Advanced DSC Techniques

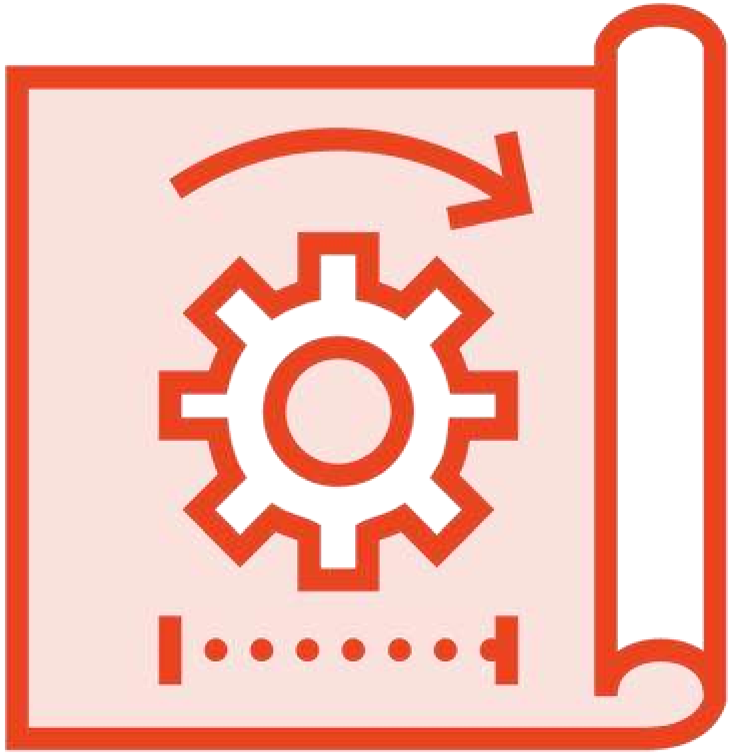
Jeff Hicks

Author/Teacher

@jeffhicks | https://jdhitsolutions.com/blog

You can have a mix of push and pull configurations

Using Named Configurations



Pulled configurations use a configuration ID

- Difficult to identify or associate

Create a *named* configuration

- Still one MOF per server

Easier to deploy role-based configurations

https://docs.microsoft.com/powershell/scripting/dsc/pull-server/pullclientconfignames

```
PS C:\> BasicCompanyServer -computername SRV1 -OutputPath C:\DSC\BasicCompanyServer
```

Using Named Configurations

Create the artifact as usual

```
PS C:\> BasicCompanyServer -computername SRV1 -OutputPath C:\DSC\BasicCompanyServer
PS C:\> Rename-Item -Path C:\DSC\BasicCompanyServer\SRV1.mof -NewName
BasicCompanyServer.mof
```

Using Named Configurations

Create the artifact as usual
Rename the file with your configuration name

```
PS C:\> BasicCompanyServer -computername SRV1 -OutputPath C:\DSC\BasicCompanyServer
PS C:\> Rename-Item -Path C:\DSC\BasicCompanyServer\SRV1.mof -NewName
BasicCompanyServer.mof
PS C:\> New-DscChecksum -Path C:\DSC\BasicCompanyServer\BasicCompanyServer.mof -force
```

Using Named Configurations

Create the artifact as usual
Rename the file with your configuration name
Create a checksum

```
PS C:\> Copy-Item –Path "c:\dsc\basicCompanyServer\basic*" -destination "C:\Program
Files\WindowsPowerShell\DscService\Configuration“ –tosession $s
```

Using Named Configurations

Copy the files to the pull server

```
PS C:\> $myRegKey = (New-Guid).guid
PS C:\> Set-Content -Path C:\DSC\RegistrationKeys.txt -Value $myRegKey
```

Using Named Configurations

Web-based pull servers need a shared secret

Copy RegistrationKeys.txt to “C:\Program Files\WindowsPowerShell\DscService“ on the pull server

Named Configuration LCM

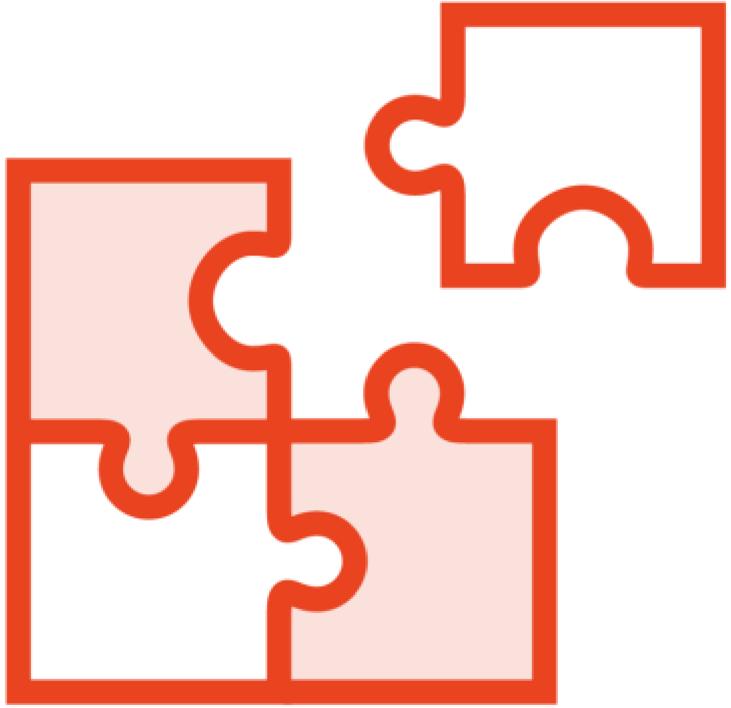
```
Node $ComputerName {
    Settings {
        RebootNodeIfNeeded   = $True
        ActionAfterReboot    = "ContinueConfiguration"
        AllowModuleOverwrite = $True
        ConfigurationMode    = "ApplyAndAutoCorrect"
        RefreshMode          = "Pull"
        RefreshFrequencyMins = 30
        ConfigurationID      = ""
    }

    ConfigurationRepositoryWeb SRV2 {
        ServerURL               = "http://SRV2:8080/PSDSCPullServer.svc"
        AllowUnsecureConnection = $True
        RegistrationKey         = "$RegistrationKey"
        ConfigurationNames      = @("BasicCompanyServer")
    }
}
```

Demo

Using Named DSC Configurations

Partial DSC Configurations



The managed node may have multiple stakeholders

- IT Operations
- Security
- Developers

Each area may want to use DSC

- Create a partial configuration
- Want (or need) to retain ownership

```
Configuration Security { ... }
Configuration Operations { ... }
```

Partial Configurations

Separate configurations have been created
Most likely in separate files that you'll have to coordinate

```
[DSCLocalConfigurationManager()]
configuration PartialConfig {
    Param([string[]]$Computername)

    Node $Computername {
        PartialConfiguration Security {
            Description = 'Configuration to configure system security.'
            RefreshMode = 'Push'
        }
        PartialConfiguration Operations {
            Description = 'Configuration for IT Ops'
            RefreshMode = 'Push'
        }
        Settings {
            RebootNodeIfNeeded   = $True
            ConfigurationMode    = 'ApplyAndAutoCorrect'
            AllowModuleOverwrite = $True
        }
    }
}
```

```
PS C:\> PartialConfig -Computername SRV1 -OutputPath C:\DSCConfigs\PartialDemo
PS C:\> Set-DscLocalConfigurationManager -Path C:\DSCConfigs\PartialDemo
```

Partial Configurations

Set the local configuration manager to use partials.

```
PS C:\> Operations -Computername SRV1 -OutputPath C:\DSCConfigs\Operations
PS C:\> Security -Computername SRV1 -OutputPath C:\DSCConfigs\Security
```

Partial Configurations

Compile the partial configurations

```
PS C:\> Publish-DscConfiguration -Path C:\DSCConfigs\Security
PS C:\> Publish-DscConfiguration -Path C:\DSCConfigs\Operations
```

Partial Configurations

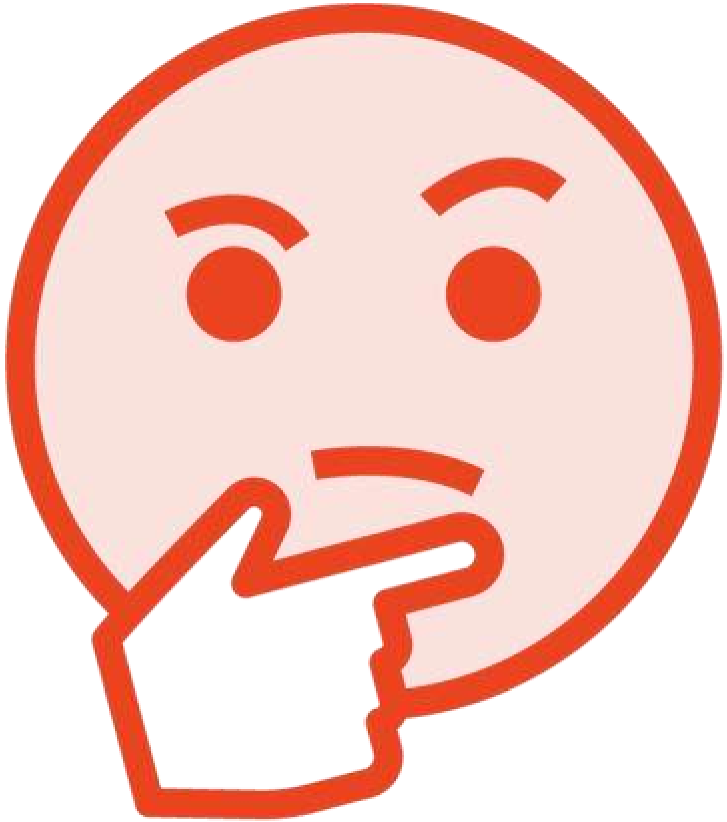
Publish the partial configurations to the node

```
PS C:\> Start-DscConfiguration -ComputerName SRV1 -Wait -UseExisting -Verbose
```

Partial Configurations

Initiate the configuration

Partial DSC Configurations



One configuration per node

Multiple MOFs must be merged into a single MOF

- What if there are conflicts?
- What about cross-dependencies?
- Resource incompatibilities?

There is no way to pre-validate

“Partial configurations are a technical solution to a human problem.”

Opinion

Demo

Partial DSC Configurations

DSC Reporting

You will want to know node status

What is not in compliance?

Your reporting solution may depend on scale

DSC Reporting

Create your own tooling

- Get-DSCConfiguration
- Get-DSCConfigurationStatus
- Test-DSCConfiguration

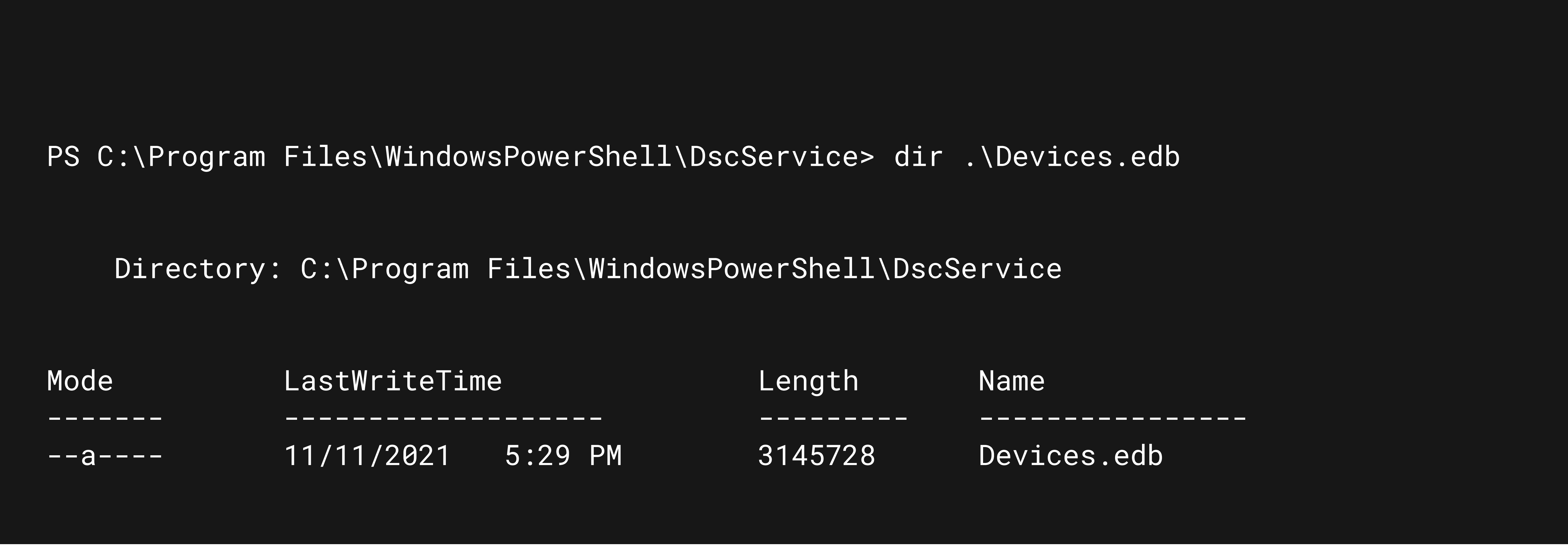
Leverage PowerShell scheduled jobs

Query nodes directly

```
ReportServerWeb SRV2 {
    ServerURL                = 'http://srv2:8080/PSDSCPullServer.svc'
    AllowUnsecureConnection  = $true
    RegistrationKey          = $Key
}
```

DSC Reporting

Enable a reporting server
Requires a DSC Pull server
Configure the node’s local configuration manager



```
PS C:\Program Files\WindowsPowerShell\DscService> dir .\Devices.edb


    Directory: C:\Program Files\WindowsPowerShell\DscService


Mode          LastWriteTime              Length      Name
-------       -------------------        ---------   ----------------
--a----       11/11/2021   5:29 PM       3145728     Devices.edb
```

DSC Reporting

Reporting results stored in a database
Mostly the same information you can get directly from the node
Requires advanced PowerShell scripting

Warnings

Microsoft changed the database format in v5

The jet database can quickly grow out of control

There are no management tools

Not fault-tolerant

Consider the SQL Server back-end option if setting up a pull server

Demo

DSC Reporting

Jeff’s DSC Suggestions

Focus on value

- Where is DSC worth your time?
- DSC technology alone is not enough

Group Policy is still a thing

Build a Push infrastructure

Always keep security in mind

Keep an eye on the future

- Azure Automation
- PowerShell DSC 2.0

Course Summary

Configuration management is a key element for IT Operations

Windows PowerShell DSC is a management framework you can use today

Identify resources

Build configurations

Deploy to managed nodes

“Make it so!”