

PowerShell DSC Einsteigerkurs



Haiko Hertes
Head of IT
Automotive Process Institute GmbH

Jan-Henrik Damaschke
Senior Consultant Security and Cloud
Bright Skies GmbH

02 | Aufbau einer Configuration

Haiko Hertes
Head of IT
Automotive Process Institute GmbH

Jan-Henrik Damaschke
Senior Consultant Security and Cloud
Bright Skies GmbH

DSC-Configuration

- Syntax ist vergleichbar mit klassischem PowerShell-Code
- Einige besondere Keywords
 - Configuration
 - Node
 - Diverse Properties („Name“, „Ensure“, ...)
 - Weitere (später...)

DEMO

Eine erste kleine Beispiel-Konfiguration

DSC-Configuration

- WindowsFeature-Ressource:

Die Ressource **WindowsFeature** in Windows PowerShell DSC (Desired State Configuration) bietet einen Mechanismus, um sicherzustellen, dass Rollen und Features einem Knoten hinzugefügt oder entfernt werden.

Syntax

```
WindowsFeature [string] #ResourceName
{
    Name = [string]
    [ Credential = [PSCredential] ]
    [ Ensure = [string] { Absent | Present } ]
    [ IncludeAllSubFeature = [bool] ]
    [ LogPath = [string] ]
    [ DependsOn = [string[]] ]
    [ Source = [string] ]
}
```



DSC-Configuration

- Es lassen sich auch PowerShell-übliche Elemente benutzen
 - Param()
 - Variablen
 - ...
- Im Wesentlichen das, was bei einer function() möglich ist!

DSC-Configuration

- Weitere „einfachere“ Builtin-Ressources:
 - Service
 - Zum Erzeugen und Steuern von Diensten
 - File
 - Zum Prüfen und Kopieren von Dateien und Verzeichnissen
 - Script
 - Für „fast alles“ verwendbar
 - Test-Script -> Liefert true/false
 - Get-Script -> Liefert den aktuellen Konfigurationsstand
 - Set-Script -> Erzeugt den gewünschten Zustand

DSC-Configuration

- Die Reihenfolge der einzelnen Konfigurationselemente („Instanzen“) spielt erstmal keine Rolle
- Wenn eine bestimmte Reihenfolge nötig ist, kann diese mit „DependsOn“ vorgegeben werden

DEMO

Param() und DependsOn

DEMO

Weitere Ressourcen

DSC-Configuration

```
1 Configuration DemoWebSeite
2 {
3     Node "SRV1"
4     {
5
6         Service W3SVC
7         {
8             Name = 'W3SVC'
9             State = 'Running'
10            StartUpType = 'Automatic'
11            DependsOn = '[WindowsFeature]IIS'
12        }
13
14        File FTPFolder
15        {
16            Ensure = 'Present'
17            Type = 'Directory'
18            DestinationPath = 'C:\FTPFolder'
19        }
20
21        WindowsFeature IIS
22        {
23            Ensure = 'Present'
24            Name = 'Web-Server'
25            IncludeAllSubFeature = $true
26        }
27
28        Script FTPSite1
29        {
30            SetScript = {New-WebFtpSite -Name 'FTPSite' -Port 21 -PhysicalPath 'C:\FTPFolder' -HostHeader 'FTPRoot'}
31            GetScript = {Return @{FTPSite = (Get-Website 'FTPSite')}}
32            TestScript = {Return (Get-Website 'FTPSite')}
33            DependsOn = '[File]FTPFolder', '[WindowsFeature]IIS'
34        }
35    }
36 }
```

DEMO

Mehrere Nodes gleichzeitig

Wie geht es weiter?

- Modul 3: Enacting einer Configuration mittels Push
 - Der Workflow von DSC



Microsoft