Konfiguartion Server Core

Da der Servercore eine gui hat und nur über CMD und Powershell konfoguriert werden kann. Wird es hier etwas sapnnender.

Es gibt mehrere Möglichkeiten den Core zu konfigurieren

1. Das programm sconfig  
     
   In der CMD den Befehl sconfig eingeben.

Normalerweise erscheint dieser Bildschirm  
  
Ein Bild, das Text, Elektronik, Screenshot, Computer enthält.

KI-generierte Inhalte können fehlerhaft sein.

1. Aber sconfig hat wegen der Konfiguration der VM keine Netzwerkkarte gefunden und mäckert deswegen
2. Ein Bild, das Text, Screenshot, Software, Betriebssystem enthält.

   KI-generierte Inhalte können fehlerhaft sein.
3. Um das zu beheben, muss der Treiber für die Netzwerkkarte von der Befehlszeile aus installiert werden. Und wenn wir bereits dabei sind auch den Treiber für das Spichermanagment

Zuerst schaue ich aber mit ipconfig /all an, wie es ums Netzwerk steht

pnputil /add-driver e:\netkvm\2k19\amd64\netkvm.inf /install

pnputil /add-driver e:\balloon\2k19\amd64\balloon.inf /install

erst einmal die Treiber fürs die netzwerkkarte. Mit einem vorher nachher  
  
Ein Bild, das Text, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Der treiber wurde erfolgreich installiert, mal sehen wie das ipconfig nun aussieht

Ein Bild, das Text, Elektronik, Screenshot, Display enthält.

KI-generierte Inhalte können fehlerhaft sein.

IPv4, IPv6, APIPA, da der Server bisher keinen DHCP findet. Netzwerk ist also da.

Jetzt geht’s über sconfig weiter-

Ein Bild, das Text, Elektronik, Screenshot, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Als erstes ändere ich über Punkt 2 den Namen des Servers, Danach ändere ich die Netzwerkeinstellen in Punkt 8. Danach konfiguriere ich über Punkt 4 und 7 die Remoteverbindung. Zum Schluss installiere ich über Punkt 6 die Updates

Name ändern über punkt 2

Ein Bild, das Text, Screenshot, Electric Blue (Farbe), Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Netzwerk einrichten über unkt 8

Ein Bild, das Text, Screenshot, Schrift, Electric Blue (Farbe) enthält.

KI-generierte Inhalte können fehlerhaft sein.

Da der Server keinen Kontakt zu einem DHCP hatte, hat der die APIPA

Diese ersetze ich nun durch eine statische IP von 192.168.2.4

Füge den Standartgateway 192.168.2.1 und den DNS Server 192.168.2.1

Ein Bild, das Text, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Jetzt fehlt noch die ip zum dns server den ich über punkt 2 festlege

Ein Bild, das Text, Elektronik, Screenshot, Display enthält.

KI-generierte Inhalte können fehlerhaft sein.

Nun konfiguriere ich über die punkte 4 und 7 die Remoteverbindung, damit ich diesen Server Core über einen Servermanager remote konfigurieren kann

Ein Bild, das Text, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Hier wähle ich die punkte 1 und 3 aus

Und nun den RDS aktivieren  
  
Ein Bild, das Text, Screenshot, Electric Blue (Farbe), Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Dort aktiviere ich die zweite Option

Um zu testen, ob das ganze erfolgreich war, nutze ich den Server Manager und füge diesen Core hinzu

Ein Bild, das Text, Screenshot, Software, Webseite enthält.

KI-generierte Inhalte können fehlerhaft sein.

Dort gebe ich in der Suchmaske den Namen des Servers an.

Ein Bild, das Text, Screenshot, Display, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Wenn alles geklappt hat, erscheint der Servercore im Servermanager

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Elektronik, Screenshot, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Zwei dinge passen noch nicht ganz, einmal haben wir hier IPv6 aktiv. Und die Firewall braucht noch ein paar Regeln.

Dazu nutze ich die Powershell für folgende Befehle

Disable-NetAdapterBinding -name Ethernet -ComponentID ms\_tcpip6

netsh advfirewall firewall set rule group="datei- und druckerfreigabe" new enable=yes

netsh advfirewall firewall set rule group="Remotedienstverwaltung" new enable=yes

netsh advfirewall firewall set rule group="Remote-Ereignisprotokollverwaltung" new enable=yes

netsh advfirewall firewall set rule group="Remotevolumeverwaltung" new enable=yes

netsh advfirewall firewall set rule group="Remotedesktop" new enable=yes

netsh advfirewall firewall set rule group="Windows-Remoteverwaltung" new enable=yes

netsh advfirewall firewall set rule group="Netzwerkerkennung" new enable=yes

netsh advfirewall firewall set rule group="Remoteereignisüberwachung" new enable=yes

Zum schluss muss der Server Core als Trusted Host auf dem gui server hinzugefügt werden. Dazu nnutze ich diese befehle in der Powershell:  
  
Set-Item WSMan:\localhost\Client\TrustedHosts -Value "hostname-vom-core"

Restart-Service WinRM

Automatisierte Installation

Als Erleichterung der Installation habe ich mir ein Skript erstellt, das die Installation erleichtert

Es ist derzeit auf Proxmox VM#s ausgelegt

# Parameter die hinter der ps1 geschreiben werden

param (

[switch]$joinDomain, [switch]$update

)

$Hardware = Get-NetAdapter

if($hardware -eq $null)

{

write-host "keine hardware gefunde.. Installiere"

pnputil /add-driver g:\netkvm\2k19\amd64\netkvm.inf /install

pnputil /add-driver g:\balloon\2k19\amd64\balloon.inf /install

}else{

write-host " hardware gefunden.. weiter gehts"

}

#eingaben die in variablen gespeichert werden

$ip = Read-Host "IP-Adresse eingeben"

$gw = Read-Host "IP für Gateway eingeben"

$dns = Read-Host "IP für den DNS Server eingeben"

$name = Read-Host "Name für den Rechner angeben"

$domain = Read-Host "Bei Einladung in eine Domäne,den Domänenname eingeben"

$netadapter = Get-NetAdapter

$aindex =$netadapter[0].ifIndex

$aname = $netadapter[0].name

#ändern der grundkonfig

Rename-Computer -NewName $name

Set-NetIPInterface -InterfaceIndex $aindex -Dhcp disable

New-NetIPAddress -interfacealias $aname -IPAddress $ip -AddressFamily IPv4 -PrefixLength 24 -DefaultGateway $gw

Set-DnsClientServerAddress -Interfacealias $aname -ServerAddresses $dns

Disable-NetAdapterBinding -name $aname -ComponentID ms\_tcpip6

netsh advfirewall firewall set rule group="datei- und druckerfreigabe" new enable=yes

netsh advfirewall firewall set rule group="Remotedienstverwaltung" new enable=yes

netsh advfirewall firewall set rule group="Remote-Ereignisprotokollverwaltung" new enable=yes

netsh advfirewall firewall set rule group="Remotevolumeverwaltung" new enable=yes

netsh advfirewall firewall set rule group="Remotedesktop" new enable=yes

netsh advfirewall firewall set rule group="Windows-Remoteverwaltung" new enable=yes

netsh advfirewall firewall set rule group="Netzwerkerkennung" new enable=yes

netsh advfirewall firewall set rule group="Remoteereignisüberwachung" new enable=yes

# --- Remoteverwaltung aktivieren ---

Write-Host "Aktiviere Remoteverwaltung und PowerShell-Remoting..."

Enable-PSRemoting -Force

Set-Item -Path WSMan:\localhost\Service\AllowUnencrypted -Value $true

Set-Item -Path WSMan:\localhost\Service\Auth\Basic -Value $true

winrm quickconfig -force

# --- Remote Desktop aktivieren ---

Write-Host "Aktiviere Remote Desktop..."

Set-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Terminal Server" -Name "fDenyTSConnections" -Value 0

# RDP durch die Firewall lassen

Enable-NetFirewallRule -DisplayGroup "Remotedesktop"

# das ist das herunterladen von updates und die instalation. Habs auskommentiert weils zu lange dauert

if ($update){

Install-Module -name PSWindowsUpdate -Force

Get-Wulist

install-windowsupdate -microsoftupdate -acceptall

}

# das zeugs soll als parameter ausgeführt werden

if ($joinDomain){

Add-Computer -DomainName $domain -NewName $name -Credential (get-credential)

}

Shutdown /s /t 5

Zum Schluss installiere ich dann die Updates, die etwas länger brauchen.

Über den punkt 6 komme ich zu den updates.  
Ein Bild, das Text, Screenshot, Software, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

param (

[switch]$joinDomain,

[switch]$update

)

# === Netzwerktreiber prüfen ===

$Hardware = Get-NetAdapter

if ($Hardware -eq $null) {

Write-Host "⚠️ Keine Netzwerkkarte gefunden. Versuche Treiberinstallation..."

try {

pnputil /add-driver e:\netkvm\2k19\amd64\netkvm.inf /install

pnputil /add-driver e:\balloon\2k19\amd64\balloon.inf /install

Write-Host "✅ Treiberinstallation abgeschlossen."

} catch {

Write-Host "❌ Treiberinstallation fehlgeschlagen: $\_"

exit 1

}

} else {

Write-Host "✅ Netzwerkkarte erkannt. Fahre fort..."

}

# === Benutzereingaben ===

$ip = Read-Host "➡️ IP-Adresse eingeben (z. B. 192.168.0.100)"

$gw = Read-Host "➡️ Gateway-IP eingeben"

$dns = Read-Host "➡️ DNS-Server-IP eingeben"

$name = Read-Host "➡️ Rechnername"

$domain = Read-Host "➡️ Domänenname (wenn gewünscht, sonst leer)"

# === Netzwerkkonfiguration ===

$netadapter = Get-NetAdapter | Where-Object { $\_.Status -eq "Up" } | Select-Object -First 1

if (-not $netadapter) {

Write-Host "❌ Kein aktiver Netzwerkadapter gefunden."

exit 1

}

$aindex = $netadapter.ifIndex

$aname = $netadapter.Name

Write-Host "🔧 Konfiguriere IP, Gateway und DNS..."

Rename-Computer -NewName $name -Force

Set-NetIPInterface -InterfaceIndex $aindex -Dhcp Disabled

New-NetIPAddress -InterfaceAlias $aname -IPAddress $ip -PrefixLength 24 -DefaultGateway $gw

Set-DnsClientServerAddress -InterfaceAlias $aname -ServerAddresses $dns

Disable-NetAdapterBinding -Name $aname -ComponentID ms\_tcpip6

# === Firewallregel aktivieren ===

Write-Host "🛡️ Aktiviere Firewallregel für Datei- und Druckerfreigabe..."

try {

Enable-NetFirewallRule -DisplayGroup "Datei- und Druckerfreigabe"

Write-Host "✅ Firewallregel erfolgreich aktiviert."

} catch {

Write-Host "❌ Fehler beim Aktivieren der Firewallregel: $\_"

}

# === Optional: Updates installieren ===

if ($update) {

Write-Host "⬇️ Installiere Windows Updates (kann dauern)..."

Install-Module -Name PSWindowsUpdate -Force -Confirm:$false

Import-Module PSWindowsUpdate

Get-WUList

Install-WindowsUpdate -MicrosoftUpdate -AcceptAll -AutoReboot

}

# === Optional: Domain Join ===

if ($joinDomain) {

try {

Write-Host "🏢 Trete der Domäne bei..."

Add-Computer -DomainName $domain -NewName $name -Credential (Get-Credential) -Force

} catch {

Write-Host "❌ Fehler beim Domain Join: $\_"

}

}

# === Neustart ===

Write-Host "🔁 Neustart in 5 Sekunden..."

Start-Sleep -Seconds 5

Restart-Computer