



# **Configuring PowerShell Desired State Configuration Pull Server**

Michael Grafnetter

Mainstream Technologies

# Reminder: PowerShell DSC

```
Configuration WebsiteConfig
{
    Node MyWebServer
    {
        WindowsFeature IIS
        {
            Ensure = "Present"
            Name = "Web-Server"
        }
        Website MyWebsite
        {
            Ensure = "Present"
            Name = "MyWebsite"
            PhysicalPath = "C:\Inetpub\MyWebsite"
            State = "Started"
            Protocol = @"http"
            BindingInfo = @"*:80:"
        }
    }
}
```

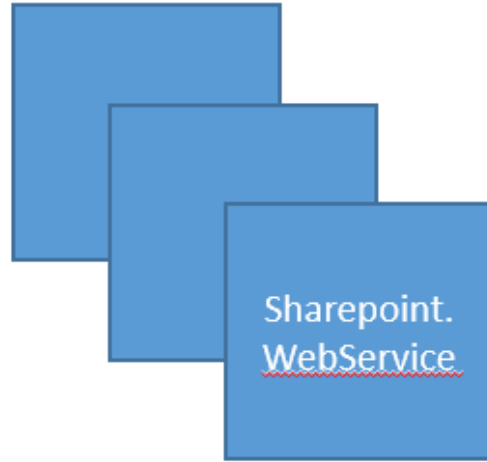
# PowerShell DSC LifeCycle

## Configurations

```
Configuration SharePoint {  
  Node Webservice {  
    #Install the IIS Role  
    WindowsFeature IIS {  
      Ensure = "Present"  
      Name = "Web-Server"  
    }  
  
    #Install ASP.NET 4.5  
    WindowsFeature ASP {  
      Ensure = "Present"  
      Name = "Web-Asp-Net45"  
    }  
  }  
}
```

## Node Configurations (.MOF configuration documents)

Compiled, put  
on pull server  
(via compilation jobs)



Applied  
(via node pulls)

## Nodes



# DSC Modes

- Push
- **Pull**
  - **File Share (SMB)**
  - **Web Service (HTTP)**
  - **Azure Automation DSC**

# Push Model

*Authoring Phase*  
(May include imperative as well as declarative code)

PS V1, V2, V3

PS V4\*\*\*

3<sup>rd</sup> party languages and tools

*Staging Phase*

- Fully declarative configuration representation using DMTF standard MOF instances
- Configuration is calculated for all nodes

**Configuration Staging Area**  
(Contains DSC data)

*"Make it So" Phase*  
(Declarative configuration is reified through imperative providers.)



\*\*\* When authoring in PowerShell, on top of PSV3 imperative features, PSV4 adds:

- Declarative syntax extensions
- Schema-driven Intellisense
- Schema validation (early-binding)

Providers implement changes:

- Monotonic
- Imperative
- Idempotent



# Push Model Drawbacks

- Not scalable
- All nodes have to be online
- Resource modules must be present on clients
- No central log

# Pull Model

*Authoring Phase*  
(May include imperative as well as declarative code)

PS V1, V2, V3

PS V4\*\*\*

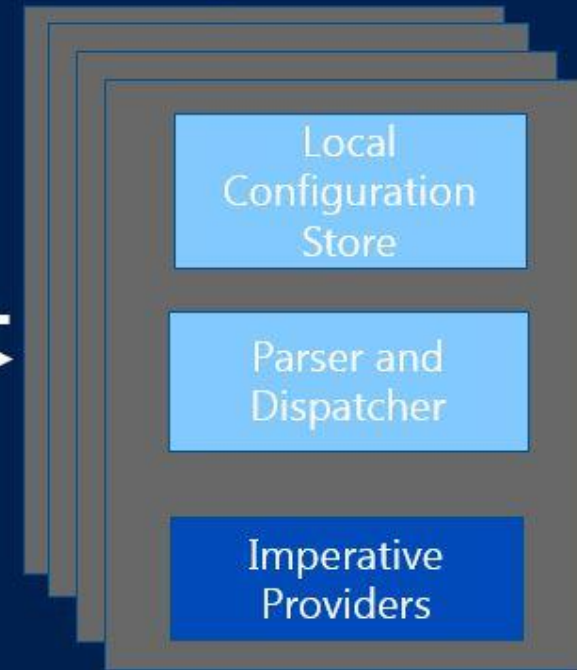
3<sup>rd</sup> party languages and tools

*Staging Phase*

- Fully declarative configuration representation using DMTF standard MOF instances
- Configuration is calculated for all nodes

**Pull Server**  
(Contains DSC data and Modules)

*"Make it So" Phase*  
(Declarative configuration is reified through imperative providers.)



\*\*\* When authoring in PowerShell, on top of PSV3 imperative features, PSV4 adds:

- Declarative syntax extensions
- Schema-driven Intellisense
- Schema validation (early-binding)

Providers implement changes:

- Monotonic
- Imperative
- Idempotent









# **SMB DSC Pull Server**







# DSC Share Contents

## Configuration

 4b7b8f9a-ee7e-429e-b717-311a376ba6cf.mof  
 4b7b8f9a-ee7e-429e-b717-311a376ba6cf.mof.checksum  
 e16daeda-cfb2-43d7-9ee9-93268f6efbf7.mof  
 e16daeda-cfb2-43d7-9ee9-93268f6efbf7.mof.checksum

## Modules

 xTimeZone\_1.3.0.0.zip  
 xTimeZone\_1.3.0.0.zip.checksum  
 xWebAdministration\_1.11.0.0.zip  
 xWebAdministration\_1.11.0.0.zip.checksum

# Creating File Shares using DSC

```
Node $NodeName
{
    File ConfigDirectory
    {
        Type = 'Directory'
        DestinationPath = $RepositoryPath
        Ensure = 'Present'
    }
    xSMBShare ConfigShare
    {
        DependsOn = '[File]ConfigDirectory'
        Name = 'Configuration'
        Path = $RepositoryPath
        FullAccess = 'Everyone'
        Ensure = 'Present'
    }
}
```

# Generating Checksum Files

```
PS > New-DscChecksum -Path C:\DSCRepository -Force -Verbose
VERBOSE: Overwrite checksum file 'C:\DSCRepository\4b7b8f9a-ee7e-429e-b717-311a376ba6cf.mof.checksum'
VERBOSE: Overwrite checksum file 'C:\DSCRepository\cNtfsAccessControl_1.3.0.zip.checksum'
VERBOSE: Overwrite checksum file 'C:\DSCRepository\e16daeda-cfb2-43d7-9ee9-93268f6efbf7.mof.checksum'
VERBOSE: Overwrite checksum file 'C:\DSCRepository\xSmbShare_1.1.0.0.zip.checksum'
VERBOSE: Overwrite checksum file 'C:\DSCRepository\xTimeZone_1.3.0.0.zip.checksum'
VERBOSE: Overwrite checksum file 'C:\DSCRepository\xWebAdministration_1.11.0.0.zip.checksum'
```



# Demo

# Client Configuration

```
[DscLocalConfigurationManager()]
Configuration SMBClient
{
    Node $AllNodes.NodeName
    {
        Settings
        {
            RefreshMode = 'Pull'
            ConfigurationID = $Node.Guid
            ConfigurationMode = 'ApplyAndAutoCorrect'
            ConfigurationModeFrequencyMins = 15
            RefreshFrequencyMins = 30
            RebootNodeIfNeeded = $true
            ActionAfterReboot = 'ContinueConfiguration'
        }
        ConfigurationRepositoryShare SMBRepository
        {
            SourcePath = '\\1on-dc1\DSCRepository'
        }
        ResourceRepositoryShare SMBRepository
        {
            SourcePath = '\\1on-dc1\DSCRepository'
        }
    }
}
```

# Client Configuration

```
@{
  # Node specific data
  AllNodes = @(
    @{
      NodeName = 'LON-SVR1'
      Guid      = "e16daeda-cfb2-43d7-9ee9-93268f6efbf7"
    },
    @{
      NodeName = "LON-SVR2"
      Guid      = "4b7b8f9a-ee7e-429e-b717-311a376ba6cf"
    }
  )
}
```



# Demo

# SMB Pull Server Drawbacks

- Not suitable for internet
- No feedback
- Requires Failover Clustering for HA
- Client IDs are required





# Web DSC Pull Server

# Installing the Web Service

The screenshot shows the 'Add Roles and Features Wizard' in Windows Server. The 'Select features' step is active, with a sidebar on the left containing navigation options: 'Before You Begin', 'Installation Type', 'Server Selection', 'Server Roles', 'Features' (highlighted), 'Web Server Role (IIS)', 'Role Services', 'Confirmation', and 'Results'. The main area displays a list of features to be installed on the selected server. The 'Features' section is expanded, showing a list of features with checkboxes. A red circle highlights the 'Windows PowerShell Desired State Configuration' feature, which is checked and highlighted in blue. Other features in the list include 'Windows Biometric Framework', 'Windows Feedback Forwarder', 'Windows Identity Foundation 3.5', 'Windows Internal Database', 'Windows PowerShell (2 of 5 installed)', 'Windows PowerShell 4.0 (Installed)', 'Windows PowerShell 2.0 Engine', 'Windows PowerShell ISE (Installed)', 'Windows PowerShell Web Access', 'Windows Process Activation Service', 'Windows Search Service', and 'Windows Server Backup'.

**Add Roles and Features Wizard**

## Select features

Before You Begin  
Installation Type  
Server Selection  
Server Roles  
**Features**  
Web Server Role (IIS)  
Role Services  
Confirmation  
Results

Select one or more features to install on the selected server.

### Features

- Windows Biometric Framework
- Windows Feedback Forwarder
- Windows Identity Foundation 3.5
- Windows Internal Database
- Windows PowerShell (2 of 5 installed)
  - Windows PowerShell 4.0 (Installed)
  - Windows PowerShell 2.0 Engine
  - Windows PowerShell Desired State Configuration**
  - Windows PowerShell ISE (Installed)
  - Windows PowerShell Web Access
- Windows Process Activation Service
- Windows Search Service
- Windows Server Backup
- Windows Server Migration Tools

# DSC Web Service Configuration

```
xDscWebService PSDSCPullServer
{
    Ensure           = 'Present'
    EndpointName     = 'DSCPullServer'
    Port             = 8080
    PhysicalPath     = "$env:SystemDrive\inetpub\PSDSCPullServer"
    CertificateThumbPrint = $CertificateThumbPrint
    AcceptSelfSignedCertificates = $true
    State           = 'Started'
    DependsOn      = '[WindowsFeature]DSCServiceFeature'
}
```

# IIS Site

**Connections**

- Start Page
- LON-DC1 (CONTOSO\Administrator) (Server)
- Application Pools
- Sites
  - DSCPullServer**
    - bin
    - en
- Server Farms



## DSCPullServer Content

Filter:



Go



Show All

Group by:

No Grouping

Name

Type



bin

File Folder



Global.asax

ASAX File



PSDSCPullServer.mof

MOF File



PSDSCPullServer.svc

SVC File



PSDSCPullServer.xml

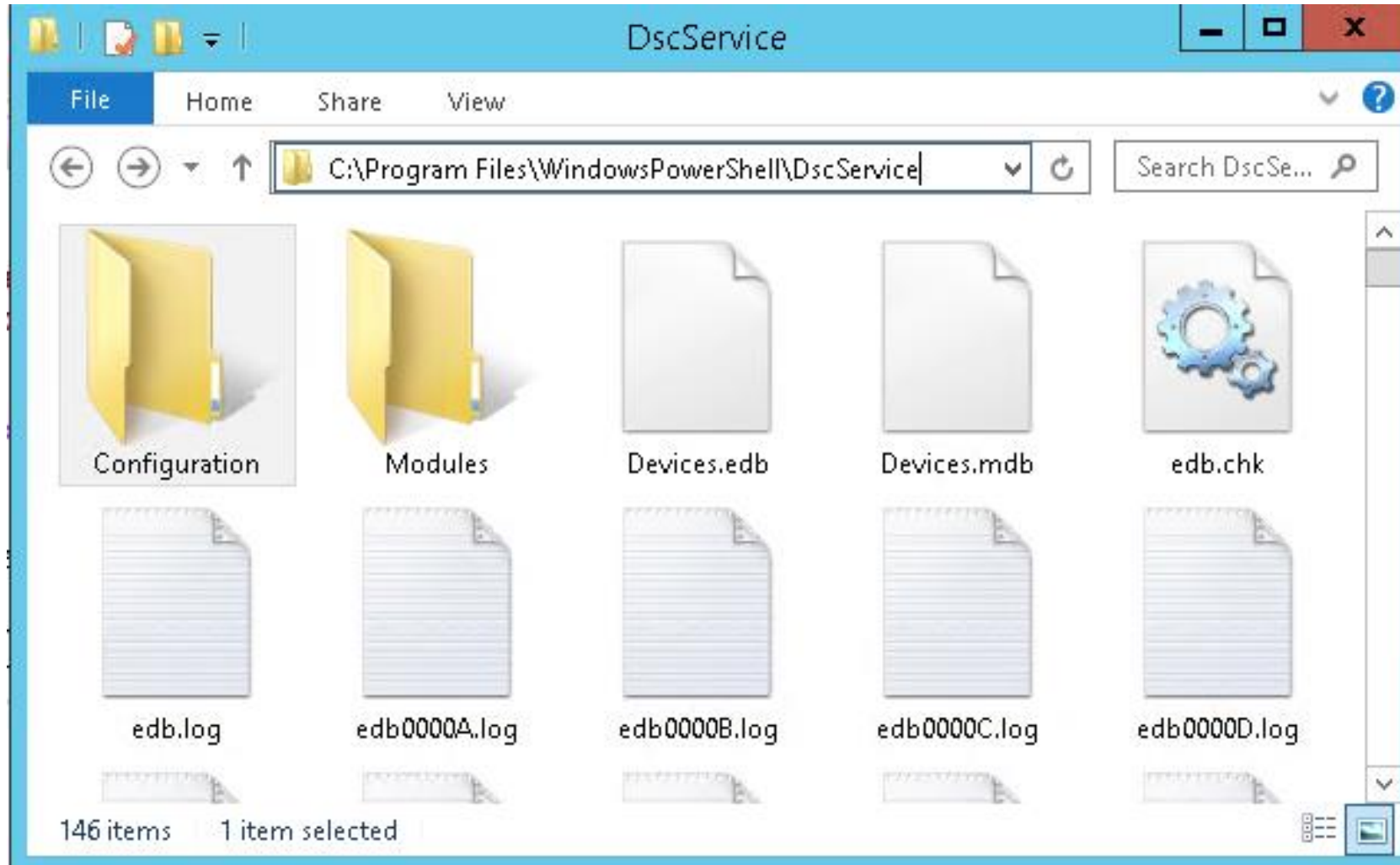
XML Document



web.config

CONFIG File

# Service Directory (Default Path)





# Demo

# Client Configuration

```
[DscLocalConfigurationManager()]
Configuration WebClient
{
    Node $AllNodes.NodeName
    {
        Settings
        {
            RefreshMode = 'Pull'
            ConfigurationID = $Node.Guid
        }
        ConfigurationRepositoryWeb WebRepository
        {
            ServerURL = 'https://lon-dc1.contoso.pri:8080/PSDSCPullServer.svc'
        }
        ResourceRepositoryWeb WebRepository
        {
            ServerURL = 'https://lon-dc1.contoso.pri:8080/PSDSCPullServer.svc'
        }
        ReportServerWeb WebRepository
        {
            ServerURL = 'https://lon-dc1.contoso.pri:8080/PSDSCPullServer.svc'
        }
    }
}
```



# Demo



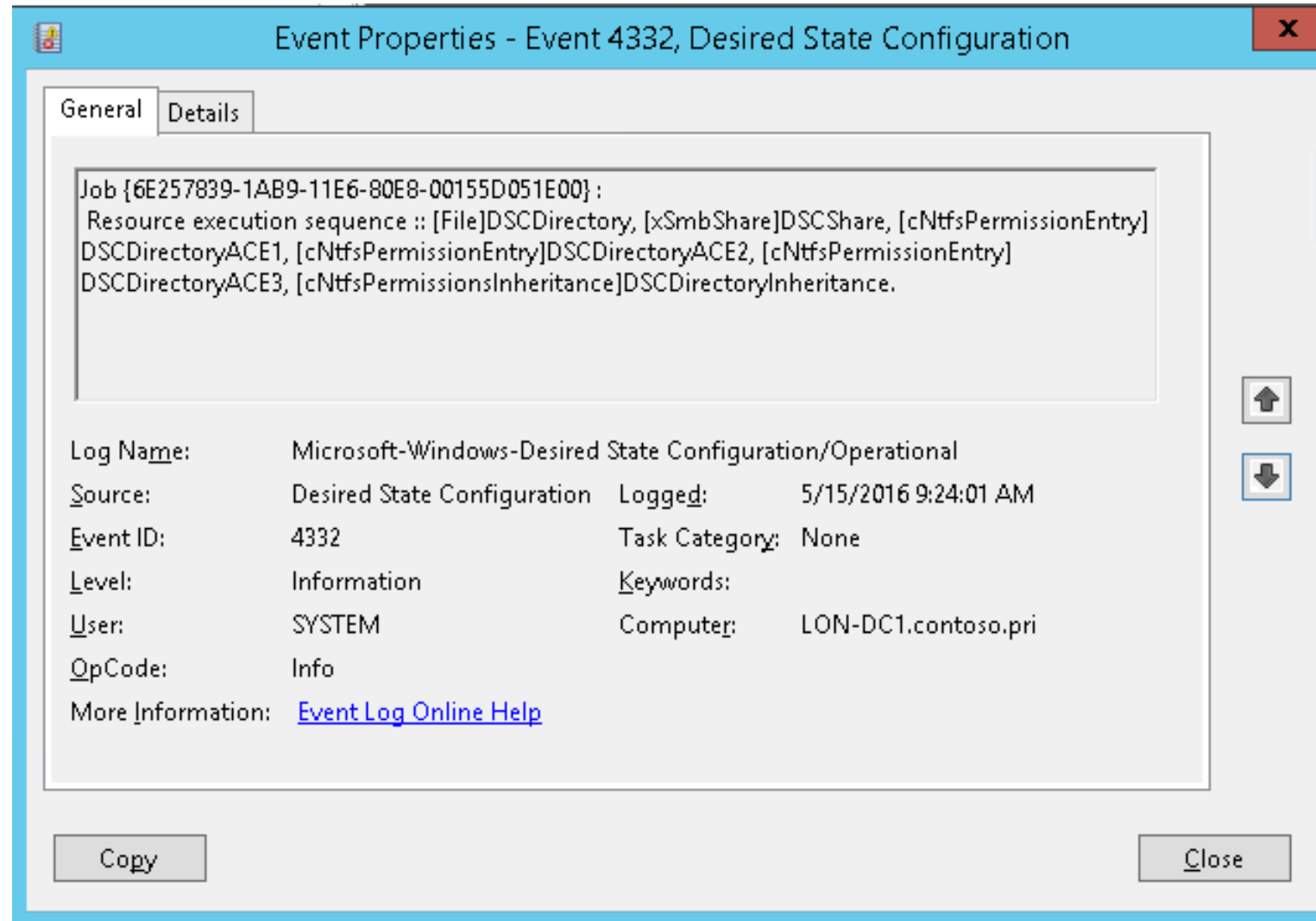
# Monitoring DSC

```
Get-WinEvent -LogName "Microsoft-Windows-Dsc/Operational"
```

```
wevtutil.exe set-log "Microsoft-Windows-Dsc/Analytic" /q:true /e:true
```

```
wevtutil.exe set-log "Microsoft-Windows-Dsc/Debug" /q:True /e:true
```

# Windows Event Log



The screenshot shows the 'Event Properties' dialog box for Event 4332, 'Desired State Configuration'. The 'General' tab is selected, displaying the following information:

**Job {6E257839-1AB9-11E6-80E8-00155D051E00} :**  
Resource execution sequence :: [File]DSCDirectory, [xSmbShare]DSCShare, [cNtfsPermissionEntry]DSCDirectoryACE1, [cNtfsPermissionEntry]DSCDirectoryACE2, [cNtfsPermissionEntry]DSCDirectoryACE3, [cNtfsPermissionsInheritance]DSCDirectoryInheritance.

**Log Name:** Microsoft-Windows-Desired State Configuration/Operational  
**Source:** Desired State Configuration    **Logged:** 5/15/2016 9:24:01 AM  
**Event ID:** 4332    **Task Category:** None  
**Level:** Information    **Keywords:**  
**User:** SYSTEM    **Computer:** LON-DC1.contoso.pri  
**OpCode:** Info  
**More Information:** [Event Log Online Help](#)

Buttons: Copy, Close

# Reporting

Out-GridView

Filter

+ Add criteria

StartDate	HostName	Status	Type	DurationInSeconds	RebootRequested
2016-05-15T18:07:13.3880000+02:00	LON-SVR1	Success	Initial	0	False
2016-05-15T18:08:04.8820000+02:00	LON-SVR1	Success	LocalConfigurationManager	0	False
2016-05-15T18:08:05.1590000+02:00	LON-SVR1	Failure	Initial	0	False
2016-05-15T18:08:05.3630000+02:00	LON-SVR1	Success	Initial	0	False
2016-05-15T18:08:16.0810000+02:00	LON-SVR1	Success	LocalConfigurationManager	0	False
2016-05-15T18:08:16.1790000+02:00	LON-SVR1	Failure	Initial	0	False
2016-05-15T18:09:00.0840000+02:00	LON-SVR1	Success	LocalConfigurationManager	0	False
2016-05-15T18:09:00.5140000+02:00	LON-SVR1	Success	Initial	0	False
2016-05-15T18:14:58.5550000+02:00	LON-SVR1	Success	LocalConfigurationManager	0	False
2016-05-15T18:14:59.0740000+02:00	LON-SVR1	Success	Initial	0	False

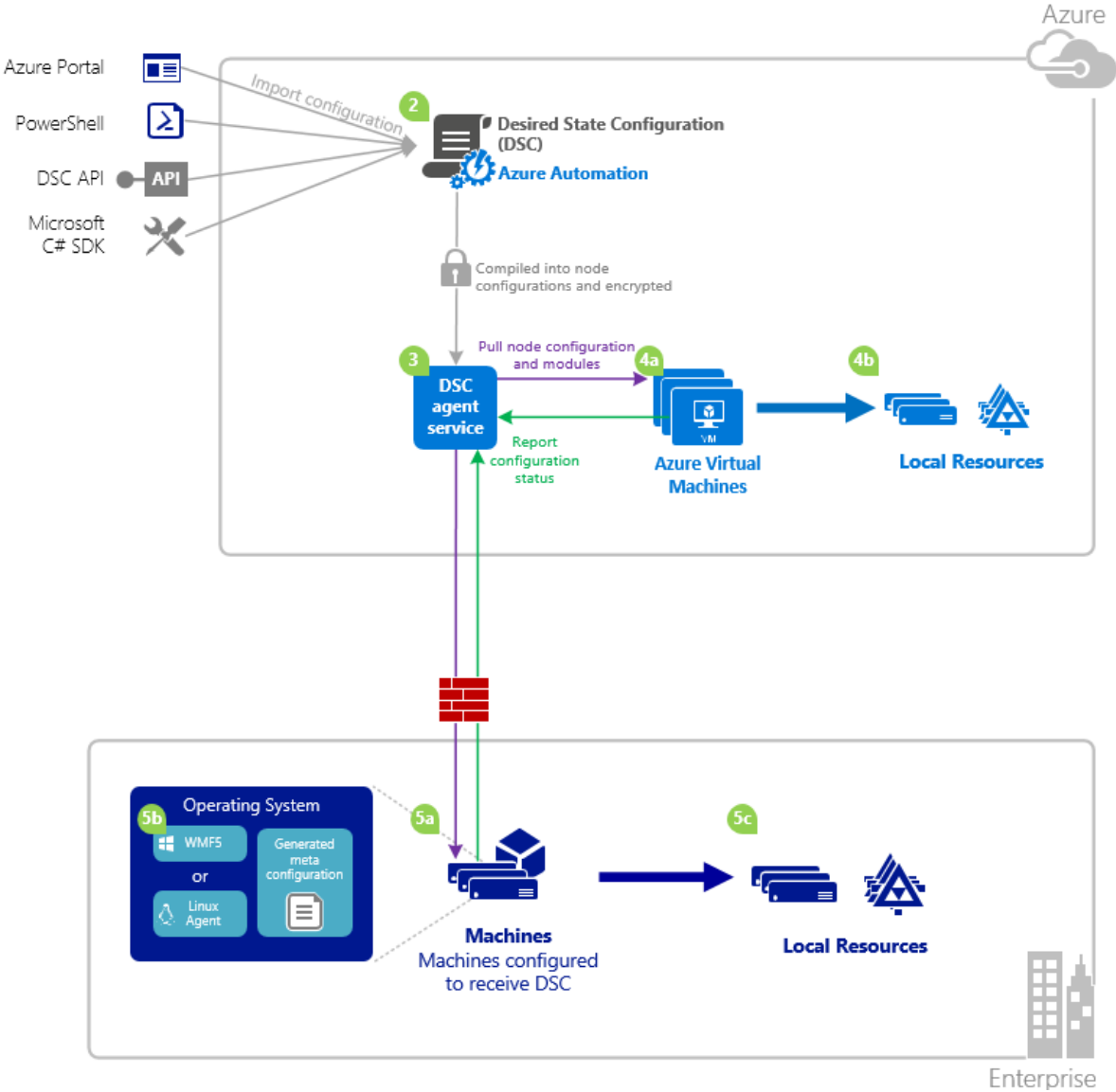


# Demo



# Azure Automation DSC

# Azure Automation DSC



# Usage Options

- Azure virtual machines (Classic + ARM)
- Amazon Web Services (AWS) virtual machines
- Physical/virtual machines on-premises
  
- Report-only endpoint from on-premises
  
- Windows + Linux

# Azure Automation

The screenshot displays the Microsoft Azure portal interface for the 'DSCDemo' Automation account. The breadcrumb navigation at the top shows 'Microsoft Azure > DSCDemo > DSC Configurations'. The account name 'DSCDemo' is displayed with a lightning bolt icon and the label 'Automation account'. Below this, there are four action buttons: 'Delete', 'Move', 'Feedback', and 'Refresh'. The 'Essentials' section provides key details: Resource group 'TechEd', Location 'West Europe', Subscription name 'Windows Azure MSDN - Visual Studio UI...', Status 'Active', Last modified '5/15/2016, 6:25 PM', and Last modified by 'michael.grafnetter@outlook.com'. An 'All settings' link is available. The dashboard features several tiles: 'Runbooks' (1), 'Jobs' (0), 'Hybrid Worker Groups' (0), 'ASSETS' (18), 'DSC Configurations' (0), 'DSC Node Configurat...' (0), and 'DSC Nodes' (0). The 'DSC Configurations' tile is highlighted with a blue border.

Microsoft Azure > DSCDemo > DSC Configurations

DSCDemo  
Automation account

Delete Move Feedback Refresh

Essentials

Resource group: TechEd  
Location: West Europe  
Subscription name: Windows Azure MSDN - Visual Studio UI...  
Status: Active  
Last modified: 5/15/2016, 6:25 PM  
Last modified by: michael.grafnetter@outlook.com

All settings

Runbooks: 1  
Jobs: 0  
Hybrid Worker Groups: 0  
ASSETS: 18  
DSC Configurations: 0  
DSC Node Configurat...: 0  
DSC Nodes: 0



# Compiling Configuration into MOF

The screenshot displays the 'WebFarm Configuration' interface. The top navigation bar includes 'WebFarm Configuration' and 'WebFarm Configuration source' tabs. Below the navigation bar are 'Compile', 'Export', and 'Delete' buttons. The main content area is divided into two panes. The left pane shows 'Essentials' with a list of properties: Resource group (TechEd), Location (westeurope), Subscription ID (f916a6a4-54d5-4a55-9ac8-cafce0eff2b7), Last published (5/15/2016, 8:40 PM), Account (DSCDemo), Subscription name (Windows Azure MSDN - Visual Studio Ulti...), Status (Published), and Configuration source (View configuration source). The right pane shows the configuration source code for 'Configuration WebFarm', which includes imports for PSDesiredStateConfiguration, xWebAdministration, and xTimeZone, and defines a 'Node WebNode' with a 'WindowsFeature IIS' and a 'File HomePage'.

**WebFarm Configuration**

Compile Export Delete

Essentials ^

Resource group  
TechEd

Location  
westeurope

Subscription ID  
f916a6a4-54d5-4a55-9ac8-cafce0eff2b7

Last published  
5/15/2016, 8:40 PM

Account  
DSCDemo

Subscription name  
Windows Azure MSDN - Visual Studio Ulti...

Status  
Published

Configuration source  
[View configuration source](#)

Deployments to Pull Server Add tiles (+)

Compilation jobs

STATUS	CREATED	LAST UPDATED
Queued	5/15/2016, 8:40 PM	5/15/2016, 8:40 PM

**WebFarm Configuration source**

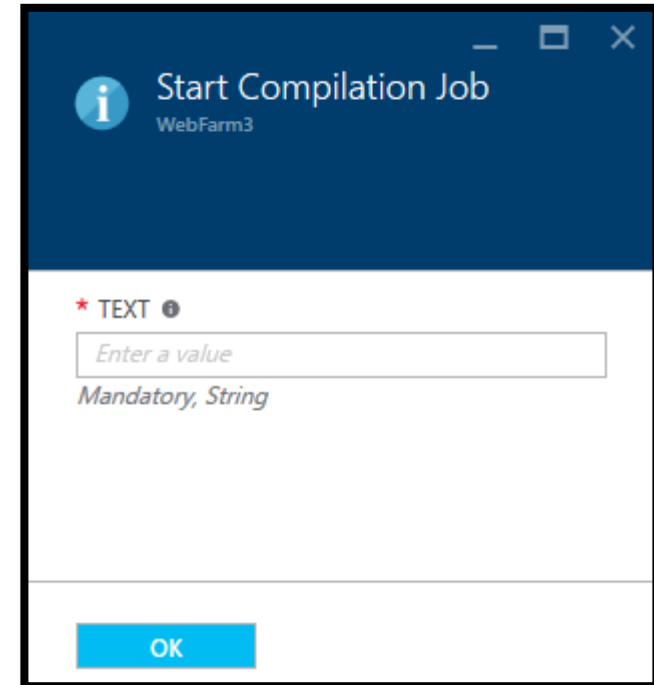
```
Configuration WebFarm
{
    Import-DscResource -ModuleName PSDesiredStateConfiguration
    Import-DscResource -ModuleName xWebAdministration
    Import-DscResource -ModuleName xTimeZone

    Node WebNode
    {
        WindowsFeature IIS
        {
            Ensure = 'Present'
            Name = 'Web-Server'
        }

        File HomePage
        {
            DependsOn = '[WindowsFeature]IIS'
            Ensure = 'Present'
            Type = 'File'
            DestinationPath = 'C:\inetpub\wwwroot\index.html'
            Contents = '<h1>Hello World!</h1>'
        }
    }
}
```

# Configuration Parameters

```
Configuration WebFarm3
{
    param(
        [Parameter(Mandatory = $true)]
        [ValidateNotNullOrEmpty()]
        [string] $Text
    )
}
```



Start Compilation Job  
WebFarm3

\* TEXT ⓘ

Mandatory, String

OK

# Compiling Configuration into MOF

```
Configuration WebFarm
{
  Node WebNode
  {
    WindowsFeature IIS
    {
      Ensure = 'Present'
      Name = 'Web-Server'
    }

    File HomePage
    {
      DependsOn = '[WindowsFeature]IIS'
      Ensure = 'Present'
      Type = 'File'
    }
  }
}
```

Available on Pull Server ⊕ Add Tiles

Node Configurations

**1**

NAME	LAST UPDATED
WebFarm.WebNode	5/15/2016, 8:42 PM

# Uploading Custom Modules

The screenshot displays the Azure PowerShell Modules interface. The main window, titled 'Modules', shows a list of installed modules. The 'Add Module' dialog box is open on the right, featuring an information message, an upload file option, and an 'OK' button.

Module Name	Last Updated	Status
AzureRM.Compute	4/26/2016, 2:48 AM	Available
AzureRM.Profile	4/26/2016, 2:48 AM	Available
AzureRM.Resources	4/26/2016, 2:49 AM	Available
AzureRM.Sql	4/26/2016, 2:50 AM	Available
AzureRM.Storage	4/26/2016, 2:50 AM	Available
Microsoft.PowerShell.Core	4/26/2016, 2:45 AM	Available
Microsoft.PowerShell.Diagnost...	4/26/2016, 2:45 AM	Available
Microsoft.PowerShell.Manage...	4/26/2016, 2:46 AM	Available
Microsoft.PowerShell.Security	4/26/2016, 2:46 AM	Available
Microsoft.PowerShell.Utility	4/26/2016, 2:47 AM	Available
Microsoft.WSMan.Management	4/26/2016, 2:47 AM	Available
Orchestrator.AssetManageme...	4/26/2016, 2:51 AM	Available
xTimeZone	5/15/2016, 9:20 AM	Available

**Add Module**

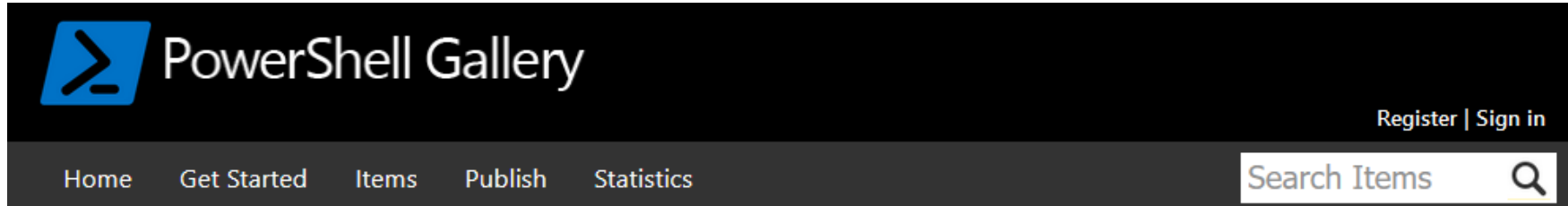
Importing a module may take several minutes.


\* Upload File (.zip format, 100 MB max size)

Select a file

OK


# Importing Modules from Gallery



 PowerShell Gallery Register | Sign in

Home Get Started Items Publish Statistics

---



## xTimeZone 1.3.0.0

This DSC Resources can easily set the System Time Zone.


**Inspect**

```
PS> Save-Module -Name xTimeZone -Path <path>
```

**Install**


```
PS> Install-Module -Name xTimeZone
```

**Deploy**

 Deploy to Azure Automation

See [Get Started](#) for more details.

**Owners**

 PowerShellTeam

**3,311**  
Downloads

**2,672**  
Downloads of 1.3.0.0

**2015-09-11**  
Last published

[Project Site](#)  
[License](#)  
[Contact Owners](#)  
[Report Abuse](#)  
[How to Download](#)  
[Module Statistics](#)

# Adding VM Nodes

The image shows two overlapping dialog boxes from the Azure portal. The 'Add Azure VMs' dialog is on the left, and the 'Select VMs' dialog is on the right. The 'Add Azure VMs' dialog has a sidebar with 'Virtual Machines' selected. The 'Select VMs' dialog displays a table of available VMs and a 'Create' button.

**Add Azure VMs**

- Virtual Machines  
Select virtual machines to onboard
- Registration  
Configure registration data

**Select VMs**

Azure Linux VMs are not supported for in-portal onboarding, though they may appear in the list below. Azure Classic VMs will not appear in the list below but can be onboarded using the Azure Classic VMs experience, via All settings -> Extensions -> Add -> Azure Automation DSC.

NAME	TYPE	LOCATION
✓ MyCoolV2VMxyz	Microsoft.Compute/virtualMachines	centralus
✓ WinServer2k12R2	Microsoft.Compute/virtualMachines	centralus

**Create** **OK**

# Adding VM Nodes

Add Azure VMs	Registration
Virtual Machines Select virtual machines to onboard >	* Registration key <input type="text" value="Primary key"/> <input type="text" value="Secondary key"/>
Registration Configure registration data >	Node Configuration Name ⓘ <input type="text" value="Ex: Myconfig.webserver"/>  Refresh Frequency ⓘ <input type="text" value="30"/>  Configuration Mode Frequency ⓘ <input type="text" value="15"/>  Configuration Mode ⓘ <input type="text" value="ApplyAndMonitor"/> ▼  Allow Module Override ⓘ <input type="checkbox"/>  Reboot Node if Needed ⓘ
<input type="button" value="Create"/>	<input type="button" value="OK"/>

# Azure VM Extension

The screenshot displays three panels from the Azure portal interface for a virtual machine named 'MYCoolVMxyz2'.

- Settings Panel:** Shows a search bar for settings and a list of configuration options for the virtual machine (CLASSIC). The 'Extensions' option is highlighted.
- Extensions Panel:** Features an 'Add' button and a table of installed extensions.
- DSC Panel:** Shows the details for the DSC extension, including its publisher, version, status, and a success message.

PUBLISHER	NAME	VERSION	STATUS
Microsoft.Compute	BGInfo	1.2.2	Ready
Microsoft.Powershell	DSC	1.10.1.0	Success

**DSC Configuration Details:**

- PUBLISHER:** Microsoft.Powershell
- VERSION:** 1.10.1.0
- STATUS:** Success
- MESSAGE:** DSC configuration was applied successfully.

[View detailed status](#)



# Checking the Compliance



The screenshot shows a window titled "DSC Nodes" with a dark blue header. Below the header are four action buttons: "Add Azure VM", "Add on-prem...", "Learn more", and "Refresh". Below these buttons is a table with four columns: "NAME", "STATUS", "NODE CONFIGURATION", and "LAST SEEN". The table contains one row for the node "DSCTESTVM", which has a status of "Compliant" (indicated by a green checkmark), a node configuration of "WebFarm.WebNode", and a last seen time of "5/15/2016, 9:24 PM".

NAME	STATUS	NODE CONFIGURATION	LAST SEEN
DSCTESTVM	✓ Compliant	WebFarm.WebNode	5/15/2016, 9:24 PM



# Demo



# **Configuring PowerShell Desired State Configuration Pull Server**

Michael Grafnetter

Mainstream Technologies