SPICE WORLD 2022 HYBRID

Configuration Management with DSC and Windows PowerShell

SPICEWORLD2022
HYBRID

Jeff Hicks

JDH Information Technology Solutions, Inc.
@jeffhicks

SPTCEWORLD2022
HYBRID

Microsoft® Most Valuable Professional

The Configuration Challenge

- How long does it take to configure a server to desired specifications?
  - Corporate standard
  - Regulatory compliance
- How long does it remain in that state?
  - Accidental changes
  - Intentional but undesired changes
  - Expected drift
- How do you know something has changed?
- How can you easily correct server drift?

Configuration Management Process

- Define configuration settings
- Deploy configurations
- Monitor
- Remediate
- Repeat

There are many tools on the market that can manage this process.

Desired State Configuration (DSC)

- Windows PowerShell 5.1 based solution
  - Uses PowerShell Remoting
  - Extensive community support
- Best for small to mid-size organizations
  - Azure has its own version of DSC
- Approach this as a ***framework***
  - Expect to script
  - Fine-tune to your environment
- Did I mention its free?

What About PowerShell 7?

- Microsoft is working on the next generation of DSC
- Many concepts will remain the same
- Underlying technologies might change
- Still a framework
- Will require PowerShell 7.x

What is DSC?

- An extension to the PowerShell language
  - Introduced in v4
  - Uses PowerShell syntax and language
  - Create configuration scripts
- Create and manage server configuration files
  - Use PowerShell create and deploy configurations
- Ensures servers are always configured the way you need
  - A local configuration manager on the server does the heavy lifting

Why DSC?

- Prevent server configuration “drift”
- Separate configuration from implementation
- “Continuous” server deployment
- Manage servers on-site or in a cloud
- Leverage your existing PowerShell skills

Requirements

- Windows PowerShell 5.1 (desktop and server)
- PowerShell remoting enabled
- Active Directory domain (preferred for ease of use)
- A public key infrastructure to handle credentials

DSC Process

- Authoring Phase (client-side)
  - Can include imperative and declarative commands
  - Create MOF definitions
- Staging Phase (client-side)
  - Declarative MOFs staged to managed nodes
  - Configuration calculaated per node
- “Make It So” Phase (server-side)
  - Declarative configurations implemented through imperative providers
  - You don’t have to worry about it

Deployment Options

- Pull
  - Requires a centralized server
  - Managed nodes configured to pull configurations and resources on a schedule
  - Ideal for large organizations
  - I don’t recommend today
- Push
  - Copy configurations from client (or management server) to managed nodes
  - DSC Resources installed on the managed nodes
  - Deploy configurations on your schedule

Managing Configurations

- Created with a PowerShell command
  - One configuration can create multiple and unique MOFs for different servers
- Generate one MOF per server (managed node)
- Apply configurations as often as necessary
- Last applied configuration “wins”
- Configurations managed on the server by the Local Configuration Manager (LCM)
  - Installed by default with Windows PowerShell
  - Configure the LCM with DSC

basic.ps1*

```powershell
configuration BasicServer {
    Param([string[]]$Computername)

    #always import this
    Import-DscResource -ModuleName PSDesiredStateConfiguration -ModuleVersion 1.1
    #import required modules
    Import-DSCResource -ModuleName ComputerManagementDSC -ModuleVersion 8.5.0

    Node $Computername {

        File Work {
            Ensure          = 'Present'
            DestinationPath = 'C:\Work'
            Type            = 'Directory'
        }

        File Readme {
            Ensure          = 'Present'
            DependsOn       = '[File]Work'
            DestinationPath = 'C:\work\Readme.txt'
            Contents        = 'Company work items go in this folder.'
            Type            = 'File'
        }

        smbShare Work {
            Name         = 'CorpWork'
            Path         = 'C:\Work'
            Ensure       = 'Present'
            DependsOn    = '[File]Work'
            Description  = 'Corporate work share'
            FullAccess   = 'Company\Domain Admins'
        }

        WindowsFeature PSv2 {
            Name = 'PowerShell-V2'
            Ensure = 'Absent'
        }

        WindowsFeature Backup {
            Name = 'Windows-Server-Backup'
            Ensure = 'Present'
        }
    } #node
```

DSC Resources

- Managed element you define in your configuration
- Core resources shipped “out of the box”
  - Packaged as a PowerShell module
- Install additional resources from the PowerShell Gallery
  - Standard Microsoft resources
    - May be marked with an *x* prefix
  - Community maintained resources
    - May be marked with a *c* prefix of *DSC* suffix
- You can write your own
  - MOF-based
  - Class-based

Show Me

SPICEWORLD2022
HYBRID

Advanced Stuff

- Partial Configurations
- Composite Configurations
- Credentials
- Custom DSC Resources
- Reporting
- Pester infrastructure testing

Consider DSC a framework. You will need to provide tooling, practices, and policies.

Resources

The DSC Community

- https://dsccommunity.org/

DSC Overview

- https://docs.microsoft.com/powershell/scripting/dsc/overview

Pluralsight

- https://pluralsight.pxf.io/dsc

PSAutoLab

- https://github.com/pluralsight/PS-AutoLab-Env
- Install-Module PSAutolab

Find Me Online

@jeffhicks

https://jdhitsolutions.com/blog

https://github.com/jdhitsolutions

https://jeffhicks.substack.com

https://pluralsight.pxf.io/jeffhicks

Thank you.

SPICEWORLD2022
HYBRID