Basis konfiguration af ESXi via PowerCLI

Mads Fog Albrechtslund
vExpert 2014, PernixPro og Veeam Technical Expert 2014
Konsulent, Businessmann A/S
Twitter: @Hazenet

businessmann
Moving businesses forward through IT

Hvad skal konfigureres?

- DNS
- Default Gateway
- Search Domain
- Hostname
- NTP
- Syslog
- Core Dump
- System Resource Reservation
- SSH Server
- SATP/PSP Rule
- Local Datastore name

Opdeling af scriptet

- Instruktion og hjælp
- Pause funktion, IP-check og PSSnapin
- Certificate og ESXi connection
- Konfiguration af ESXi
- Læs konfiguration fra ESXi
- Vis konfigurationen og skriv til fil

Instruktion og hjælp

```
<#
.SYNOPSIS
This is a Powershell script to configure ESXi Host, after clean installation.

.DESCRIPTION
The script, assumes that it is a clean installation of ESXi, that the root password is set and a static IP is
configured.
When given a ESXi Host IP and a ESXi Hostname will configure the following:

* DNS
* Default Gateway
* Search Domain
* Hostname
* NTP
* Syslog
* Core Dump
* System Resource Reservation
* SSH Server
* SATP/PSP Rule for HP 3PAR
* Local Datastore name

.EXAMPLE
./ESXi-Config-Script.ps1 -ESXiHostIP 192.168.1.30 -ESXiHostname esxi01
Configures the ESXi Host (192.168.1.30), according to the variables in the script.
Afterwards reads the config from the ESXi Host, and writes it to the console.

.EXAMPLE
./ESXi-Config-Script.ps1 -ESXiHostIP 192.166.1.30 -ESXiHostname esxi01 -CreateOutputFile
Configures the ESXi Host (192.168.1.30), according to the variables in the script.
Afterwards reads the config from the ESXi Host, and writes it to the console,
and creates a outputfile, in the directory specified in the script variables.

.NOTES
Developed by Mads Fog Albrectslund, Bussinessmann A/S
Version: 1.0
Date: 2014-04-30

.Link
http://www.businessmann.dk

#>
```

Input og variabler

```powershell
# Input Paramters
[CmdletBinding()]
Param(
    [Parameter(Mandatory=$true)]
    [string]$ESXiHostIP,
    [Parameter(Mandatory=$true)]
    [string]$ESXiHostName,

    [switch]$CreateOutputFile
    )

# Ask for ESXi password
$ESXiPasswordFromUser = Read-Host 'What is the ESXi password?' -AsSecureString

# Convert ESXi Password to clear text
$ESXiPassword =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($ESXiPassw
ordFromUser))

# ESXi Variables
$NTPServer = "time.euro.apple.com"
$DNS1 = "192.168.1.10"
$DNS2 = "192.168.1.11"
$Domain = "hazenet.dk"
$SearchDomain = "hazenet.dk"
$DefaultGateway = "192.168.1.1"
$SysLogServerAndPort = "tcp://192.168.1.20:514"
$CoreDumpVMKnic = "vmk0"
$CoreDumpIP = "192.168.1.20"
$CoreDumpPort = "6500"
$SystemResourceResevationMHz = 1000
$SystemResourceResevationMB = 2000
$LocalDatastoreName = "Local_"+($ESXiHostname.ToUpper())

# Output File Directory Variable
$OutputFileDirectory = "C:\Users\mfa.BM-GRUPPEN\Desktop\"
```

Pause, IP-check og PSSnapin

```powershell
# Fucntion to pause the execute of the script, with a message
Function INVOKE-PAUSE() {

    Param(
        $DisplayMessage=$TRUE,
        $PauseMessage="Press any key to continue . . ."
    )

    If ($DisplayMessage)
    {
        WRITE-HOST $PauseMessage
    }
    $HOST.UI.RawUI.ReadKey("NoEcho,IncludeKeyDown") | OUT-NULL
    $HOST.UI.RawUI.Flushinputbuffer()
}

#Check if $ESXiHostIP is a valid IP address
$IsESXiHostIpFormatValid = ($ESXiHostIP -As [IPAddress]) -As [Bool]

if (!$IsESXiHostIpFormatValid)
{
   Write-host "$ESXiHostIP is not in a valid IP input format" -foregroundcolor red
   Write-host "Script will quit, please re-run script to try again" -foregroundcolor red
   INVOKE-PAUSE
   BREAK
}

# Add PowerCLI PSSnapin if not already added
if (-not (Get-PSSnapin VMware.VimAutomation.Core -ErrorAction SilentlyContinue)) {
   Add-PSSnapin VMware.VimAutomation.Core
}
```

Certificate og ESXi connect

```powershell
# Get current InvalidCertificateAction from Get-PowerCLIConfiguration
$InvalidCertificateActionVar = Get-PowerCLIConfiguration -Scope Session | Select -ExpandProperty
InvalidCertificateAction

# Set InvalidCertificateAction to "Ignore"
Set-PowerCLIConfiguration -Scope Session -InvalidCertificateAction Ignore -Confirm:$false | Out-Null

Write-Host "Connecting to ESXi Host..."

# Connect to ESXi Host
Connect-VIServer -Server $ESXiHostIP -User root -Password $ESXiPassword | Out-Null

Write-Host "Connected to ESXi Host"

# Set InvalidCertificationAction back to what is was before
Set-PowerCLIConfiguration -Scope Session -InvalidCertificateAction $InvalidCertificateActionVar -Confirm:$false |
Out-Null
```

Konfiguration, part 1

```
# Get the ESXi Host
$ESXiHost = Get-VMHost $ESXiHostIP

# Configure NTP
$ESXiHost | Add-VMHostNtpServer -NtpServer $NTPServer -Confirm:$false | Out-Null
Set-VMHostService -HostService ($ESXiHost | Get-VMHostservice | Where-Object {$_.key -eq "ntpd"}) -policy
"automatic" | Out-Null
$NTPDService = $ESXiHost | Get-VMHostService | where {$_.Key -eq 'ntpd'}
Start-VMHostService -HostService $NTPDService -Confirm:$false | Out-Null
Write-Host "NTP, configured"


#Configure DNS, Domain, HostName, SearchDomain, Default Gateway and disable DnsFromDhcp
Set-VMHostNetwork -Network ($ESXiHost | Get-VmHostNetwork) -DomainName $Domain -DNSAddress $DNS1, $DNS2 -HostName
$ESXiHostname -SearchDomain $SearchDomain -VMKernelGateway $DefaultGateway -DnsFromDhcp:$false -Confirm:$false |
Out-Null
Write-Host "DNS, Domain, HostName, SearchDomain, Default Gateway and DnsFromDhcp, configured."

#Configure Syslog and open firewall to outgoing Syslog trafic
$ESXiHost | Set-VMHostSyslogServer -SyslogServer $SysLogServerAndPort | Out-Null
$ESXiHost | Get-VMHostFirewallException | where {$_.Name -eq ('syslog')} | Set-VMHostFirewallException -Enabled:
$true | Out-Null
Write-Host "Syslog, configured."

#Configure Core Dump
$CoreDumpESXCli = $ESXiHost | Get-EsxCli
$CoreDumpESXCli.System.Coredump.Network.Set($null, $CoreDumpVMKnic, $CoreDumpIP, $CoreDumpPort) | Out-Null
$CoreDumpESXCli.System.Coredump.Network.Set($true) | Out-Null
Write-Host "CoreDump, configured."

#Enable SSH and disable SSH Warning
$SSHService = $ESXiHost | Get-VMHostService | where {$_.Key -eq 'TSM-SSH'}
Start-VMHostService -HostService $SSHService -Confirm:$false | Out-Null
Get-AdvancedSetting -Entity $ESXiHostIP | Where {$_.Name -eq "UserVars.SuppressShellWarning"} | Set-AdvancedSetting
-Value "1" -Confirm:$false | Out-Null
Write-Host "SSH, configured."
```

Konfiguration, part 2

```powershell
#Configure System Resource Reservation
$ResourceInfo = New-Object VMware.Vim.HostSystemResourceInfo
$ResourceInfo.key = "host/system"
$ResourceInfo.config = New-Object VMware.Vim.ResourceConfigSpec
$ResourceInfo.config.cpuAllocation = New-Object VMware.Vim.ResourceAllocationInfo
$ResourceInfo.config.cpuAllocation.reservation = $SystemResourceResevationMHz
$ResourceInfo.config.cpuAllocation.expandableReservation = $true
$ResourceInfo.config.cpuAllocation.limit = -1
$ResourceInfo.config.cpuAllocation.shares = New-Object VMware.Vim.SharesInfo
$ResourceInfo.config.cpuAllocation.shares.shares = 500
$ResourceInfo.config.cpuAllocation.shares.level = "custom"
$ResourceInfo.config.cpuAllocation.overheadLimit = -1
$ResourceInfo.config.memoryAllocation = New-Object VMware.Vim.ResourceAllocationInfo
$ResourceInfo.config.memoryAllocation.reservation = $SystemResourceResevationMB
$ResourceInfo.config.memoryAllocation.expandableReservation = $true
$ResourceInfo.config.memoryAllocation.limit = -1
$ResourceInfo.config.memoryAllocation.shares = New-Object VMware.Vim.SharesInfo
$ResourceInfo.config.memoryAllocation.shares.shares = 500
$ResourceInfo.config.memoryAllocation.shares.level = "custom"
$ResourceInfo.config.memoryAllocation.overheadLimit = -1

$ResourceInfoView = Get-View -Id 'HostSystem-ha-host'
$ResourceInfoView.UpdateSystemResources($resourceInfo)
Write-Host "System Ressource Reservation, configured."
```

Konfiguration, part 3

```powershell
#Configure HP 3PAR SATP/PSP Rule
$SATPesxcli = $ESXiHost | Get-EsxCli
$SATPesxcli.storage.nmp.satp.rule.add($null,"tpgs_on","HP 3PAR Custom iSCSI/FC/FCoE ALUA Rule",$null,$null,
$null,"VV",$null,"VMW_PSP_RR","iops=1","VMW_SATP_ALUA",$null,$null,"3PARdata") | Out-Null
Write-Host "HP 3PAR SATP/PSP Rule, configured."

#Rename Local datastore1
$LocalDatastore = $ESXiHost | Get-Datastore | where {$_.name -match "datastore1"}
$LocalDatastore | Set-Datastore -name $LocalDatastoreName | Out-Null
Write-Host "Local datastore name, configured."

Write-Host " "
```

Læs konfiguration, part 1

```powershell
#Read Network Configuration from ESXi Host
$ReadIP = $ESXiHost | Get-VMHostNetworkAdapter -VMKernel | Select -ExpandProperty IP
$ReadSubnet = $ESXiHost | Get-VMHostNetworkAdapter -VMKernel | Select -ExpandProperty SubnetMask
$ReadDNS = $ESXiHost | Get-VmHostNetwork | Select -ExpandProperty DNSAddress
$ReadDomainName = $ESXiHost | Get-VmHostNetwork | Select -ExpandProperty DomainName
$ReadHostName = $ESXiHost | Get-VmHostNetwork | Select -ExpandProperty HostName
$ReadSearchDomain = $ESXiHost | Get-VmHostNetwork | Select -ExpandProperty SearchDomain
$ReadDefaultGateway = $ESXiHost | Get-VmHostNetwork | Select -ExpandProperty VMKernelGateway
$ReadDnsFromDhcp = $ESXiHost | Get-VmHostNetwork | Select -ExpandProperty DnsFromDhcp
if ($ReadDnsFromDhcp -eq $false){$ReadDnsFromDhcp = $true} elseif ($ReadDnsFromDhcp -eq $true){$ReadDnsFromDhcp =
$false}
$Details | Add-Member -Name "IP" -Value $ReadIP -Membertype NoteProperty
$Details | Add-Member -Name "Subnet" -Value $ReadSubnet -Membertype NoteProperty
$Details | Add-Member -Name "DNS" -Value $ReadDNS -Membertype NoteProperty
$Details | Add-Member -Name "DomainName" -Value $ReadDomainName -Membertype NoteProperty
$Details | Add-Member -Name "Hostname" -Value $ReadHostName -Membertype NoteProperty
$Details | Add-Member -Name "SearchDomain" -Value $ReadSearchDomain -Membertype NoteProperty
$Details | Add-Member -Name "Default Gateway" -Value $ReadDefaultGateway -Membertype NoteProperty
$Details | Add-Member -Name "DNS From DHCP Disabled" -Value $ReadDnsFromDhcp -Membertype NoteProperty

#Read NTP from ESXi Host
$ReadNTP = $ESXiHost | Get-VMHostNtpServer
$Details | Add-Member -Name "NTP Server" -Value $ReadNTP -Membertype NoteProperty

#Read SSH Service from ESXi Host
$ReadSSH = $ESXiHost | Get-VMHostService | where {$_.Key -eq 'TSM-SSH'} | Select -ExpandProperty Running
$ReadSSHWarning = Get-AdvancedSetting -Entity $ESXiHostIP | Where {$_.Name -eq "UserVars.SuppressShellWarning"} |
Select -ExpandProperty Value
if ($ReadSSHWarning -eq 1){$ReadSSHWarning = $true} elseif ($ReadSSHWarning -eq 0){$ReadSSHWarning = $false}
$Details | Add-Member -Name "SSH Service Running" -Value $ReadSSH -Membertype NoteProperty
$Details | Add-Member -Name "SSH Warning Disabled" -Value $ReadSSHWarning -Membertype NoteProperty
```

Læs konfiguration, part 2

```powershell
#Read CoreDump from ESXi Host
$ReadCoreDumpEnabled = $CoreDumpESXCli.System.Coredump.Network.Get() | Select -ExpandProperty Enabled
if ($ReadCoreDumpEnabled -eq "true"){$ReadCoreDumpEnabled = $true} elseif ($ReadCoreDumpEnabled -eq "false")
{$ReadCoreDumpEnabled = $false}
$ReadCoreDumpIP = $CoreDumpESXCli.System.Coredump.Network.Get() | Select -ExpandProperty NetworkServerIP
$ReadCoreDumpPort = $CoreDumpESXCli.System.Coredump.Network.Get() | Select -ExpandProperty NetworkServerPort
$Details | Add-Member -Name "CoreDump Enabled" -Value $ReadCoreDumpEnabled -Membertype NoteProperty
$Details | Add-Member -Name "CoreDump IP" -Value $ReadCoreDumpIP -Membertype NoteProperty
$Details | Add-Member -Name "CoreDump Port" -Value $ReadCoreDumpPort -Membertype NoteProperty

#Read System Resourcce Allocation from ESXi Host
$ReadMemory = $ESXiHost.ExtensionData.SystemResources.Child | where {$_.Key -eq "host/system"} | %
{$_.Config.memoryAllocation.Reservation}
$ReadCPU = $ESXiHost.ExtensionData.SystemResources.Child | where {$_.Key -eq "host/system"} | %
{$_.Config.cpuAllocation.Reservation}
$Details | Add-Member -Name "System Memory Resevation MB" -Value $ReadMemory -Membertype NoteProperty
$Details | Add-Member -Name "System CPU Resevation MHz" -Value $ReadCPU -Membertype NoteProperty

#Read Syslog Server from ESXi Host
$ReadSysLogServer = $ESXiHost | Get-VMHostSysLogServer | Select -ExpandProperty Host
$ReadSysLogPort = $ESXiHost | Get-VMHostSysLogServer | Select -ExpandProperty Port
$ReadSysLogFirewall = $ESXiHost | Get-VMHostFirewallException | where {$_.Name -eq ('syslog')} | Select -
ExpandProperty Enabled
$Details | Add-Member -Name "Syslog Server" -Value $ReadSysLogServer -Membertype NoteProperty
$Details | Add-Member -Name "Syslog Port" -Value $ReadSysLogPort -Membertype NoteProperty
$Details | Add-Member -Name "Syslog Traffic Allowed" -Value $ReadSysLogFirewall -Membertype NoteProperty
```

Læs konfiguration, part 3

```powershell
#Read HP 3PAR SATP/PSP Rule from ESXi Host
$ReadHP3ParSATP = $SATPesxcli.storage.nmp.satp.rule.list() | where {$_.description -like "*3par*"} | Select -
ExpandProperty Name
$ReadHP3ParPSP = $SATPesxcli.storage.nmp.satp.rule.list() | where {$_.description -like "*3par*"} | Select -
ExpandProperty DefaultPSP
$ReadHP3ParPSPOptions = $SATPesxcli.storage.nmp.satp.rule.list() | where {$_.description -like "*3par*"} | Select -
ExpandProperty PSPOptions
$Details | Add-Member -Name "HP 3PAR SATP" -Value $ReadHP3ParSATP -Membertype NoteProperty
$Details | Add-Member -Name "HP 3PAR PSP" -Value $ReadHP3ParPSP -Membertype NoteProperty
$Details | Add-Member -Name "HP 3PAR PSP Options" -Value $ReadHP3ParPSPOptions -Membertype NoteProperty

#Read Local Datastore name from ESXi Host
$ReadnLocalDatastoreName = $LocalDatastore.Name
$Details | Add-Member -Name "Local Datastore Name" -Value $LocalDatastoreName -Membertype NoteProperty

# Disconnect from ESXi Host
Disconnect-VIServer -Server $ESXiHostIP -Confirm:$false

Write-Host " "
Write-Host "Configuration read from the ESXi Host:"
```

Vis konfiguration og skriv fil

```
# Display Configuraion as a list
$Details | fl

if ($CreateOutputFile)
{
    # Output file variables
    $CurrentDateAndTime = Get-Date
    $CurrentUserAndDomain = whoami
    $TheContent = $Details | fl
    $OutputFile = $OutputFileDirectory+$ESXiHostname.ToUpper()+"_"+$CurrentDateAndTime.ToString('yyyy-MM-dd')
+".txt"

    # Clear content of Output file, if it already exists
    if(Test-Path $OutputFile)
    {
        Clear-Content $OutputFile
    }

    #Write Output file
    "Script run date: "+$CurrentDateAndTime | Out-File $OutputFile -Append
    "Script run by: "+$CurrentUserAndDomain | Out-File $OutputFile -Append
    $TheContent | Out-File $OutputFile -Append
}
```

DEMO

Mulige forbedringer?

- Konfigurer vSwitches
- vMotion
- VM Network
- Indlæs settings fra XML
- Meld ESXi Host ind i et Cluster i vCenter Server

Spørgsmål?

Mads Fog Albrechtslund

vExpert 2014, PernixPro og Veeam Technical Expert 2014

Konsulent, Businessmann A/S

Twitter: @Hazenet