Microsoft

PowerShell DSC

Haiko Hertes

The not too deep dive…

Technical Summit 2016
connecting.technologies.

#msts2016

Wer spricht da?

- Haiko Hertes aus Leipzig
- Viele Jahre Erfahrung als IT-Trainer und MCT
- Derzeit „Head of IT“ bei API
  (Automotive Process Instsitute GmbH)
- Seit Anfang 2016 als MVP ausgezeichnet

Microsoft CERTIFIED Trainer
Microsoft CERTIFIED IT Professional
MVP Microsoft® Most Valuable Professional
Microsoft CERTIFIED Solutions Associate
Microsoft CERTIFIED Solutions Expert

Microsoft

Know the audience…

Wer…

… arbeitet schon regelmäßig mit PowerShell?

… hat schon mal mit DSC gearbeitet?

… arbeitet regelmäßig mit DSC?

Microsoft

Was ist DSC?

Ein kleiner Einstieg…

Microsoft

Was ist DSC?

- Deklarative Skripte
- Beschreiben Zielkonfiguration eines oder mehrerer Systeme
- „Vergleichbar“ mit Puppet, Chef, Ansible, …
- Lässt sich sowohl zum Prüfen (Test-DscConfiguration) als auch zum Umsetzen (Start-DscConfiguration) der Konfiguration verwenden

Was ist DSC?

- Bei Bedarf auch mehrere Systeme parallel
- Benötigt WMF 4.0/5.0/5.1
- Sei Mitte 2016 auch für Linux!
- Local Configuration Manager (LCM) auf den Systemen setzt die gewünschte Konfiguration um

Was ist DSC?

Name der DSC

Nodes = Zielserver

```
1   Configuration DemoWebSeite
2   {
3       Node "Server1.lab.hertes.net"
4       {
5           WindowsFeature IIS
6           {
7               Name = "Web-Server"
8               Ensure = "Present"
9           }
10      }
11  }
```

Konfigurations
-Element
„Instanz“

Microsoft

Was ist DSC?

- Innerhalb des Configuration-Blocks können fast alle Elemente einer function() benutzt werden, z.B.

  - Param()-Block
  - Variablen

- Zusätzlich:

  - Abhängigkeiten („DependsOn“)

Was ist DSC?

```
1  Configuration DemoWebSeite
2  {
3      Param(
4          [string[]]$TargetNodes = "Server1"
5      )
6
7      Node $TargetNodes
8      {
9          WindowsFeature IIS
10         {
11             Name = "Web-Server"
12             Ensure = "Present"
13         }
14
15         Service W3SVC
16         {
17             Name = "W3SVC"
18             State = "Running"
19             StartUpType = "Automatic"
20             DependsOn = "[WindowsFeature]IIS"
21             # DependsOn = "[ResourceType]ResourceName"
22         }
23     }
24 }
```

Param()-Block

Zielsystem(e) als Variable parametrisiert

Zweiter Teil hängt vom ersten ab (Reihenfolge egal!)

Was ist DSC?

Archive Resource

Environment Resource

File Resource

Group Resource

Log Resource

Package Resource

Registry Resource

Script Resource

Service Resource

User Resource

WindowsFeature Resource

WindowsProcess Resource

Was ist DSC?

> Gilt für: Windows PowerShell 4.0, Windows PowerShell 5.0

Die Ressource **Service** in Windows PowerShell DSC bietet einen Mechanismus zum Verwalten von Diensten auf dem Zielknoten.

Syntax

```
Service [string] #ResourceName
{
    Name = [string]
    [ BuiltInAccount = [string] { LocalService | LocalSystem | NetworkService }  ]
    [ Credential = [PSCredential] ]
    [ DependsOn = [string[]] ]
    [ StartupType = [string] { Automatic | Disabled | Manual }  ]
    [ State = [string] { Running | Stopped }  ]
    [ Description = [string] ]
    [ DisplayName = [string] ]
```

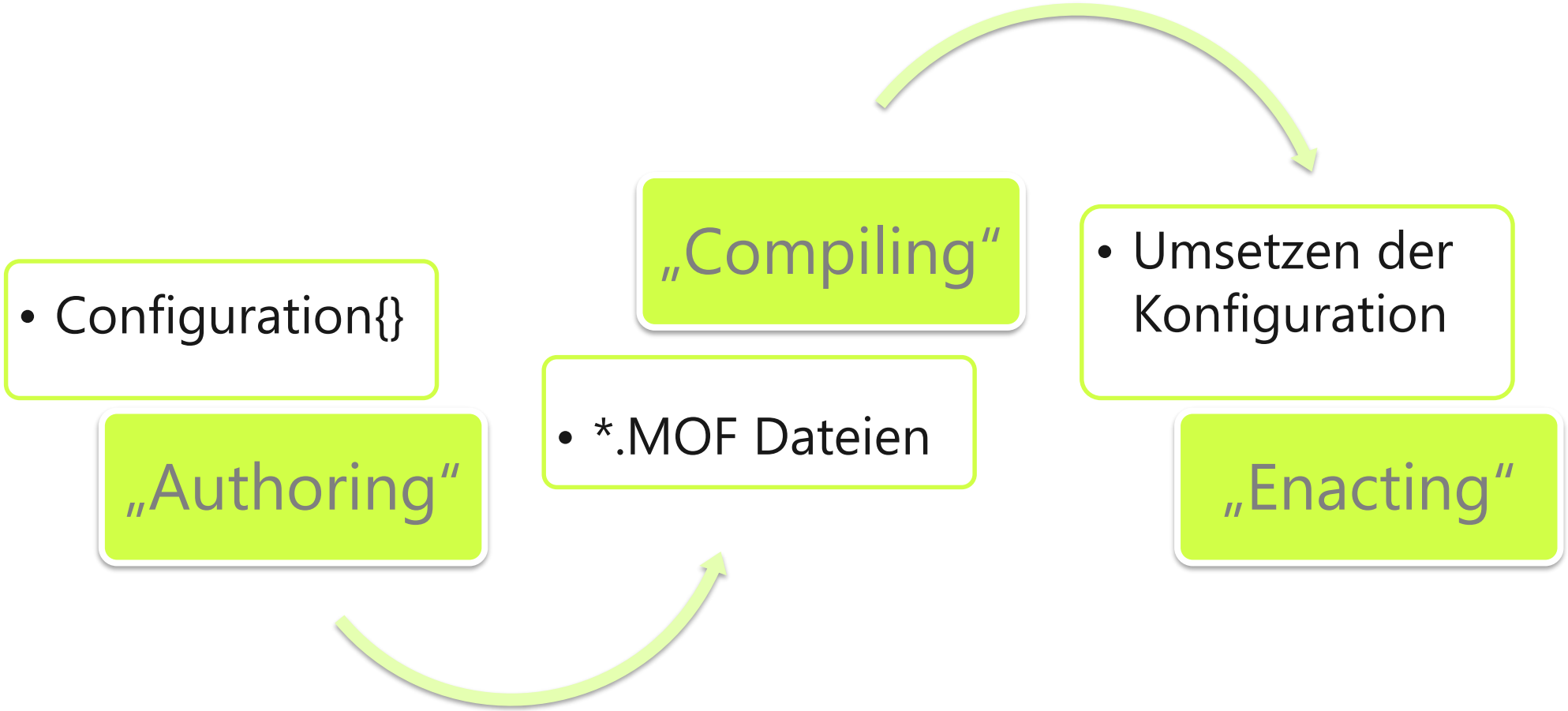
Was ist DSC?

- Einige Einsatzszenarien:

  - Ersteinrichtung von Serversystemen
  - Aufbau komplexer Umgebungen (LOB mit NLB, Failover-Cluster, …)
  - Konfiguration von Core- und Nano-Servern
  - Sicherstellung benötigter (Sicherheits-)Einstellungen

Der DSC-Worflow

Der DSC-Workflow



Authoring-Phase

- Zusammenschreiben des bzw. der Configuration{}-Files (i.d.R. als *.ps1 Datei)
- Eigene DSC-Syntax mit viele Elementen der „normalen“ PowerShell

Compiling-Phase

- Ausführen des Configuration-Blockes (z.B. mit F8 in der ISE)
  - Wichtig ist, dass der Block nicht nur deklariert, sondern auch aufgerufen wird (wie bei einer function{})
- Dabei entstehen *.MOF-Dateien
  - Diese enthalten die Konfiguration in einer anderen Syntax
  - Sprache bereits aus CIM bekannt, von „Distributed Management Task Force“ entwickelt

Enacting-Phase

- Für das Umsetzen der Konfiguration(en) gibt es zwei Möglichkeiten:
  - Push

    Zielserver bekommen die Konfiguration manuell zugewiesen
  - Pull

    LCM auf den Zielservern laden die Konfiguration in regelmäßigen Abständen von einem Pull-Server

Demo

Der DSC-Workflow im „Schnelldurchlauf“

Microsoft

Weitere Ressourcen

… und komplexere Konfigurationen

Weitere Ressourcen

- „Früher“: Nur DSC Ressource Kit unter https://gallery.technet.microsoft.com/ - nicht mehr gepflegt!
- Heute: Ressourcen können aus der Gallery (http://www.powershellgallery.com/) bzw. von GitHub (https://github.com/powershell/) heruntergeladen werden

Microsoft

Weitere Ressourcen

- Mit WMF 5.0: Package Manager!

```
Install-PackageProvider -Name NuGet `
                        -MinimumVersion 2.8.5.201 `
                        -Force
Install-Module -Name xFailOverCluster `
               -RequiredVersion 1.5.0.0 `
               -Force
```

Microsoft

Weitere Ressourcen

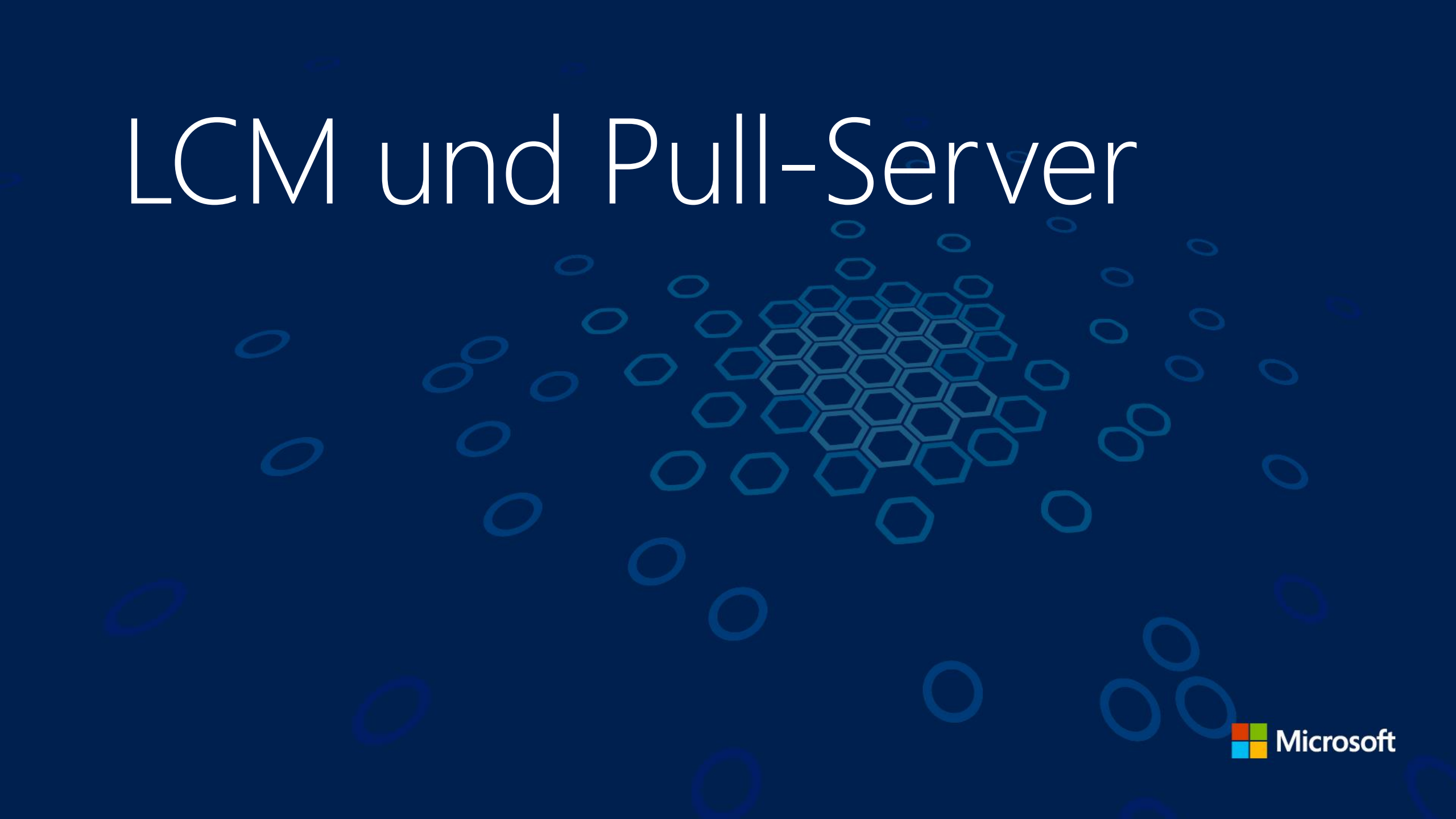
https://github.com/PowerShell/DscResources

xActiveDirectory
xAdcsDeployment
xAzure
xAzurePack
xBitlocker
xCertificate
xChrome
xComputerManagement
xCredSSP
xDFS
xDSCResourceDesigner
xDatabase
xDefender
xDhcpServer
xDismFeature
xDnsServer
xDscDiagnostics
xExchange
xFailOverCluster
xFirefox
xHyper-V
xInternetExplorerHomePage
xJea
xMySql
xNetworking
xPSDesiredStateConfiguration
xPendingReboot
xPhp
xPowerShellExecutionPolicy
xRemoteDesktopAdmin
xRemoteDesktopSessionHost
xRobocopy
xSCDPM
xSCOM
xSCSMA
xSCSPF
xSCSR
xSCVMM
xSQLServer
xSafeHarbor
xSharePoint
xSmbShare
xSqlPs
xStorage
xSystemSecurity
xTimeZone
xWebAdministration
xWebDeploy
xWinEventLog
xWindowsEventForwarding
xWindowsRestore
xWindowsUpdate
xWordPress

Demo

Verwenden von externen Ressourcen und komplexeren Konfigurationen

LCM und Pull-Server



Microsoft

LCM / Pull

- Zum Verwenden eines Pull-Servers muss…
  - … dieser aufgesetzt werden
  - … dem LCM des Ziel-Servers mitgeteilt werden, wie dieser zu erreichen ist

- Geht natürlich komplett mit PowerShell und DSC ;-)
  - Dazu ist die Ressource „xPSDesiredStateConfiguration“ nötig

LCM / Pull

```
Import-DSCResource -ModuleName xPSDesiredStateConfiguration
Import-DscResource -ModuleName PSDesiredStateConfiguration

Node $NodeName
{
    WindowsFeature DSCServiceFeature
    {
        Ensure = 'Present'
        Name   = 'DSC-Service'
    }

    xDscWebService DSCPullSRV
    {
        Ensure                      = 'Present'
        EndpointName                = 'DSCPullSRV'
        Port                        = 8080
        PhysicalPath                = "$env:SystemDrive\inetpub\DSCPullSRV"
        CertificateThumbPrint       = $certificateThumbPrint
        ModulePath                  = "$env:PROGRAMFILES\WindowsPowerShell\DscService\Modules"
        ConfigurationPath           = "$env:PROGRAMFILES\WindowsPowerShell\DscService\Configuration"
        RegistrationKeyPath         = "$env:PROGRAMFILES\WindowsPowerShell\DscService"
        AcceptSelfSignedCertificates = $true
        State                       = 'Started'
        DependsOn                   = '[WindowsFeature]DSCServiceFeature'
        UseSecurityBestPractices = $False # quite new parameter...

    }
}
```

Installation des Pull-Servers mit Zertifikat und Pfade...

LCM / Pull

```
1   [DSCLocalConfigurationManager()]
2   Configuration DscSetPullMode
3   {
4       param(...)
11
12      Node $ComputerName
13      {
14
15          Settings
16          {
17              RefreshMode = 'Pull'
18              RefreshFrequencyMins = 30 |
19              RebootNodeIfNeeded = $true
20              ConfigurationMode = 'ApplyAndAutoCorrect'
21              ConfigurationModeFrequencyMins = 15
22              AllowModuleOverwrite = $true
23          }
24
25          ConfigurationRepositoryWeb DSC-Pull
26          {
27              ServerURL = "https://$($PullServer):8080/PSDSCPullServer.svc"
28              RegistrationKey = $guid
29              CertificateID = $CertThumbprint
30              ConfigurationNames = "$ComputerName-DSC-Config"
31              AllowUnsecureConnection = $true
32          }
33      }
34  }
```

Konfiguration des LCM zur Nutzung des Pull-Servers

https://msdn.microsoft.com/en-us/powershell/dsc/metaconfig

Demo

Setup des Pull-Servers und Ausrollen einer Konfiguration mit diesem

Microsoft

Composite Ressource

Microsoft

Composite Ressources

- DSC-Konfigurationen können als Ressource in andere Konfigurationen eingebunden werden
- Soll erreichen, dass Konfigurationen nicht zu umfangreich werden
- Code-Recycling

Composite Ressources

- Referenzierte Konfiguration muss als **.schema.psm1**. gespeichert werden
- Zusätzlich ist eine Manifest-Datei nötig

Composite Ressources

Verzeichnisstruktur ist wichtig!

```
$env:PSModulePath (C:\Program Files\WindowsPowerShell\Modules)
    |- MyDscResources
            MyDscResources.psd1
        |- DSCResources
            |- EineDemoRessource
                |- EineDemoRessource.psd1
                |- EineDemoRessource.schema.psm1
```

Microsoft

Composite Ressources

Hilfreiches Skript zum Anlegen der Strukturen:

https://gallery.technet.microsoft.com/Helper-Function-to-Create-bf248c68

(„Helper Function to Create a PowerShell DSC Composite Resource")

Demo

Composite Ressource

Microsoft

DSC auf Linux und Nano-Server

Microsoft

DSC auf Linux

Offiziell supported:

- CentOS 5, 6, and 7 (x86/x64)
- Debian GNU/Linux 6 und 7 (x86/x64)
- Oracle Linux 5, 6 und 7 (x86/x64)
- Red Hat Enterprise Linux Server 5, 6 und 7 (x86/x64)
- SUSE Linux Enterprise Server 10, 11 und 12 (x86/x64)
- Ubuntu Server 12.04 LTS und 14.04 LTS (x86/x64)

DSC auf Linux

Etwas höhere Anforderungen ehe man starten kann…

Erforderliches Paket	Beschreibung	Mindestversion
glibc	GNU-Bibliothek	2…4 – 31.30
python	Python	2.4 – 3.4
omiserver	Open Management Infrastructure	1.0.8.1
openssl	OpenSSL-Bibliotheken	0.9.8 oder 1.0
ctypes	Python CTypes-Bibliothek	Muss mit Python-Version übereinstimmen
libcurl	cURL http-Clientbibliothek	7.15.1

Microsoft

DSC auf Linux

- Sowohl Push als auch Pull sind möglich!
- Bei Push: Anmeldeinformationen müssen dem root-User entsprechen!
- An stelle von `Start-DscConfiguration` wird `./StartDscLocalConfigurationManager.py -configurationmof /tmp/localhost.mof` verwendet

Microsoft

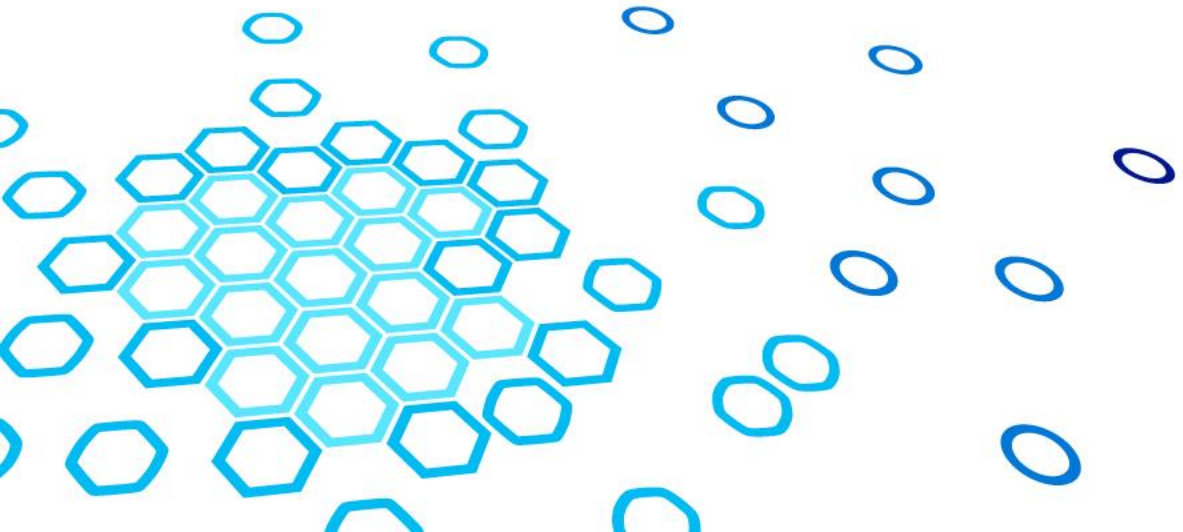
DSC auf Nano-Server

- Wird natürlich unterstützt ;-)
  (wenn auch nicht komplett…)
  https://msdn.microsoft.com/de-de/powershell/dsc/nanodsc
- Allerdings ist ein extra Paket notwendig!
- Beim Erzeugen der VHD:
- -Package Microsoft-NanoServer-DSC-Package

Microsoft

What else?

- In PowerShell 5.0 sind „Partial Configurations“ möglich
  - damit kann z.B. ein „Default Setup“ für alle Server und dann eine zusätzliche Individual-Konfiguration pro Server verwendet werden



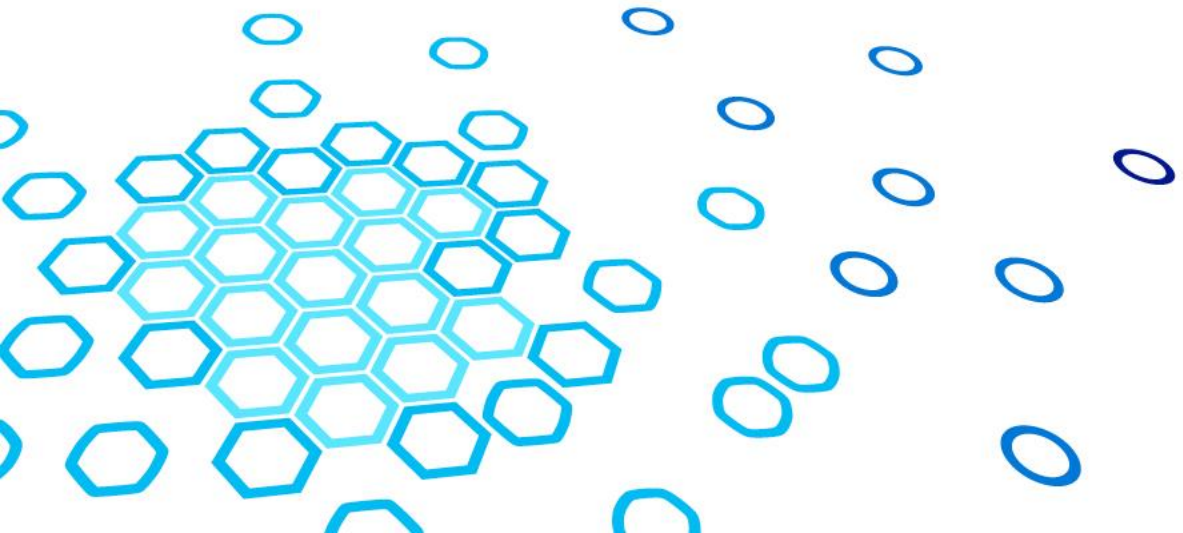
What else?

- Pull-Server wird ggf. „kritische Infrastruktur“ – ggf. über Cluster oder NLB absichern
- MOF-Files sind möglicher Angriffspunkt – absichern! (https://msdn.microsoft.com/en-us/powershell/dsc/securemof)

What else?

Ab Anfang 2017 steht PowerShell DSC auch in der „Deutschen Cloud“ zur Verfügung!

-> Free Trial benutzen…



Microsoft

Gibt es Fragen dazu?

…demnächst wird es auch einen MVA-Kurs zu diesem Thema von mir und Jan-Henrik Damaschke geben!

Vielen Dank

Microsoft

- Sie erreichen mich über:
  - Blog: http://www.hertes.net
  - Twitter: @HHertes
  - Weiteres: https://about.me/haiko.hertes