Creating a DSC Configuration

Jeff Hicks
Author/Teacher

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

Authoring

Windows 10 desktop

Locally installed DSC resources

DSC-aware scripting editor

- PowerShell ISE
- Visual Studio Code

Configuration

- Configuration is a keyword like Function
- It is a command type

```
Configuration BasicServer {

  # code
}
```

- You can name your configuration anything you'd like
- Insert your code in the scriptblock

```
Configuration BasicServer {

Import-DscResource
PSDesiredStateConfiguration

    # code

}
```

- Import the PSDesiredStateConfiguration module
- Import required DSC resource modules
- You should specify module version (see my course demos)

```
Configuration BasicServer {

Import-DscResource
PSDesiredStateConfiguration

 Node SRV1 {

 }


}
```

- Define a Node for each computer
- You will learn how to pass this as a parameter value
- Define all configurations for the node within the scriptblock

```
Configuration BasicServer {

Import-DscResource
PSDesiredStateConfiguration

 Node SRV1 {
     File <name> { }
 }


}
```

- Insert a configuration using a DSC Resource
- Name for each resource must be unique
- Enter resource settings in the script block

```
PS C:\> Get-DSCResource File -Syntax
File [String] #ResourceName
{
    DestinationPath = [string]
    [Attributes = [string[]]{ Archive | Hidden | ReadOnly | System }]
    [Checksum = [string]{ CreatedDate | ModifiedDate | SHA-1 | SHA-256 | SHA-512 }]
    [Contents = [string]]
    [Credential = [PSCredential]]
    [DependsOn = [string[]]]
    [Ensure = [string]{ Absent | Present }]
    [Force = [bool]]
    [MatchSource = [bool]]
    [PsDscRunAsCredential = [PSCredential]]
    [Recurse = [bool]]
    [SourcePath = [string]]
    [Type = [string]{ Directory | File }]
}
```

```
Configuration BasicServer {

Import-DscResource PSDesiredStateConfiguration

  Node SRV1 {
    File Demo {
      DestinationPath = "C:\DSCDemo"
      Ensure          = "Present"
      Type            = "Directory"
    }
  }
}
```

Save as a .ps1 script file

Using DSC Resources

DSC Resource modules must be installed on target nodes

- Deploy using your procedures

Import resource modules in your configuration

- Include module version

Pay attention to required settings

Configurations run unattended under SYSTEM

- We'll cover credentials later in the course

DependsOn

```
File Demo {
    DestinationPath = "C:\DSCDemo"
    Ensure          = "Present"
    Type            = "Directory"
}

File Readme {
    DestinationPath = "C:\DSCDemo\readme.txt"
    Ensure          = "Present"
    Type            = "File"
    Contents        = "Created for DSC demonstrations"
    Force           = $True
    DependsOn       = "[File]Demo"
}
```

```
PS C:\DSCConfigs>  .  .\basicServer.ps1
```

Compiling the Configuration

Dot source the configuration file

```
PS C:\DSCConfigs> . .\basicServer.ps1
PS C:\DSCConfigs> Get-Command basicserver

CommandType     Name                Version    Source
-----------     ----                -------    ------
Configuration   BasicServer
```

Compiling the Configuration

Dot source the configuration file

Configuration is a new command type

Configuration Help

```
PS C:\> help basicserver

NAME
    BasicServer

SYNTAX
    BasicServer [[-InstanceName] <string>] [[-DependsOn] <string[]>] [[-PsDscRunAsCredential] <pscredential>]
    [[-OutputPath] <string>] [[-ConfigurationData] <hashtable>]  [<CommonParameters>]


ALIASES
    None


REMARKS
    None
```

```
PS C:\DSCConfigs> BasicServer -OutputPath .
```

Compiling the Configuration

Run the configuration

```
PS C:\DSCConfigs> Get-ChildItem BasicServer

    Directory: C:\DSCConfigs\BasicServer


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         10/28/2021   2:37 PM           3404 SRV1.mof
```

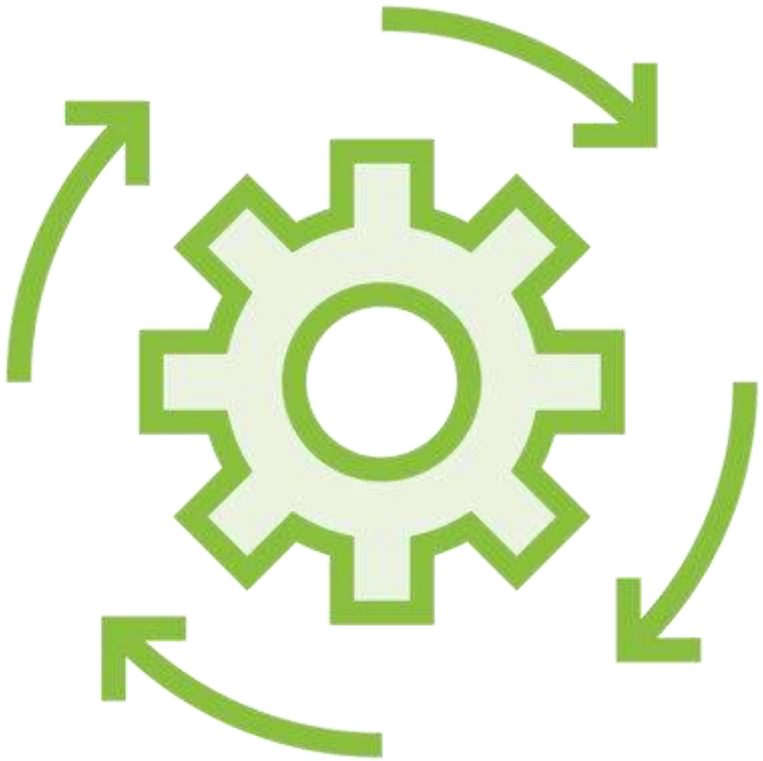
Compiling the Configuration

Run the configuration

If you don't specify a path, a directory will be created using the configuration name

The MOF is what will eventually be deployed. Don't modify.

The Local Configuration Manager (LCM)



Software element that manages DSC locally

Applies and monitors configurations

Can check for new configurations (pull)

Get-DSCLocalConfigurationManager

```
PS C:\DSCConfigs> Get-DscLocalConfigurationManager -CimSession srv1


ActionAfterReboot                : ContinueConfiguration
AgentId                          : 2BEB4D2F-15A4-11EC-A281-00155D0A7A7D
AllowModuleOverWrite             : True
CertificateID                    :
ConfigurationDownloadManagers    : {}
ConfigurationID                  :
ConfigurationMode                : ApplyOnly
ConfigurationModeFrequencyMins   : 15
Credential                       :
DebugMode                        : {NONE}
DownloadManagerCustomData        :
DownloadManagerName              :
LCMCompatibleVersions            : {1.0, 2.0}
LCMState                         : Idle
LCMStateDetail                   :
LCMVersion                       : 2.0
StatusRetentionTimeInDays        : 10
SignatureValidationPolicy        : NONE
SignatureValidations             : {}
MaximumDownloadSizeMB            : 500
PartialConfigurations            :
RebootNodeIfNeeded               : True
RefreshFrequencyMins             : 30
RefreshMode                      : PUSH
ReportManagers                   : {}
ResourceModuleManagers           : {}
PSComputerName                   : srv1
PSComputerName                   : srv1
```

LCM Settings

ActionAfterReboot

Specifies what happens after a reboot during a configuration.

- **ContinueConfiguration**
- **StopConfiguration**

Set RebootNodeIfNeeded to $True

LCM Settings

ConfigurationMode

How will the configuration be managed?

- **ApplyOnly**
- **ApplyAndMonitor**
- **ApplyAndAutoCorrect (default)**

LCM Settings

RefreshMode

How does the LCM get configurations?

- **Push (default)**
- **Pull**
- **Disabled**

```
LocalConfigurationManager {
   RebootNodeIfNeeded    = $True
   ConfigurationMode     = "ApplyAndAutoCorrect"
   ActionAfterReboot     = "ContinueConfiguration"
   RefreshMode           = "Push"
}
```

Configure the LCM

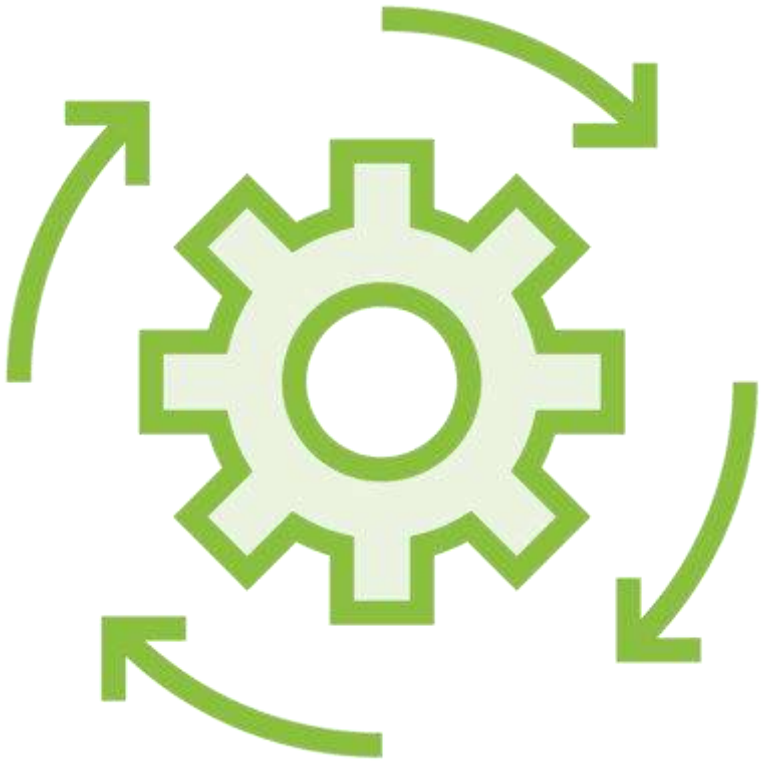
Add a resource to a node configuration

```
[DscLocalConfigurationManager()]
Configuration LCMConfig {
    Node SRV1 {
        Settings {
            RebootNodeIfNeeded   = $True
            ConfigurationMode    = "ApplyAndAutoCorrect"
            ActionAfterReboot    = "ContinueConfiguration"
            RefreshMode          = "Push"
        }
    }
}
```

Configure the LCM

Create a separate configuration

Compiling the LCM Configuration



Run the configuration

This will create a meta.mof file

- SRV1.meta.mof

We'll cover deployment in the next module

Demo

DSC Configurations