InterfaceIndex $InterfaceIndex -IPAddress $IPAddress -PrefixLength $PrefixLength -DefaultGateway $DefaultGateway**

**Set-DnsClientServerAddress -InterfaceIndex $InterfaceIndex -ServerAddresses $DNSServer**

Veranderd het netwerk adapter volgens de variabelen die boven zijn aangegeven.

$InterfaceIndex=(Get-NetAdapter | Where-Object {$\_.Status -eq “Up”}).InterfaceIndex: Gebruikt een string voor het net adapter op te halen via Where-Object. Where-Object zoekt dan een netwerk adapter die de status “up” heeft.

New-NetIPAddress: Deze is voor het IP adres static te maken door alle variabelen op te halen en te gebruiken voor de interface

-InterfaceIndex: De interface/adapter die moet worden gebruikt.

-IPAddress: Het IP adres die de machine moet krijgen.

-PrefixLength: De lengte van de subnet

-DefaultGateway: De Default Gateway van de machine/router/modem.

Set-DnsClientServerAddress: -InterfaceIndex $InterfaceIndex -ServerAddresses $DNSServer

-ServerAddresses: De DNS adres(sen)

## **“Wild ADDS Approaching!”**

Het volgende wordt het Active Directory geinstalleerd. Tezamen met het Active Directory wordt automatisch ook DNS geinstalleerd.

**# Install the required Windows features**

**Write-Host "Installing ADDS and DNS features..."**

**Install-WindowsFeature -Name AD-Domain-Services, DNS -IncludeManagementTools**

**# Import the AD DS deployment module**

**Import-Module ADDSDeployment**

**# Install AD DS, create the forest, and promote the server to a domain controller**

**Write-Host "Installing AD DS, creating the forest, and promoting to domain controller..."**

**$ADDSParams = @{**

**CreateDnsDelegation = $false**

**DatabasePath = "C:\Windows\NTDS"**

**DomainMode = "WinThreshold"**

**DomainName = $DomainName**

**DomainNetbiosName = $NetbiosName**

**ForestMode = "WinThreshold"**

**InstallDns = $true**

**LogPath = "C:\Windows\NTDS"**

**NoRebootOnCompletion = $true**

**SysvolPath = "C:\Windows\SYSVOL"**

**Force = $true**

**SafeModeAdministratorPassword = $SafeModeAdminPassword**

**}**

**try {**

**Install-ADDSForest @ADDSParams**

**}**

**catch {**

**Write-Error "Failed to install AD DS: $\_"**

**Stop-Transcript**

**Exit 1**

**}**

### **# Install the required Windows features**

**Write-Host "Installing ADDS and DNS features..."**

**Install-WindowsFeature -Name AD-Domain-Services, DNS -IncludeManagementTools**

Installeert de Windows Feature of te wel de serverrollen Active Directory en DNS. De -IncludeManagementTools zorgt ervoor dat we ook de beheer tools er bij krijgen zodat we deze zelf kunnen configureren.

### **# Import the AD DS deployment module**

**Import-Module ADDSDeployment**

Importeert de module van de voor de deployment.

### **# Install AD DS, create the forest, and promote the server to a domain controller**

**Write-Host "Installing AD DS, creating the forest, and promoting to domain controller..."**

**$ADDSParams = @{**

**CreateDnsDelegation = $false**

**DatabasePath = "C:\Windows\NTDS"**

**DomainMode = "WinThreshold"**

**DomainName = $DomainName**

**DomainNetbiosName = $NetbiosName**

**ForestMode = "WinThreshold"**

**InstallDns = $true**

**LogPath = "C:\Windows\NTDS"**

**NoRebootOnCompletion = $true**

**SysvolPath = "C:\Windows\SYSVOL"**

**Force = $true**

**SafeModeAdministratorPassword = $SafeModeAdminPassword**

**}**

**try {**

**Install-ADDSForest @ADDSParams**

**}**

**catch {**

**Write-Error "Failed to install AD DS: $\_"**

**Stop-Transcript**

**Exit 1**

**}**

Maakt een parameter tafel met alle instellingen die gebruikt worden voor het instaleren van de AD Forest

CreateDnsDelegation: Bepaalt of er wel of geen delegation moet worden gedaan van het DNS, in dit geval op false

DatabasePath:

DomainMode: Het functioneel level van het Domain omdat, we voor de forest Windows 2022 gebruiken, gebruiken we WinThreshold, Dit is voor WinServer 2016 of hoger.

DomainName: naam van het domein

DomainNetbiosName: Net Bios naam van het domein

ForestMode: Hetzelfde als domain mode.

InstallDns: Omdat, Active Directory het beste met DNS werkt wordt al vanuit het Active Directory de DNS geinstalleerd. Als je dit later wilt doen zet dit dan op $false

LogPath:

NoRebootOnCompletion: Staat standaard op $false maar, omdat wij nog een paar extra commando’s moeten uitvoeren staat dit op $true. Start de server automatisch op na instaleren forest.

Sysvol Path: bepaald het pad van sysvol.

Force: Forceert de configuratie en instellingen van deze parameters.

SafeModeAdministratorPassword: Het wachtwoord waarvoor we eerder een variable voor hebben gezet.

De try and catch commando’s zijn er in geval dat er een fout voorkomt dat hij de installatie stopt de transcriptie beindigt en het script ook beindigt.

Install-ADDSForest @ADDSParams: Installeert de Active Directory Forest met de parameters.

## **“DNS hoe werkt dat?”**

Terwijl we eigenlijk al de DNS hebben geinstalleerd bij het instaleren en configureren van de Active Directory moet ook DNS geconfigureerd worden.

**Write-Host "Configuring DNS..."**

**try {**

**# Set DNS forwarders**

**Set-DnsServerForwarder -IPAddress $ForwarderIP -PassThru**

**# Create reverse lookup zone**

**$NetworkID = $IPAddress.Substring(0, $IPAddress.LastIndexOf("."))**

**Add-DnsServerPrimaryZone -NetworkID "$NetworkID.0/$PrefixLength" -ReplicationScope "Forest" -DynamicUpdate "Secure"**

**# Enable DNS scavenging**

**Set-DnsServerScavenging -ScavengingState $true -ScavengingInterval 7.00:00:00 -RefreshInterval 7.00:00:00 -NoRefreshInterval 7.00:00:00**

**Write-Host "DNS configuration completed successfully."**

**}**

**catch {**

**Write-Error "Failed to configure DNS: $\_"**

**}**

### **# Configure DNS after AD DS promotion**

Zoals bij de Active Directory wordt er een try and catch uitgevoerd als er eventueel fouten zijn waardoor het script moeten zou moeten stoppen.

### **# Set DNS forwarders – Set-DnsServerForwarder**

**Set-DnsServerForwarder -IPAddress $ForwarderIP -PassThru**

De DNS forwarder wordt als eerste gedaan bij de configuratie. Dit zorgt ervoor dat de naam kan worden vertaald naar het IP

-IPAddress: Gebruikt het $ForwarderIP van de variabelen

-PassThru: Zorgt ervoor dat we een output krijgen van het commando waar dit normaal niet is.

### **# Create reverse lookup zone**

**$NetworkID = $IPAddress.Substring(0, $IPAddress.LastIndexOf("."))**

**Add-DnsServerPrimaryZone -NetworkID "$NetworkID.0/$PrefixLength" -ReplicationScope "Forest" -DynamicUpdate "Secure"**

De reverse lookup zone wordt als tweede geconfigureerd. De reverse lookup zone doet het tegenovergestelde van de forwarder waarbij het juist het IP naar de juiste naam vertaald.

$NetworkID: Dit is de variabel die wordt gebruikt om ervoor te zorgen dat het juiste stuk van het adres wordt gepakt. Dus i.p.v. 10.7.220.10 pakt deze 10.7.220.0

Add-DnsServerPrimaryZone: Maakt een primair zone aan voor de reverse look up zone

-NetworkID: Gebruikt de $NetworkID variabel tezamen met het juiste prefix.

-ReplicationScope: Waar de DNS zich moet repliceren in dit geval door de hele forest.

-DynamicUpdate: Zijn de enige dynamische updates die hij uitvoert, dit wordt standaard gezet op secure, dit betekent dat hij zich alleen update tezamen met het Active Directory/Domein.

### **# Enable DNS scavenging**

**Set-DnsServerScavenging -ScavengingState $true -ScavengingInterval 7.00:00:00 -RefreshInterval 7.00:00:00 -NoRefreshInterval 7.00:00:00**

DnsServerScavenging zorgt ervoor dat oude records in de DNS automatisch worden verwijderd in 7 dagen, dit kan worden aangepast naar wens.

## **“De wachtrij van het volgende script”**

Omdat, er een herstart moet gebeuren is voor het tweede script de task scheduler gebruikt die zich in Windows bevindt zodat gelijk na de herstart het nieuwe script gelijk kan gaan draaien.

**# Create a scheduled task to run the post-restart script with administrator privileges**

**$postRestartScriptPath = "C:\Windows\Setup\Scripts\power3-Post-Restart-Config.ps1"**

**$Action = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument "-ExecutionPolicy Bypass -WindowStyle Hidden -File `"$postRestartScriptPath`""**

**$Trigger = New-ScheduledTaskTrigger -AtStartup**

**$Settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfGoingOnBatteries -StartWhenAvailable -RunOnlyIfNetworkAvailable -WakeToRun**

**$Principal = New-ScheduledTaskPrincipal -UserId "SYSTEM" -LogonType ServiceAccount -RunLevel Highest**

**$Task = New-ScheduledTask -Action $Action -Trigger $Trigger -Settings $Settings -Principal $Principal**

**Register-ScheduledTask -TaskName "Complete AD DS Setup" -InputObject $Task -Force**

### **# Create a scheduled task to run the post-restart script with administrator privileges**

**$postRestartScriptPath = "C:\Windows\Setup\Scripts\power3-Post-Restart-Config.ps1"**

De locatie waar het tweede zich bevindt wat in dezelfde folder is waar het eerste script staat.

**$Action = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument "-ExecutionPolicy Bypass -WindowStyle Hidden -File `"$postRestartScriptPath`""**

Maakt een actie die de scheduled task moet uitvoeren.

-execute: De executable die hij moet gaan gebruiken

-Argument: Wat de Powershell moet doen terwijl hij executeert

-ExecutionPolicy: staat op Bypass zodat het script altijd doorgaat en zichzelf dan weer op limited zet

-WindowStyle: Bepaalt of je de Powershell script ziet executeren staat op hidden zodat het stilletjes blijft draaien.

-File: Welk bestand de PowerShell script moet gebruiken.

**$Trigger = New-ScheduledTaskTrigger -AtStartup**

De trigger is wat er moet gebeuren voordat deze task van actie gaat.

-AtStartup: Dit argument is zodat hij zodra opnieuw opstart de task gaat uitvoeren.

**$Settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfGoingOnBatteries -StartWhenAvailable -RunOnlyIfNetworkAvailable -WakeToRun**

Deze variabel zorgt ervoor dat het niet uitmaakt wat er gebeurt na de herstart dat hij zich altijd laat dragen met de belangrijkste instelling “-StartWhenAvailable”

**$Principal = New-ScheduledTaskPrincipal -UserId "SYSTEM" -LogonType ServiceAccount -RunLevel Highest**Maakt de principal aan, de principal is wat bepaald welke user hij moet gebruiken en op welk level het script moet draaien

-UserID: Dit is puur gezegd de user die gebruikt moet worden.

-LogonType: De soort login dat gebruikt moet worden voor de user of ieder geval wat voor een account dit is. Omdat, dit een builtin account is die meestal de services regelt, is de LogonType – ServiceAccount

-RunLevel: Bepaalt op welk level hij moet draaien omdat, het script moet worden gedraaid op administrator moet deze op Highest staan.

**$Task = New-ScheduledTask -Action $Action -Trigger $Trigger -Settings $Settings -Principal $Principal**

Deze variabel is wat de task zelf ook aan maakt met alle andere variabelen die er aangemaakt zijn.

**Register-ScheduledTask -TaskName "Complete AD DS Setup" -InputObject $Task -Force**

Registreert de task in de scheduler.

# **Power3-Post-Restart-Config**

## **“Eeeeeen we zijn terug! Bijna”**

Nadat de PC een herstart heeft uitgevoerd begint het script eerst met het start-transcript commando waardoor alles weer gelogd wordt.

### **# Wait for AD Web Services to be available**

**$maxAttempts = 60**

**$attempts = 0**

**while ($attempts -lt $maxAttempts) {**

**try {**

**Get-ADDomainController -ErrorAction Stop**

**break**

**}**

**catch {**

**Write-Host "Waiting for AD Web Services to be available... Attempt $($attempts + 1) of $maxAttempts"**

**Start-Sleep -Seconds 10**

**$attempts++**

**}**

**}**

**if ($attempts -eq $maxAttempts) {**

**Write-Error "AD Web Services did not become available in time. Exiting script."**

**Stop-Transcript**

**Exit 1**

**}**

Het eerste stuk van het script is het wachten voor de AD Web Service. Dit is nodig voor het OU-structuur maar, ook voor de andere commando’s die zich voordoen zoals de DHCP-configuratie en de naamverandering van de machine.

Het script zorgt ervoor dat hij voor max 60 keer kijkt of de AD Web Service aanstaat elke 10 seconden. Dit is ongeveer 10 minuten in tijd. Omdat, de computerinstellingen wanneer er een Active Directory is geïnstalleerd kan best lang duren dus is er ook een lange tijd gepakt voor de max attempts.

Overschrijdt hij de max attempts stopt het script. De attempts stoppen automatisch wanneer hij detecteert of de AD Web Service is opgestart.

### **# Import the Active Directory module**

**Import-Module ActiveDirectory**

Importeert net zoals bij het vorige script de Active Directory module maar, in dit geval voor het OU-structuur. Hierdoor kunnen we ook voor dit script de PowerShell commando’s gebruiken voor de Active Directory.

### **# Set variables**

**$DomainName = "groepjjvg.local"**

**$NewHostName = "DC01-JJVG"**

**$IPAddress = "10.7.220.10"**

**$DHCPScope = "10.7.220.0"**

**$DHCPStartRange = "10.7.220.100"**

**$DHCPEndRange = "10.7.220.200"**

**$DHCPSubnetMask = "255.255.255.0"**

**$DefaultGateway = "10.7.220.3"**

**$DNSServer = "10.7.220.10"**

Deze variabelen zijn precies hetzelfde gestructureerd als bij het eerste script maar, dan gespecificeerd voor de commando’s die in het tweede script voor komen zoals DHCP en naamsverandering.

## **“We hebben structuur nodig!”**

Het volgende is het OU-structuur deze kan aangepast worden naar wens. Deze staat nu ingesteld voor een willekeurige OU’s.

### **# Function to create OU structure**

**function Create-OUStructure {**

Maakt een functie aan die we kunnen roepen om het OU-structuur aan te maken.

### **# Create main OUs**

**New-ADOrganizationalUnit -Name "Departments" -Path "DC=$($DomainName.Split('.')[0]),DC=$($DomainName.Split('.')[1])" -ProtectedFromAccidentalDeletion $true**

**New-ADOrganizationalUnit -Name "Security Groups" -Path "DC=$($DomainName.Split('.')[0]),DC=$($DomainName.Split('.')[1])" -ProtectedFromAccidentalDeletion $true**

**New-ADOrganizationalUnit -Name "Service Accounts" -Path "DC=$($DomainName.Split('.')[0]),DC=$($DomainName.Split('.')[1])" -ProtectedFromAccidentalDeletion $true**

Voor het OU-structuur zijn er verschillende al aangemaakt maar, als er meerdere moeten worden gecreëerd kan dit om de snippet te kopiëren en dan onder een andere snippet te plakken. De primair OU’s gaan rechtstreeks onder het domein

New-ADOrganizationalUnit: Dit is de commando dat het OU-structuur aanmaakt

-Name: Pas deze aan om de naam van het OU te veranderen.

-Path: Dit pad is waar de OU of waaronder de OU moet komen te staan, de hoofd OU komt allemaal onder het domein te staan oftewel in dit geval de DC.

-ProtectedFromAccidentalDeletion: Zorgt ervoor dat je deze niet zomaar kan verwijderen, staan standaard op $true kan uitgezet worden met $false.

### **# Create sub-OUs under Departments**

**$departments = @("IT", "HR", "Finance", "Marketing", "Sales")**

**foreach ($dept in $departments) {**

**New-ADOrganizationalUnit -Name $dept -Path "OU=Departments,DC=$($DomainName.Split('.')[0]),DC=$($DomainName.Split('.')[1])" -ProtectedFromAccidentalDeletion $true**

**}**

$departments: Kan worden aangepast door **@("IT", "HR", "Finance", "Marketing", "Sales")** aan te passen. Verander dan bijvoorbeeld **“IT”** naar bijvoorbeeld **“SD”**

Foreach ($dept in $departments): Voor elke loop in de sub-OU benamingen wordt deze dan toegevoegd aan primair OU.

In de New-ADOrganizationalUnit i.p.v. dat alleen het domein wordt gespecifieerd in de pathing wordt de pathing “OU=’primairOU’,DC…”. Voor elke sub map die je wilt maken moet je het pad dan elke keer groter maken als er dan nog een OU moet worden gemaakt onder **“Marketing”** kan je dit doen om voor de **“primairOU”** het sub-OU **“Marketing”** te zetten.

Voorbeeld: **"OU=Marketing,OU=Departments,DC=$($DomainName.Split('.')[0]),DC=$($DomainName.Split('.')[1])"**

## **“Een stapje terug”**

### **# Change the hostname**

**Write-Host "Changing hostname to $NewHostName..."**

**try {**

**Rename-Computer -NewName $NewHostName -Force -ErrorAction Stop**

**Write-Host "Hostname changed successfully."**

**}**

**catch {**

**Write-Error "Failed to change hostname: $\_"**

**}**

Zorgt ervoor dat de naam van de server wordt veranderd.

-NewName: Gebruikt de variabel $NewHostName voor de naam te veranderen

-Force: Forceert de naamsverandering

-ErrorAction: Wat hij moet doen wanneer er een fout optreedt

## **“Wie wilt er allemaal een IP?”**

### **# Install DHCP role**

**Write-Host "Installing DHCP role..."**

**Install-WindowsFeature -Name DHCP -IncludeManagementTools**

Installeert de serverrol DHCP

### **# Authorize DHCP server in Active Directory**

**Write-Host "Authorizing DHCP server in Active Directory..."**

**Add-DhcpServerInDC -DnsName $NewHostName -IPAddress $IPAddress**

Voegt de juiste DNS naam en het IP adres toe aan de DHCP server voor het autoriseren van de Active Directory.

### **# Configure DHCP**

**Write-Host "Configuring DHCP..."**

**try {**

#### **# Add DHCP scope**

**Add-DhcpServerv4Scope -Name "Client Scope" -StartRange $DHCPStartRange -EndRange $DHCPEndRange -SubnetMask $DHCPSubnetMask -State Active**

Dit maakt de DHCP-server scope aan maar configureert nog niet de opties. Dit specificeert alleen waar de scope begint en waar deze eindigt en gebruikt de variabelen die bij het begin van het script staan.

#### **# Set DHCP scope options**

**Set-DhcpServerv4OptionValue -ScopeId $DHCPScope -Router $DefaultGateway**

**Set-DhcpServerv4OptionValue -ScopeId $DHCPScope -DnsServer $DNSServer**

**Set-DhcpServerv4OptionValue -ScopeId $DHCPScope -DnsDomain $DomainName**

De opties van de scope worden hier aangepast en aangegeven, dit bevat dan de DNS-Server de router dus de Default-Gateway en de domein naam.

#### **# Configure server-level options**

**Set-DhcpServerv4OptionValue -ComputerName $NewHostName -DnsDomain $DomainName -DnsServer $DNSServer**

Deze commando lijn doet precies hetzelfde als de scope opties maar specificeert deze voor IPV4.

### **# Complete post-deployment configuration of DHCP**

**Write-Host "Completing post-deployment configuration of DHCP..."**

**try {**

**# Complete DHCP configuration**

**Set-ItemProperty -Path registry::HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\ServerManager\Roles\12 -Name ConfigurationState -Value 2**

Dit zorgt ervoor dat er geen extra autorisatie na dat de DHCP is geconfigureerd nog moet gebeuren en maakt het zodat de DHCP zichzelf automatisch afmaakt.

### **# Restart the DHCP service**

**Restart-Service dhcpserver**

Als laatste wordt de DHCP-server nog is herstart om alle instellingen aan te zetten.

## **“De laatste stappen”**

### **# Remove the scheduled task**

**Unregister-ScheduledTask -TaskName "Complete AD DS Setup" -Confirm:$false**

**Write-Host "Post-restart configuration complete, including DHCP installation and post-deployment configuration."**

**Write-Host "The server will restart immediately to apply all changes."**

**Write-Host "You can check C:\Scripts\power3-Post-Restart-Config.log for execution details."**

**# Stop transcript**

**Stop-Transcript**

**# Force restart the computer immediately**

**Restart-Computer -Force**

Unregister-ScheduledTask: Haalt de scheduled task die we gemaakt hebben bij het eerste script eruit zodat deze zichzelf niet nog een keer uitvoert en herstart de server daarna gelijk.

Na deze laatste herstart is de server helemaal ingesteld met de serverrollen.