DSC Fundamentals

Jeff Hicks
Author/Teacher

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

DSC Requirements

Windows PowerShell 5.1

- PSDesiredStateConfiguration module

PowerShell Remoting

- Administrator access

Access to the PowerShell Gallery

Public Key Infrastructure

DSC Requirements

Everything should be enabled by default

- Windows Server 2016 and later
- Windows 10 or later for authoring

Use remote CIMSession

- Computername should be sufficient
- Create a CIMsession for custom connections

Will need to secure configurations

- File permissions
- Use change control

DSC Requirements

DSC requires planning and coordination

Configuration can be separate from implementation

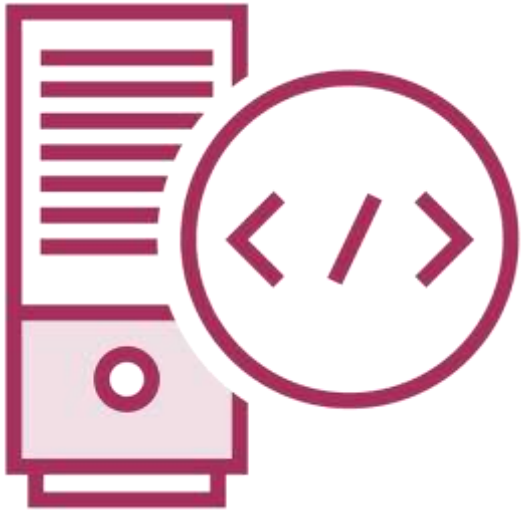
Limitations

DSC does not install the operating system

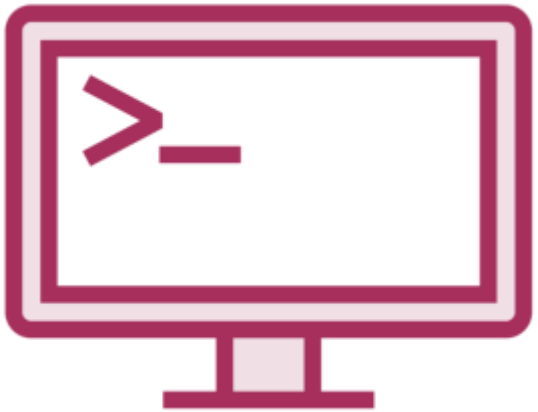
DSC resources must run unattended

There are no default graphical tools

DSC Terminology



Node



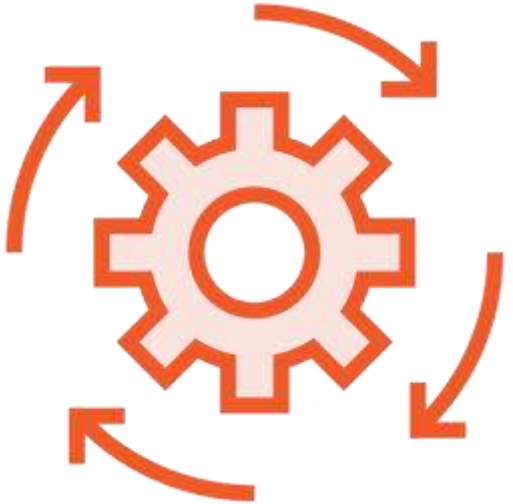
Authoring Node

Configuration

DSC Terminology

Resource

Configuration Document



Local Configuration Manager (LCM)

DSC Resources

Packaged as a PowerShell module

Implements the desired configuration

Used in the configuration command

DSC Resources

Out-of-the-Box

Install from the PowerShell Gallery

Must be installed on target nodes
Should be installed on authoring node

DSC Resources

Experimental

xNetworking
xWebAdministration

Community

cExecutionPolicy
cFolderQuota

Standard

ComputerManagementDSC
DNSServerDSC

DSC Resources

DSC Resources are typically open source – review the code!

Resources may have been superseded or deprecated

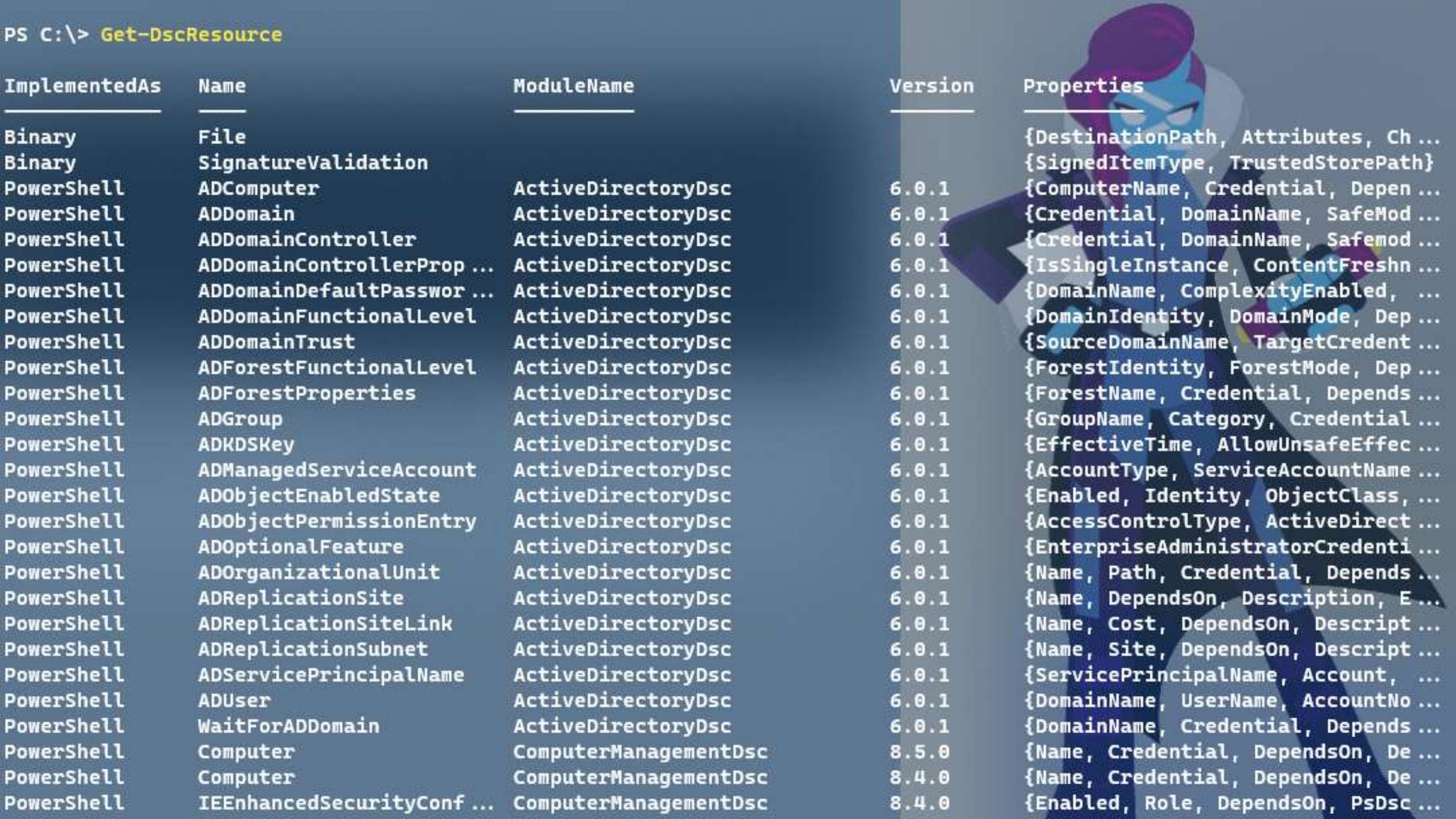
Pay attention to version numbers

TEST EVERYTHING!

```
PS C:\> Get-DSCResource
```

DSC Resources

Get *installed* DSC resource modules



```
PS C:\> Get-DscResource

ImplementedAs   Name                    ModuleName                Version    Properties
-------------   ----                    ----------                -------    ----------
Binary          File                                                         {DestinationPath, Attributes, Ch...
Binary          SignatureValidation                                          {SignedItemType, TrustedStorePath}
PowerShell      ADComputer              ActiveDirectoryDsc        6.0.1      {ComputerName, Credential, Depen...
PowerShell      ADDomain                ActiveDirectoryDsc        6.0.1      {Credential, DomainName, SafeMod...
PowerShell      ADDomainController      ActiveDirectoryDsc        6.0.1      {Credential, DomainName, Safemod...
PowerShell      ADDomainControllerProp... ActiveDirectoryDsc      6.0.1      {IsSingleInstance, ContentFreshn...
PowerShell      ADDomainDefaultPasswor... ActiveDirectoryDsc      6.0.1      {DomainName, ComplexityEnabled,  ...
PowerShell      ADDomainFunctionalLevel ActiveDirectoryDsc        6.0.1      {DomainIdentity, DomainMode, Dep...
PowerShell      ADDomainTrust           ActiveDirectoryDsc        6.0.1      {SourceDomainName, TargetCredent...
PowerShell      ADForestFunctionalLevel ActiveDirectoryDsc        6.0.1      {ForestIdentity, ForestMode, Dep...
PowerShell      ADForestProperties      ActiveDirectoryDsc        6.0.1      {ForestName, Credential, Depends...
PowerShell      ADGroup                 ActiveDirectoryDsc        6.0.1      {GroupName, Category, Credential...
PowerShell      ADKDSKey                ActiveDirectoryDsc        6.0.1      {EffectiveTime, AllowUnsafeEffec...
PowerShell      ADManagedServiceAccount ActiveDirectoryDsc        6.0.1      {AccountType, ServiceAccountName...
PowerShell      ADObjectEnabledState    ActiveDirectoryDsc        6.0.1      {Enabled, Identity, ObjectClass, ...
PowerShell      ADObjectPermissionEntry ActiveDirectoryDsc        6.0.1      {AccessControlType, ActiveDirect...
PowerShell      ADOptionalFeature       ActiveDirectoryDsc        6.0.1      {EnterpriseAdministratorCredenti...
PowerShell      ADOrganizationalUnit    ActiveDirectoryDsc        6.0.1      {Name, Path, Credential, Depends...
PowerShell      ADReplicationSite       ActiveDirectoryDsc        6.0.1      {Name, DependsOn, Description, E...
PowerShell      ADReplicationSiteLink   ActiveDirectoryDsc        6.0.1      {Name, Cost, DependsOn, Descript...
PowerShell      ADReplicationSubnet     ActiveDirectoryDsc        6.0.1      {Name, Site, DependsOn, Descript...
PowerShell      ADServicePrincipalName  ActiveDirectoryDsc        6.0.1      {ServicePrincipalName, Account,  ...
PowerShell      ADUser                  ActiveDirectoryDsc        6.0.1      {DomainName, UserName, AccountNo...
PowerShell      WaitForADDomain         ActiveDirectoryDsc        6.0.1      {DomainName, Credential, Depends...
PowerShell      Computer                ComputerManagementDsc     8.5.0      {Name, Credential, DependsOn, De...
PowerShell      Computer                ComputerManagementDsc     8.4.0      {Name, Credential, DependsOn, De...
PowerShell      IEEnhancedSecurityConf... ComputerManagementDsc   8.4.0      {Enabled, Role, DependsOn, PsDsc...
```

```
PS C:\> Get-DSCResource –name service -syntax
```

Get DSC Resources

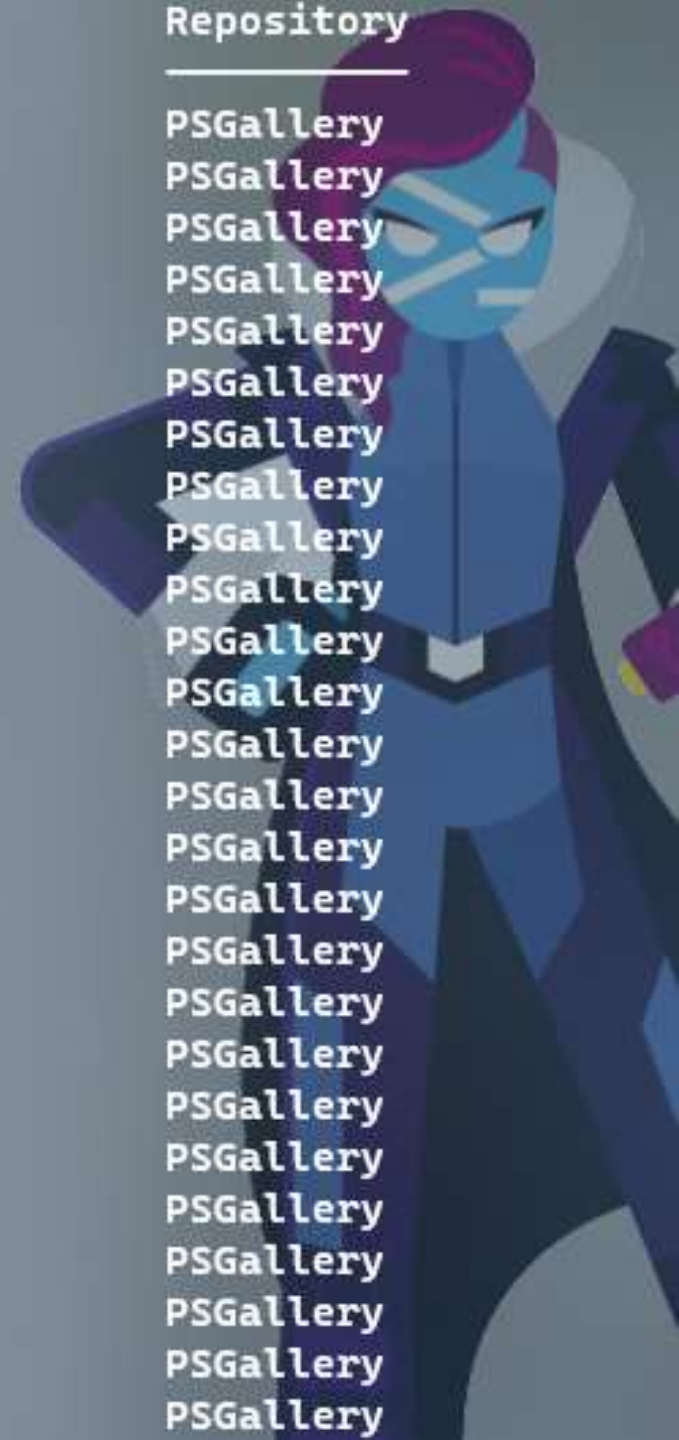
Get resource syntax

```
Service [String] #ResourceName
{
    Name = [string]
    [BuiltInAccount = [string]{ LocalService | LocalSystem |
NetworkService }]
    [Credential = [PSCredential]]
    [Dependencies = [string[]]]
    [DependsOn = [string[]]]
    [Description = [string]]
    [DisplayName = [string]]
    [Ensure = [string]{ Absent | Present }]
    [Path = [string]]
    [PsDscRunAsCredential = [PSCredential]]
    [StartupType = [string]{ Automatic | Disabled | Manual }]
    [State = [string]{ Running | Stopped }]
}
```

```
PS C:\> Find-DSCResource –tag dsc
```

Discover DSC Resources

Name	Version	ModuleName	Repository
DefaultGatewayAddress	8.2.0	NetworkingDsc	PSGallery
DnsClientGlobalSetting	8.2.0	NetworkingDsc	PSGallery
DnsConnectionSuffix	8.2.0	NetworkingDsc	PSGallery
DNSServerAddress	8.2.0	NetworkingDsc	PSGallery
Firewall	8.2.0	NetworkingDsc	PSGallery
FirewallProfile	8.2.0	NetworkingDsc	PSGallery
HostsFile	8.2.0	NetworkingDsc	PSGallery
IPAddress	8.2.0	NetworkingDsc	PSGallery
IPAddressOption	8.2.0	NetworkingDsc	PSGallery
NetAdapterAdvancedProperty	8.2.0	NetworkingDsc	PSGallery
NetAdapterBinding	8.2.0	NetworkingDsc	PSGallery
NetAdapterLso	8.2.0	NetworkingDsc	PSGallery
NetAdapterName	8.2.0	NetworkingDsc	PSGallery
NetAdapterRDMA	8.2.0	NetworkingDsc	PSGallery
NetAdapterRsc	8.2.0	NetworkingDsc	PSGallery
NetAdapterRss	8.2.0	NetworkingDsc	PSGallery
NetAdapterState	8.2.0	NetworkingDsc	PSGallery
NetBIOS	8.2.0	NetworkingDsc	PSGallery
NetConnectionProfile	8.2.0	NetworkingDsc	PSGallery
NetIPInterface	8.2.0	NetworkingDsc	PSGallery
NetworkTeam	8.2.0	NetworkingDsc	PSGallery
NetworkTeamInterface	8.2.0	NetworkingDsc	PSGallery
ProxySettings	8.2.0	NetworkingDsc	PSGallery
Route	8.2.0	NetworkingDsc	PSGallery
WINSSetting	8.2.0	NetworkingDsc	PSGallery
Computer	8.5.0	ComputerManagementDsc	PSGallery



```
PS C:\> Find-DSCResource –name xwebsite
```

Discover DSC Resources

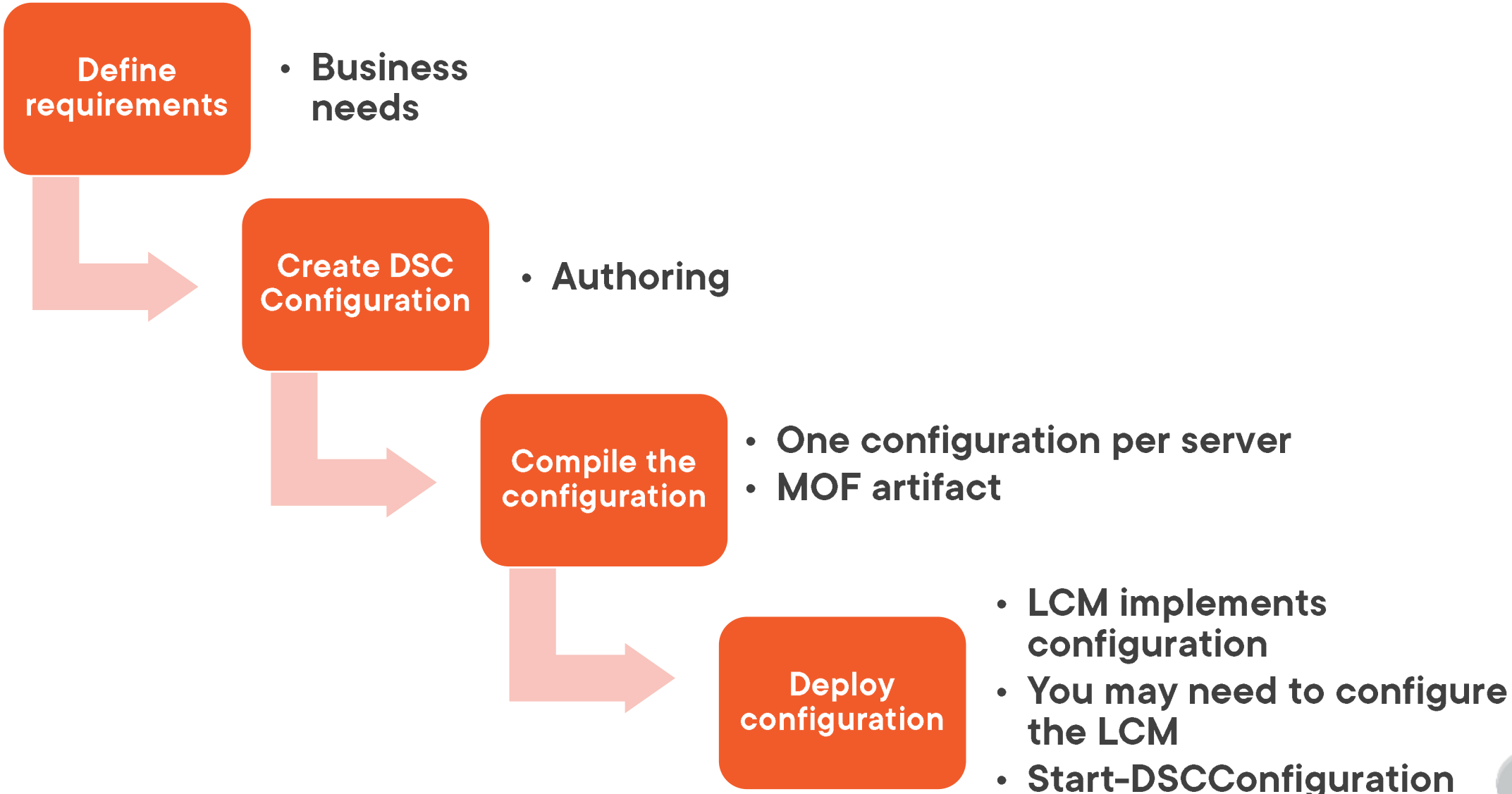
```
PS C:\> Find-DSCResource –name xwebsite

Name                Version    ModuleName           Repository
----                -------    ----------           ----------
xWebSite            3.2.0      xWebAdministration   PSGallery
```

Discover DSC Resources

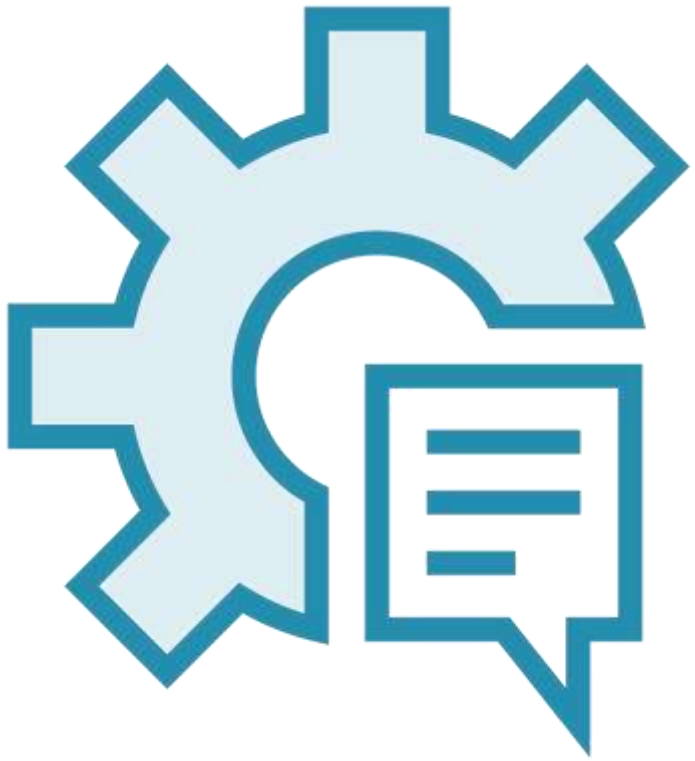
```
PS C:\> Find-DSCResource –name xwebsite

Name                Version    ModuleName           Repository
----                -------    ----------           ----------
xWebSite            3.2.0      xWebAdministration   PSGallery

PS C:\> Install-Module xWebAdministration
```

Discover DSC Resources

DSC Concepts



DSC Concepts



Configurations can be managed per node

- ApplyOnly
- ApplyAndMonitor
- ApplyAndAutoCorrect
- MonitorOnly

DSC Concepts

One configuration per server

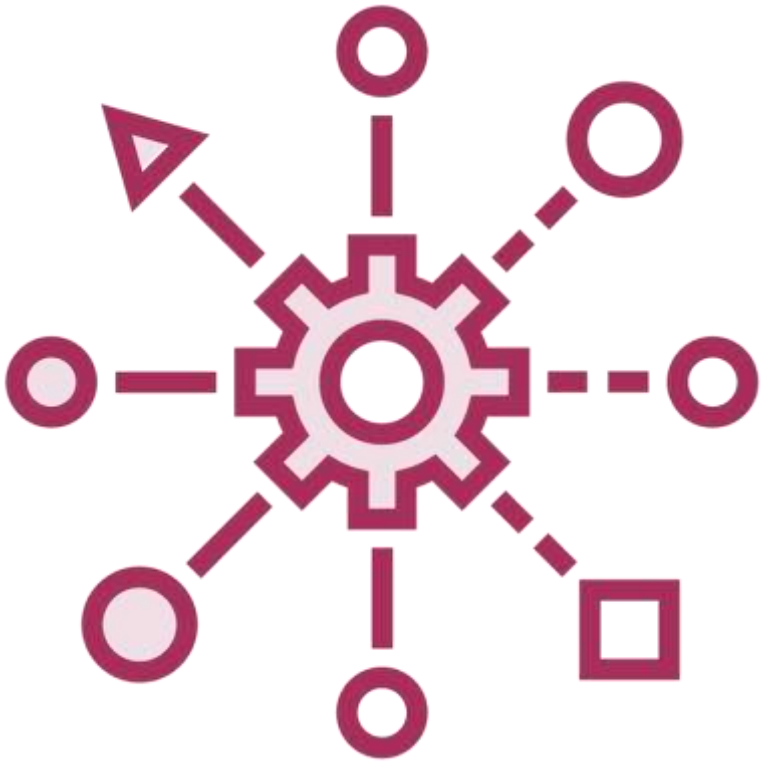
Last configuration applied "wins"

- Configurations can un-do settings

Configuration files are critical and sensitive

Build test and remediation tools

DSC Concepts



Configurations are deployed manually

- Unless you build your own tooling

Nodes can be configured to accept configurations in two ways

- Push (default)
- Pull
- Disabled

RefreshMode

Push	Pull
No central server required	Requires a centralized IIS server
Only push the configuration	Node can pull configuration and required resources
You can push as often as you need	Pull on a timed basis
Easy to set up	More complicated
	We'll look at pull servers later in the course

Demo

DSC in Action

View DSC as a framework

You will need to build out to meet your needs

PowerShell is your likely toolset

Ready to see how to do all of this?