

Tastatur statt Maus

XenApp-Administration mit PowerShell

Nicholas Dille, RDS MVP

Deutschsprachige Citrix User Group (DCUG)

Freiburg, den 05.11.2015

Wer ist der Typ?

Ehemann, Vater, Geek, Autor, Aikidoka

Teamlead Enterprise Consulting @ Makro Factory

11+ Jahre Erfahrung im SBC/VDI-Markt

RDS MVP seit 2010

<http://dille.name/blog>

@NicholasDille



Agenda

Citrix Studio



PowerShell



DevOps



XenApp



XenApp / XenDesktop

Umfangreiche PowerShell-API

717 Cmdlets in 41 Modulen

Alle Aspekte von XenApp/XenDesktop sind automatisierbar

Dokumentation: <http://docs.citrix.com/en-us/xenapp-and-xendesktop/7/cds-reference-wrapper-rho/cds-sdk-wrapper-rho/cds-sdk-cmdlet-help.html>

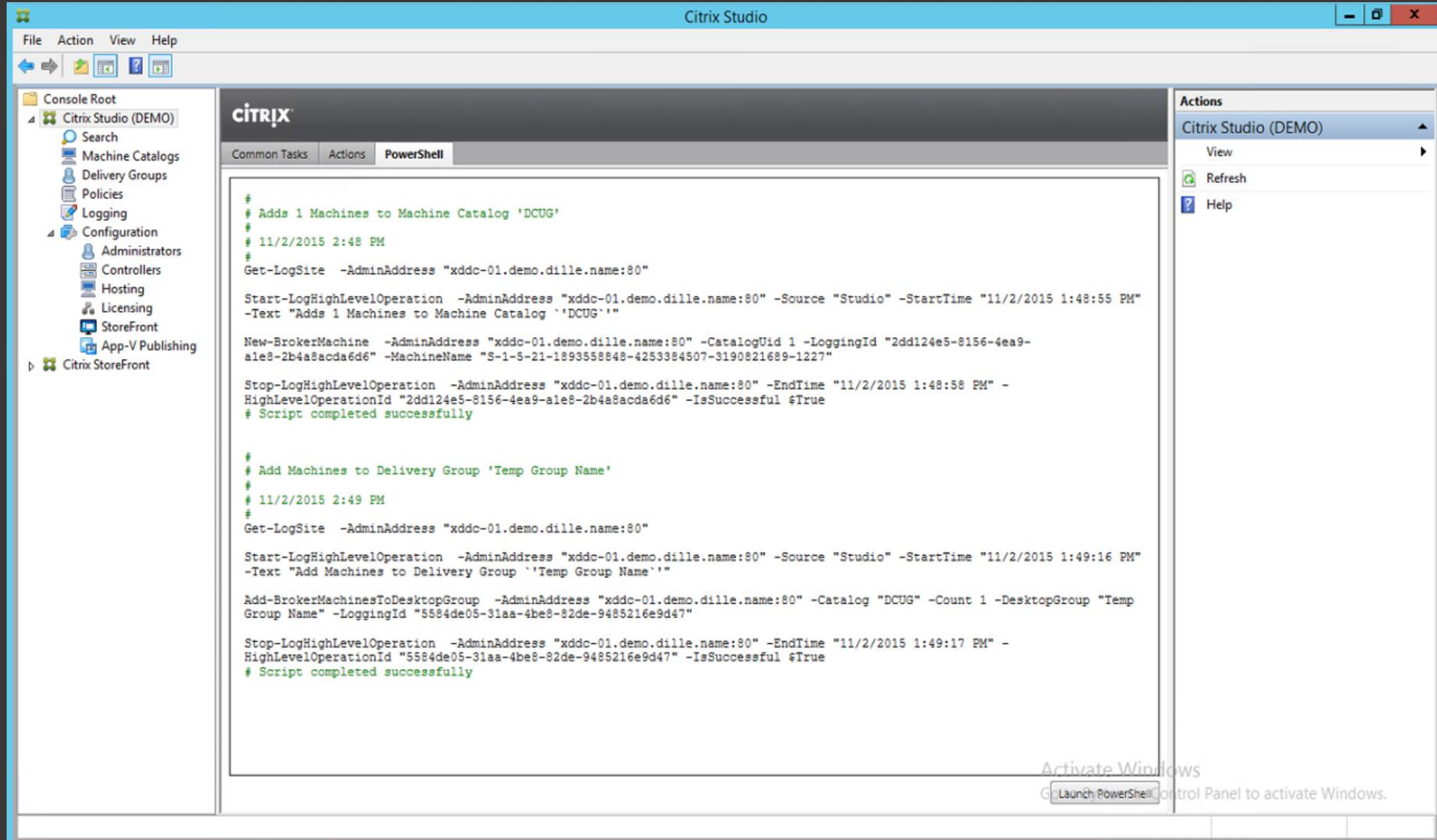
Schnarchlahme Konsole

Geschrieben in .NET

Konsole nutzt PowerShell-API

Design #FAIL

PowerShell-Historie im Studio



The screenshot shows the Citrix Studio interface with the PowerShell console open. The console displays two PowerShell scripts that have been executed successfully. The first script adds a machine to the 'DCUG' catalog, and the second script adds machines to the 'Temp Group Name' delivery group. Both scripts include logging and reporting information.

```
#
# Adds 1 Machines to Machine Catalog 'DCUG'
#
# 11/2/2015 2:48 PM
#
Get-LogSite -AdminAddress "xddc-01.demo.dille.name:80"

Start-LogHighLevelOperation -AdminAddress "xddc-01.demo.dille.name:80" -Source "Studio" -StartTime "11/2/2015 1:48:55 PM"
-Text "Adds 1 Machines to Machine Catalog ``DCUG``"

New-BrokerMachine -AdminAddress "xddc-01.demo.dille.name:80" -CatalogUid 1 -LoggingId "2dd124e5-8156-4ea9-
ale8-2b4a8acda6d6" -MachineName "S-1-5-21-1893558848-4253384507-3190821689-1227"

Stop-LogHighLevelOperation -AdminAddress "xddc-01.demo.dille.name:80" -EndTime "11/2/2015 1:48:58 PM" -
HighLevelOperationId "2dd124e5-8156-4ea9-ale8-2b4a8acda6d6" -IsSuccessful $True
# Script completed successfully

#
# Add Machines to Delivery Group 'Temp Group Name'
#
# 11/2/2015 2:49 PM
#
Get-LogSite -AdminAddress "xddc-01.demo.dille.name:80"

Start-LogHighLevelOperation -AdminAddress "xddc-01.demo.dille.name:80" -Source "Studio" -StartTime "11/2/2015 1:49:16 PM"
-Text "Add Machines to Delivery Group ``Temp Group Name``"

Add-BrokerMachinesToDesktopGroup -AdminAddress "xddc-01.demo.dille.name:80" -Catalog "DCUG" -Count 1 -DesktopGroup "Temp
Group Name" -LoggingId "5584de05-31aa-4be8-82de-9485216e9d47"

Stop-LogHighLevelOperation -AdminAddress "xddc-01.demo.dille.name:80" -EndTime "11/2/2015 1:49:17 PM" -
HighLevelOperationId "5584de05-31aa-4be8-82de-9485216e9d47" -IsSuccessful $True
# Script completed successfully
```

At the bottom of the console, there is a watermark for Windows activation: "Activate Windows Go to Control Panel to activate Windows." and a button labeled "Launch PowerShell".

PowerShell – Hilfe zur Selbsthilfe

Fähigkeiten (Auszug)

Verb-Noun

Objektorientierte Skript-Sprache

Zugriff auf .Net

Remoting

Befehle finden

Get-Command

Get-Help

Get-Member

Show-Command

Navigation

Get-ChildItem, *-Item(Property), *-Path

*-Location, *-PSDrive

Daten durchforsten

Select-Object

Where-Object

Group-Object

Sort-Object

ForEach-Object

Arbeiten mit der Pipeline

Die Pipeline kann als Tabelle visualisiert werden

Jede Zeile ist ein Objekt

Jede Spalte ist eine Eigenschaft des Objekts

Get-ChildItem | Select-Object Name,Extension

Name	Extension
------	-----------

Cmd.exe	Exe
---------	-----

Powershell.exe	Exe
----------------	-----

Shell32.dll	Dll
-------------	-----

Where-Object {\$_.BaseName -ieq 'PowerShell'}

Name	Extension
------	-----------

Powershell.exe	Exe
----------------	-----

Select-Object -First 1

Name	Extension
------	-----------

Cmd.exe	Exe
---------	-----

Group-Object Extension

Count	Extension	Group
-------	-----------	-------

3	Exe	@(@{Name=Cmd.exe,...}, @{Name=PowerShell.exe,...})
---	-----	---

1	Dll	@(@{Name=Shell32.dll,Extension=Dll})
---	-----	--------------------------------------

DevOps

Behandelt Eure Server wie Vieh und nicht wie Haustiere

Standardisierung durch Automatisierung

DevOps hat viele Facetten

Hier: Ops mit Dev-Fähigkeiten

Bspw. Massenänderungen

Demo!

Controller und Worker

Controller verwalten

Informationen auslesen

```
Get-BrokerController | select DNSName,ControllerVersion,State,DesktopsRegistered
```

Worker verwalten

Informationen auslesen

```
Get-BrokerMachine |  
Select-Object DNSName,AgentVersion,CatalogName,DesktopGroupName,RegistrationState,SessionCount
```

Wartungsmodus ändern

```
Get-BrokerMachine |  
Set-BrokerMachine -InMaintenanceMode $true
```

Massenänderungen

```
Get-BrokerMachine | Where-Object {...} | Select-Object DNSName | ForEach-Object {...}
```

Administratoren

Administratoren verwalten

Welche Objekte dürfen verändert werden?

Scope anlegen mit *-AdminScope

Scope hinzufügen mit *-BrokerScope

Welche Rechte werden erteilt?

Rolle anlegen/verändern mit *-AdminRole

Rechte zuweisen mit *-AdminPermission

Wer darf administrieren?

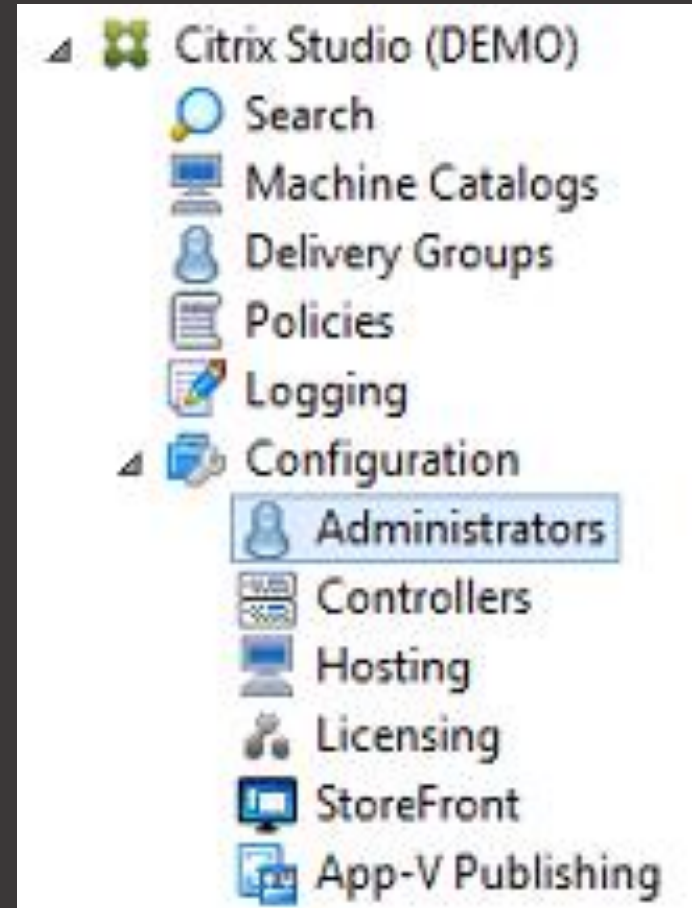
Konten/Gruppen anlegen mit *-AdminAdministrator

Scope und Rolle zuweisen mit *-AdminRight

Testen

Get-AdminEffectiveRight

Test-AdminAccess



Veröffentlichungen

Neue Veröffentlichungsgruppe

New-BrokerDesktopGroup

Get-BrokerMachine | Add-BrokerMachine

New-BrokerAccessPolicyRule

New-BrokerAppEntitlementPolicyRule

Anwendungen verwalten

New-BrokerApplication

Maschinen für Benutzer reservieren

Add-BrokerUser

Sitzungen

Sitzungen auslesen

Get-BrokerSession

Get-BrokerSession -ApplicationInUse 'Word 2016'

Get-BrokerSession -UserName DEMO\ndille

Get-BrokerSession -DesktopGroupName DCUG_Test

Get-BrokerSession -InMaintenanceMode \$true

Get-BrokerSession -SessionState Disconnected

Get-BrokerSession -SessionSupport MultiSession

Verwalten

Disconnect-BrokerSession, Stop-BrokerSession

Send-BrokerSessionMessage

Automatische Neustarts

Builtin

Start-BrokerRebootCycle kann nur alle Worker einer Desktopgruppe neu starten

Anforderungen

Aufteilen der Server in Neustartgruppen

Throttling

Verfügbarkeitstest

Demo!

Remoting ohne Installation der Cmdlets

PowerShell kann Cmdlets eines anderen Hosts importieren

Ziel: Automatisierung ohne Installation

Vorgehensweise:

Öffnen einer neuen Sitzung mit New-PSSession

Importieren von Modulen und Snapins in dieser Sitzung (Import-Module und Add-PSSnapin)

Importieren von Cmdlets aus der Sitzung (Import-PSSession und Export-PSSession)

Demo!

Migration von Maschinenkatalogen

Anforderung

Nutzung von MCS nach Umbenennungen in der Virtualisierungsplattform

Beispiele: XXX

Code auf GitHub

<https://github.com/nicholasdille/XenApp7>

Fazit

Citrix Studio

Langsam
PowerShell-Historie



PowerShell

Hilfe zur Selbsthilfe



DevOps

Vieh nicht Haustiere
Massenänderungen



XenApp

Controller/Worker
Administratoren
Veröffentlichungen
Neustarts
Remoting



Wer ist der Typ?

Ehemann, Vater, Geek, Autor, Aikidoka

Teamlead Enterprise Consulting @ Makro Factory

11+ Jahre Erfahrung im SBC/VDI-Markt

RDS MVP seit 2010

<http://dille.name/blog>

@NicholasDille

