



# Deployment met PowerShell DSC





# Voorstellen

Eddy De Silva De Freitas  
Infrastructure Engineer/Consultant bij EIC sinds 2014



*20 september 2017 EIC B.V.*



# Wat is PowerShell ?

- Windows PowerShell is een raamwerk waarmee beheer gepleegd kan worden en allerhande taken geautomatiseerd kunnen worden.
- De opvolger van het Command-prompt
- Gebaseerd op .NET Framework en .NET Core waardoor vrijwel alles mogelijk is.
- Belangrijk voor elke beheerder



# Project

- Het vervangen van het huidige SCADA systeem.
  - 25 jaar oud, draaiend op een OpenVMS besturingssysteem.
  - Nieuw SCADA systeem moet gebouwd worden op een volledig Windows platform.



# Omgeving

De nieuwe omgeving bestaat uit twee datacenters, alles redundant uitgevoerd.

Alle componenten per omgeving dubbel uitgevoerd en volledig gebaseerd op een Microsoft architectuur.



# Project teams

- Infrastructuur team
  - Verzorgen infrastructuur: virtualisatielaag, netwerk, AD, Firewall, etc.
- Release management team
  - Verzorgen uitrol van applicaties
- Developers
  - Maken van aanpassingen in SCADA software o.b.v. input van de klant



# Omgevingen

- De omgevingen zijn verdeeld in een DT en AP omgeving. DT en AP zijn fysiek gescheiden
- DT omgeving heeft 9 verschillende omgevingen



# Deployment

- Bij start van het project ging men uit van 3 test omgevingen ingericht met Windows Server 2012 R2.
- Later kwam vanuit development de vraag naar meerdere omgevingen en was Windows Server 2016 ook al beschikbaar.





# Requirements

- Consistentie
  - Infrastructuur omgevingen moeten consistent opgeleverd worden aan RM voor verdere uitrol van applicaties
- Snel geautomatiseerd/herhaalbaar
  - Opleveren van omgeving moet snel kunnen, zodat men niet teveel tijd kwijt is aan wachten in project
  - Omgevingen zijn gelijk aan elkaar, deployment proces is herhaalbaar
  - Geautomatiseerd om menselijke fouten te vermijden
- Flexibel
  - De variabele van de omgevingen moeten makkelijk te verwerken zijn in uitrol.



# Opties

- SCVMM
  - Klant heeft 'Microsoft tenzij' beleid en virtualisatielaag is Hyper-V 2012R2. Deze wordt beheerd met SCVMM.
- PowerShell DSC
  - Microsoft biedt vanaf 2012R2 PowerShell DSC (Desired State Configuration) aan. Hiermee kan men configuraties voor machines aanbieden



# SCVMM

- System Center Virtual Machine Manager
  - SCVMM is onderdeel van Microsoft System Center Suite en wordt gebruikt voor beheer van je data center. Denk hierbij aan de virtualisatielaag componenten: Computer, Network en Storage
  - SCVMM kan VM's geautomatiseerd uitrollen d.m.v service templates



# PowerShell Desired State

- DSC is een configuratiemanagementplatform dat is ingebouwd in Windows, die is gebaseerd op open standaarden. DSC is flexibel genoeg om betrouwbaar en consistent te functioneren in elke fase van de implementatie levenscyclus (ontwikkeling, test, preproductie, productie)
- DSC centreert rond "configuraties". Een configuratie is een gemakkelijk te lezen document dat een omgeving omschrijft die bestaat uit computers ("knooppunten") met specifieke kenmerken. Deze kenmerken kunnen zo eenvoudig zijn als een bepaalde Windows-functie is ingeschakeld of zo ingewikkeld als het implementeren van SharePoint.



# Keuze

- Met behulp van PowerShell i.c.m. SCVMM VM's aanmaken
- DSC zal gebruikt worden voor het configureren van de omgeving: domein aanmaken, cluster configuraties, etc.

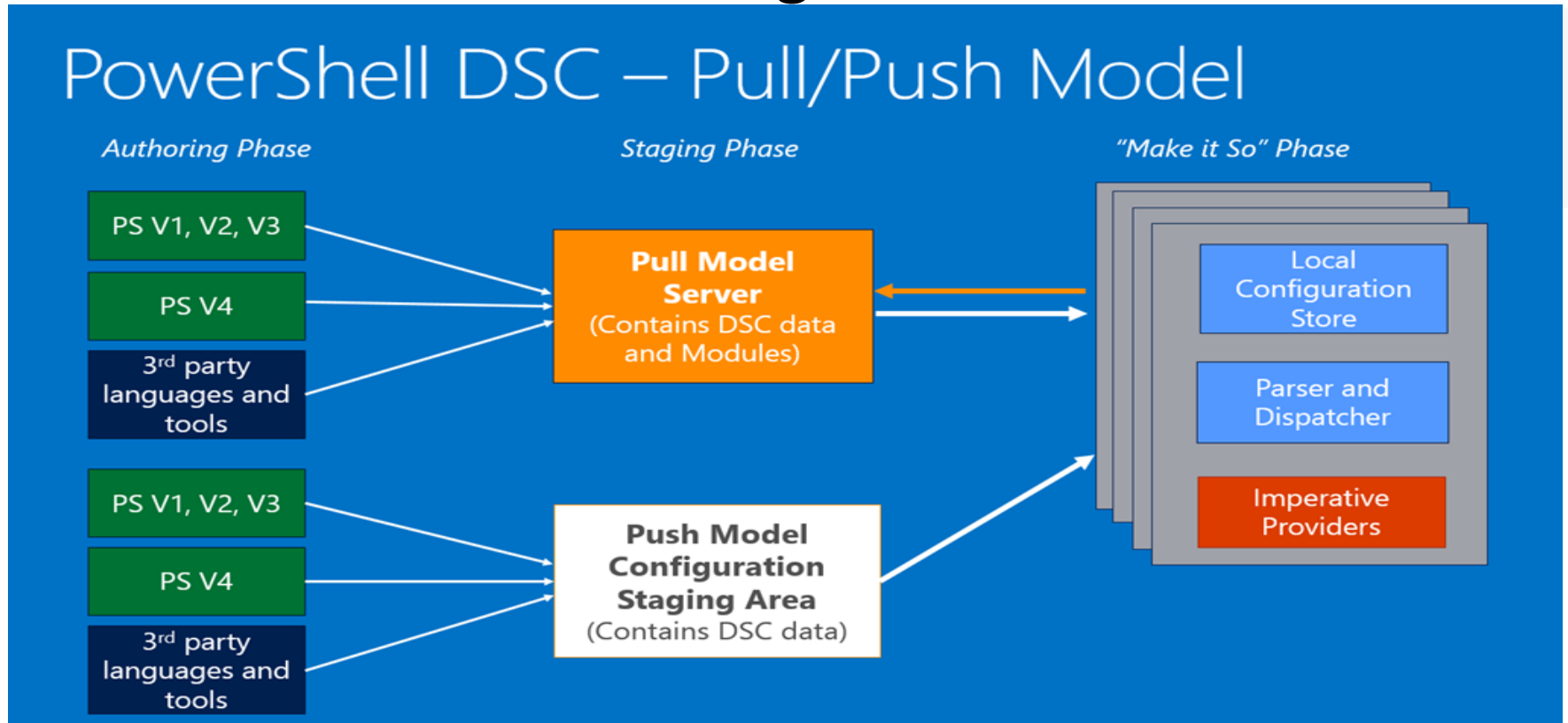


# PowerShell DCS

- DSC is a declarative platform used for configuration, deployment, and management of systems. It consists of three primary components:
  - [Configurations](#) are declarative PowerShell scripts which define and configure instances of resources. Upon running the configuration, DSC (and the resources being called by the configuration) will simply “make it so”, ensuring that the system exists in the state laid out by the configuration. DSC configurations are also idempotent: the Local Configuration Manager (LCM) will continue to ensure that machines are configured in whatever state the configuration declares.
  - [Resources](#) are the "make it so" part of DSC. They contain the code that put and keep the target of a configuration in the specified state. Resources reside in PowerShell modules and can be written to model something as generic as a file or a Windows process, or as specific as an IIS server or a VM running in Azure.
  - The [Local Configuration Manager \(LCM\)](#) is the engine by which DSC facilitates the interaction between resources and configurations. The LCM regularly polls the system using the control flow implemented by resources to ensure that the state defined by a configuration is maintained. If the system is out of state, the LCM makes calls to the code in resources to “make it so” according to the configuration.

# Werking DSC

- DSC Pull vs Push configurations





# DSC Resources

- DSC maakt gebruik van modules (DSC-resources). Deze modules worden gebruikt voor het maken van configuratie
- DSC resources maken het voor beheerders makkelijk om DSC configuratie te maken.
- DSC resources zijn te vinden op GitHub - DSC repository <https://github.com/PowerShell/>
- DSC resources kunnen zelf gemaakt worden.
  - Voor het maken van modules moet er gewerkt worden volgens de richtlijnen die te vinden zijn op GitHub
  - Advies is om zelf gebouwde modules ook te delen op GitHub





# Voorbeeld configuratie

```
Configuration WebsiteTest
```

```
{  
  
    # Import the module that contains the resources we're using.  
    Import-DscResource -ModuleName PsDesiredStateConfiguration  
  
    # The Node statement specifies which targets this configuration will be applied to.  
    Node 'localhost'  
    {  
  
        # The first resource block ensures that the web-Server (IIS) feature is enabled.  
        windowsFeature webServer  
        {  
            Ensure           = "Present"  
            Name              = "Web-Server"  
        }  
  
        # The second resource block ensures that the website content copied to the website root folder.  
        File websiteContent  
        {  
            Ensure           = 'Present'  
            SourcePath       = 'c:\test\index.htm'  
            DestinationPath  = 'c:\inetpub\wwwroot'  
            DependsOn        = "[WindowsFeature]Webserver"  
        }  
    }  
}
```



20 september 2017 EIC B.V.



# PowerShell community

PowerShell is open source.

Op GitHub is er een grote community voor PowerShell, waaronder ook een community voor DSC.

<https://PowerShell.org>



# Test / Ontwikkel

- Testmogelijkheden
  - Makkelijk op eigen client met Hyper-V configuraties testen
  - Met behulp van GitHub Repository gezamenlijk werken aan configuratie (DEV)
  - Azure heeft ook DSC Automation

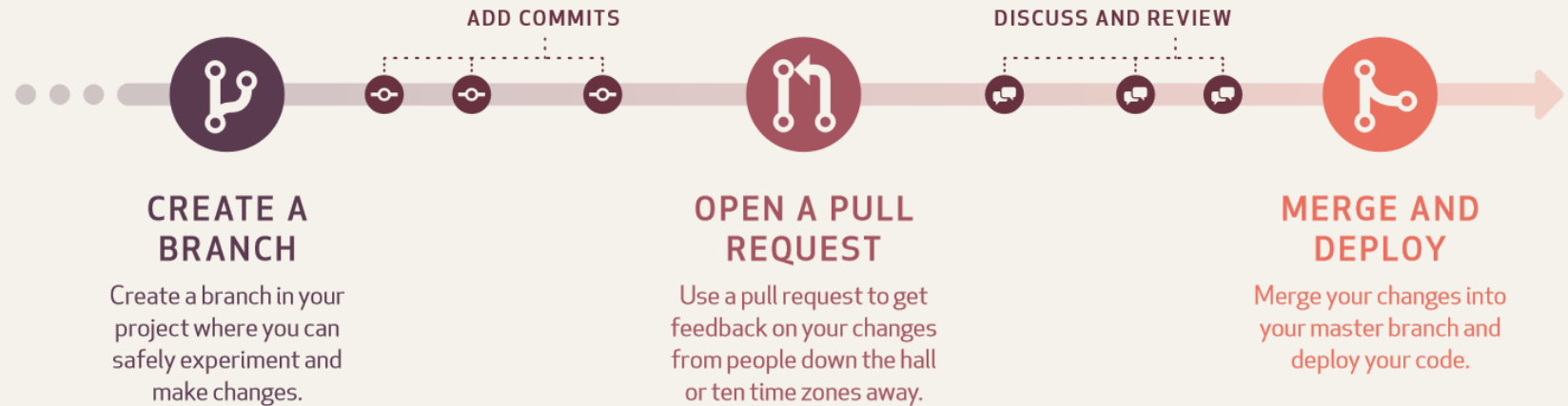


# Werkproces

- De werkwijze van DSC zorgt ervoor dat beheerders meer de ontwikkelaars werkmethode moeten gaan gebruiken.
- Gebruik maken van nieuwe tooling:
  - GitHub
  - Team Foundation Server

# WORK FAST WORK SMART THE GITHUB FLOW

The GitHub Flow is a lightweight, branch-based workflow that's great for teams and projects with regular deployments. Find this and other guides at <http://guides.github.com/>.



GitHub is the best way to build software together.

GitHub provides tools for easier collaboration and code sharing from any device. Start collaborating with millions of developers today!



# Voordelen

- Opleveren omgeving aan RM team binnen halve dag versus 2/3 dagen handmatig
- Verzekerd van resultaat door automatiseren en proces geautomatiseerd te herhalen
- Flexibiliteit door variabele te kunnen gebruiken per omgeving



# Nadelen

- Nieuwe werkwijze voor beheerders  
“DEV/OPS”
- PowerShell kennis bij beheerders
  - DSC kennis beperkt binnen infra team
  - Veel beheerders nog te veel gefocust op GUI
- Rekening houden bij troubleshooten



## Tips

- Goed in kaart brengen waar je mee wilt beginnen
- Maak telkens kleine stappen
- “Niet alles kan geautomatiseerd worden”





# Vragen?



*20 september 2017 EIC B.V.*