



centric
tech
event

© centric

CRAFT TALKS



centric
tech
event

POWERSHELL YOUR DEVOPS

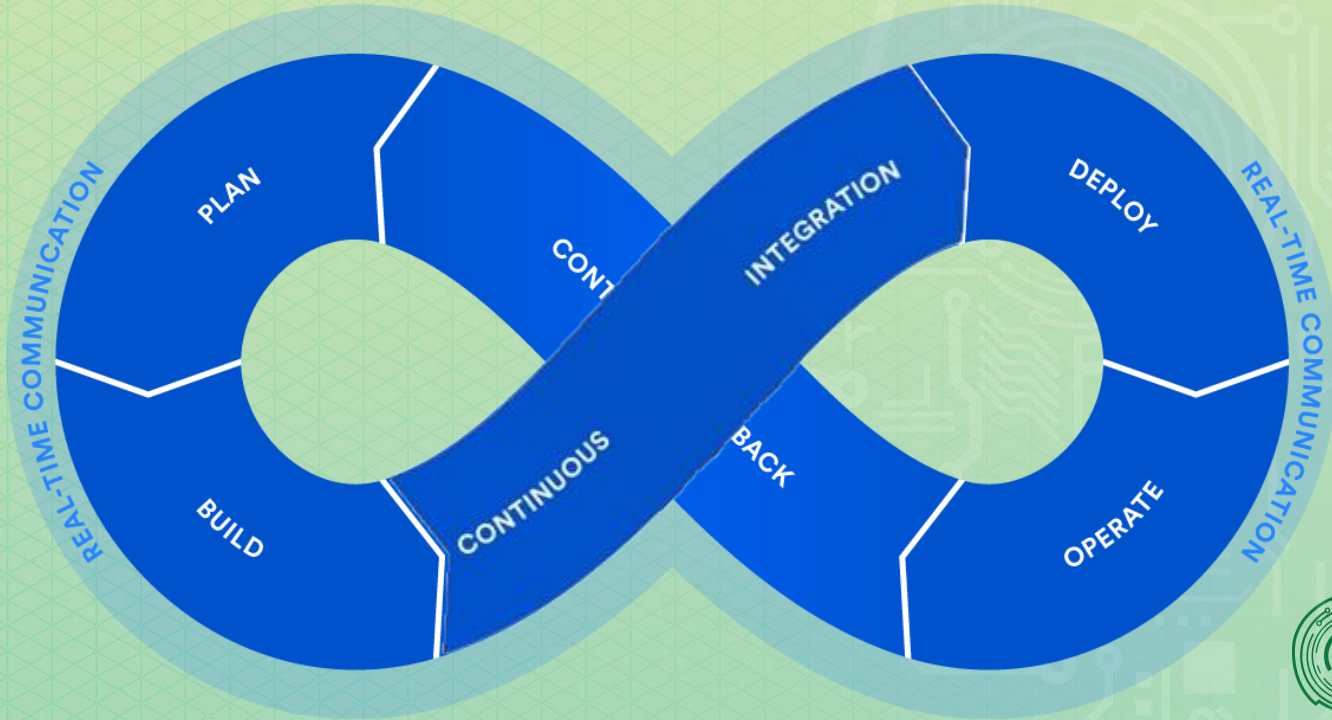


Randolph Widjaja | Architect



centric
tech
event

DevOps



Continuous Integration

- Desired State Configuration (DSC)
- VM Templates incl. applicaties (snapshots)

- DSC + VM Templates
Deployment met PowerShell scripts



PowerShell

- Logging

```
14 # Output Installation Status
15 $OutputResult = "C:\Centric_3rd_Party_Installation.txt"
16
17 # Timestamp function
18 filter timestamp {"$(Get-Date -Format G): $_"}
19
20 # Start Installer Script
21 Write-Output "Start Powershell Installation Script" | Timestamp | Out-File -FilePath $OutputResult -Append
22
```

PowerShell

- Installatie Roles en Features (DSC)

```
1 # Installation Prerequisites
2 Set-ExecutionPolicy Unrestricted -Force
3
4 If ($PSVersionTable.PSVersion | where {$_.Major -ne "4"})
5 {
6   "PowerShellVersion:" | Out-File -FilePath $OutputResult -Append
7   $PSVersionTable.PSVersion | Out-File -FilePath $OutputResult -Append
8   "wrong Powershell version, please upgrade Powershell to v4.0" | Out-File -FilePath $OutputResult -Append
9   "Script failed" | Out-File -FilePath $OutputResult -Append
10  Break
11 }
12
13 $CheckFramework = Get-windowsFeature | where {$_.DisplayName -eq ".NET Framework 3.5 (includes .NET 2.0 and 3.0)"}
14 If ($CheckFramework | where {$_.InstallState -eq "Removed" or "Available"})
15 {
16   Install-windowsFeature Net-Framework-Core -source "$LocationDFSSoftwareRepository\$SxsFolder"
17   Get-windowsFeature | where {$_.DisplayName -eq ".NET Framework 3.5 (includes .NET 2.0 and 3.0)"} | Out-File -FilePath $OutputResult -Append
18 }
19 Else
20 {
21   Get-windowsFeature | where {$_.DisplayName -eq ".NET Framework 3.5 (includes .NET 2.0 and 3.0)"} | Out-File -FilePath $OutputResult -Append
22   ".NET Framework 3.5 (includes .NET 2.0 and 3.0) Installed" | Out-File -FilePath $OutputResult -Append
23 }
24 }
```

PowerShell

- Installatie 3rd party Applicaties

```
20 # Variables 3rd Party Software
21 $apptomatch = "*Microsoft Sync Framework 2.1 Core Components*"
22 |
23 #Installation 3rd Party Software
24 Function Get-InstalledApps
25 {
26 |
27 |     if ([IntPtr]::Size -eq 4) {
28 |         $regpath = 'HKLM:\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\*' | select-object Displayname | where { $_.Displayname -Like $apptomatch}
29 |     }
30 |     else {
31 |         $regpath = @('HKLM:\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\*')
32 |         Get-ItemProperty $regpath | .{process{if($_.Displayname -and $_.UninstallString) { $_ } }} | select Displayname | sort Displayname
33 |     }
34 | }
35 |
36 | $result = Get-InstalledApps | where {$_ .Displayname -like $apptomatch}
37 |
38 | If ($result -eq $null) {
39 |     write-Output "Synchronization-v2.1 not yet Installed" | Timestamp | Out-File -FilePath $OutputResult -Append
40 |     Start-Process -FilePath $LocationDFSsoftwareRepository\Synchronization\Synchronization-v2.1-x86-ENU.msi -Argumentlist "/quiet" -PassThru -wait
41 |     write-Output "Synchronization-v2.1 Installed" | Timestamp | Out-File -FilePath $OutputResult -Append
42 | }
43 | Else {
44 |     write-Output "Synchronization-v2.1 already Installed" | Timestamp | Out-File -FilePath $OutputResult -Append
45 | }
```


PowerShell

- Controleer de externe workflow connecties

```
20 #Variables Network
21 $DMZServer = "www.centric.eu"
22
23 #Check if network connection is available
24 $CheckNetworkPortServer = Test-NetConnection $DMZServer -port 80
25 If($CheckNetworkPortServer | where {$_.TcpTestSucceeded -eq True})
26 {
27     write-Output "Network connection successfull" | Timestamp | Out-File -FilePath $OutputResult -Append
28 }
29 Else
30 {
31     write-Output "Network connection failed, please check the Network connection on port 80" | Timestamp | Out-File -FilePath $OutputResult -Append
32     write-Output "Script halted" | Timestamp | Out-File -FilePath $OutputResult -Append
33     Break
34 }
```

PowerShell

- Reboot Schema

```
1 If (!$SkipToPhaseTwo)
2 {
3     #A bunch of code to do the pre-restart portion of the automation goes here
4     #This code schedules the currently running script to restart and forward to a given checkpoint:
5     schtasks.exe /create /tn "HeadlessRestartTask" /ru SYSTEM /sc ONSTART /tr "powershell.exe -file $($MyInvocation.MyCommand.Definition) -skipToPhaseTwo"
6     write-Host ""$scriptlocation"" is scheduled to run once after reboot."
7     Restart-Computer -Force
8 }
9 write-output "The system has restarted, continuing..."
10 #self-delete the scheduled task
11 "schtasks.exe /delete /f /tn HeadlessRestartTask"
12
13 |
```

Resultaat

- 1500 Servers/Clients ✓
- 30 verschillende server/client rollen ✓
- Deployment in 1,5 uur server/client infrastructuur ✓
- Tijdwinst 5 dagen per deployment cyclus ✓

Doe mee!
www.centric.eu/craft

craft

develop yourself





centric
tech
event

© centric