# Windows PowerShell Desired State Configuration

## Introduction to Desired State Configuration

**Jeff Hicks**
Author/Teacher

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

# Welcome

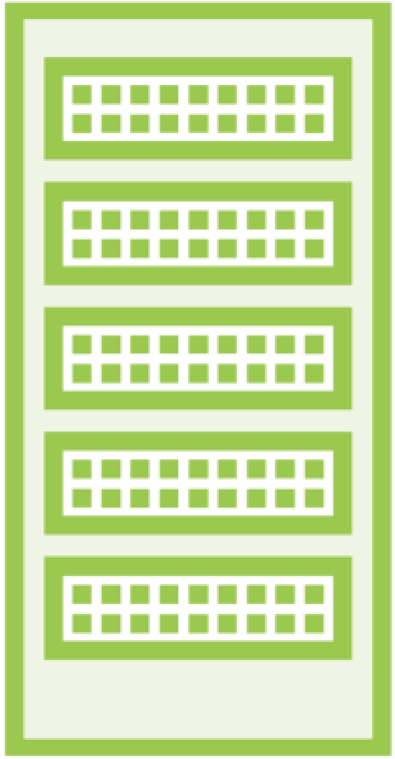
Welcome to a very special PowerShell course

Desired State Configuration (DSC) is a PowerShell-enabled framework

Solves a specific use-case

Configuration deployment and management

# The Need for DSC

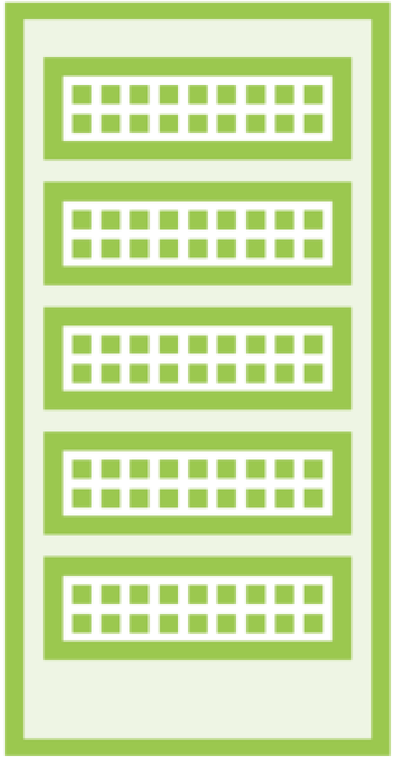


**You setup a new Windows server**

**You follow established organization-approved configuration procedures**

- Configuring services
- Setting Event Log limits
- Installing Windows Features
- Configuring the file system

# The Need for DSC



**The server configuration is ideal**

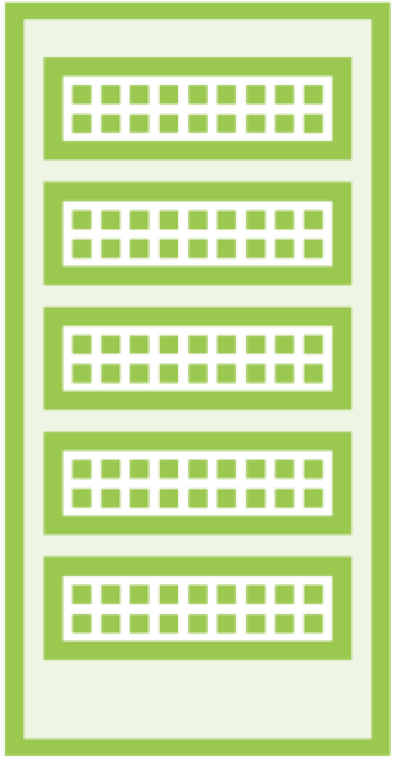
- For a day or so

**Server configurations *will* drift**

- Someone making an undocumented change
- Maybe an update changes something
- *Something* will happen

**Now the server is no longer in its desired configuration state**

# The Need for DSC



Do you even know the server is no longer properly configured?

Do you know what has changed?

How do you remediate?

How do you handle this at scale?

# The Need for DSC

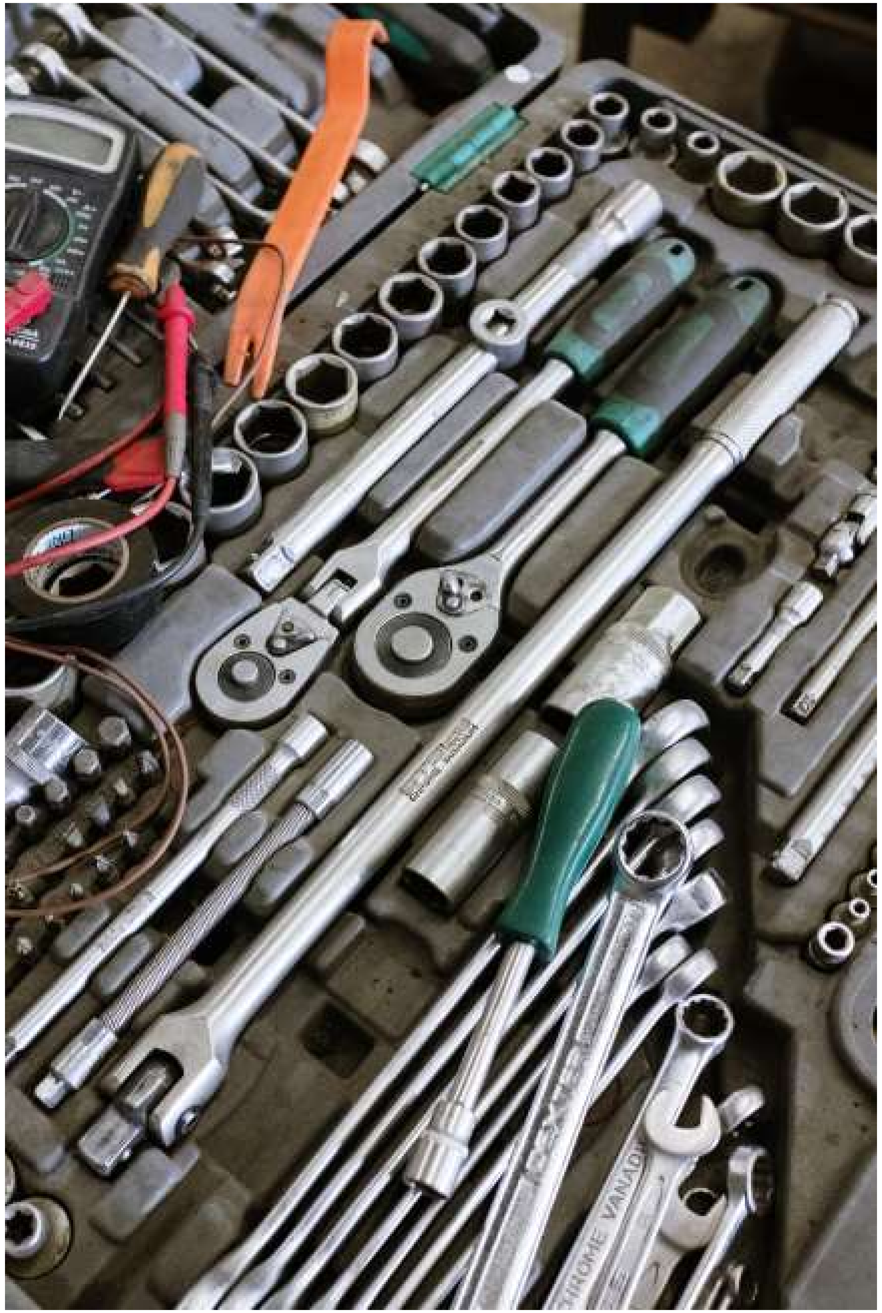
Desired State Configuration is a separate technology that leverages Windows PowerShell

Designed for on-premises Windows servers

(2012 R2 and later)

## There are other configuration management tools and techniques

- Chef
- Puppet
- Microsoft Configuration (Endpoint) Manager
- Custom scripting

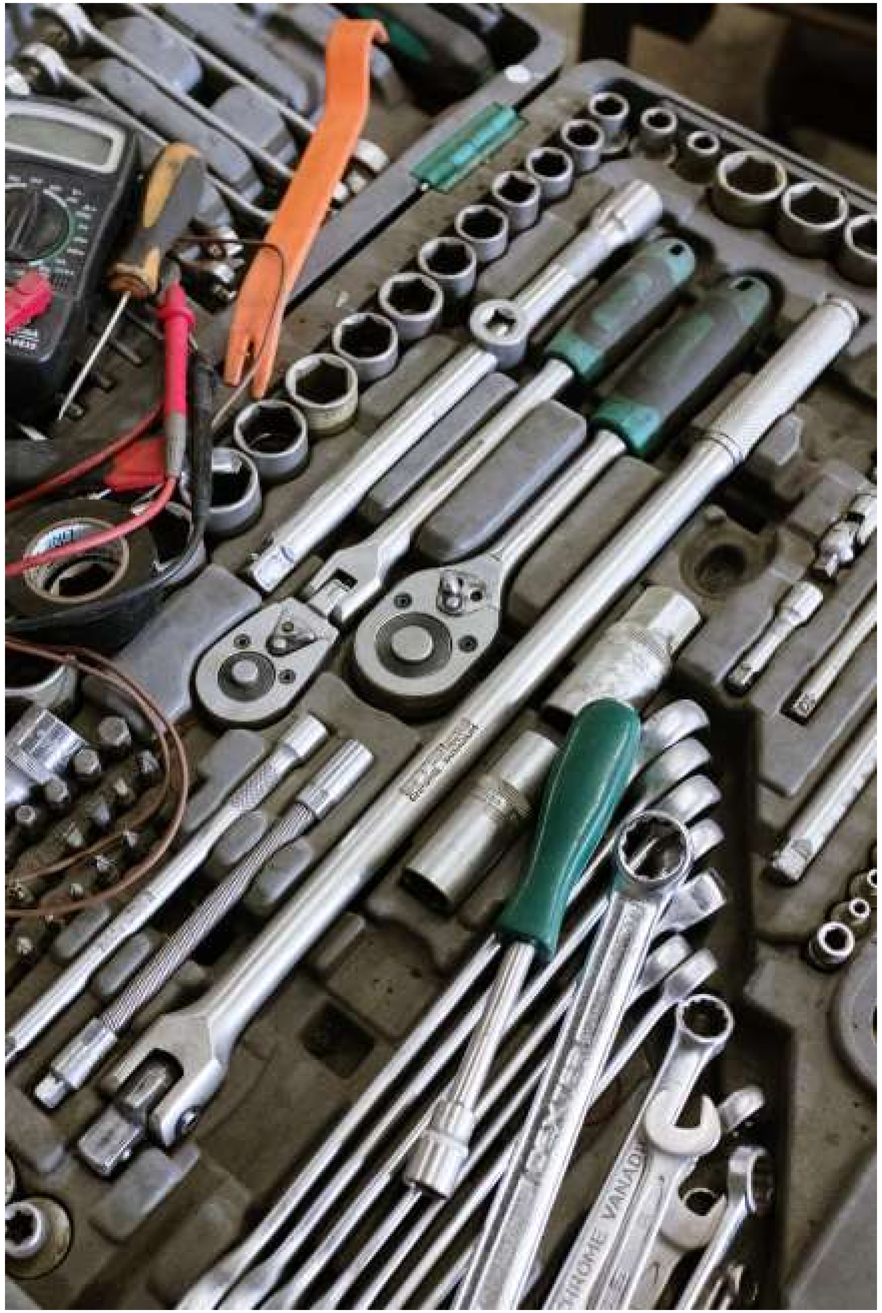


Apply defined configuration

Detect changes

Remediate

Ideally, automate



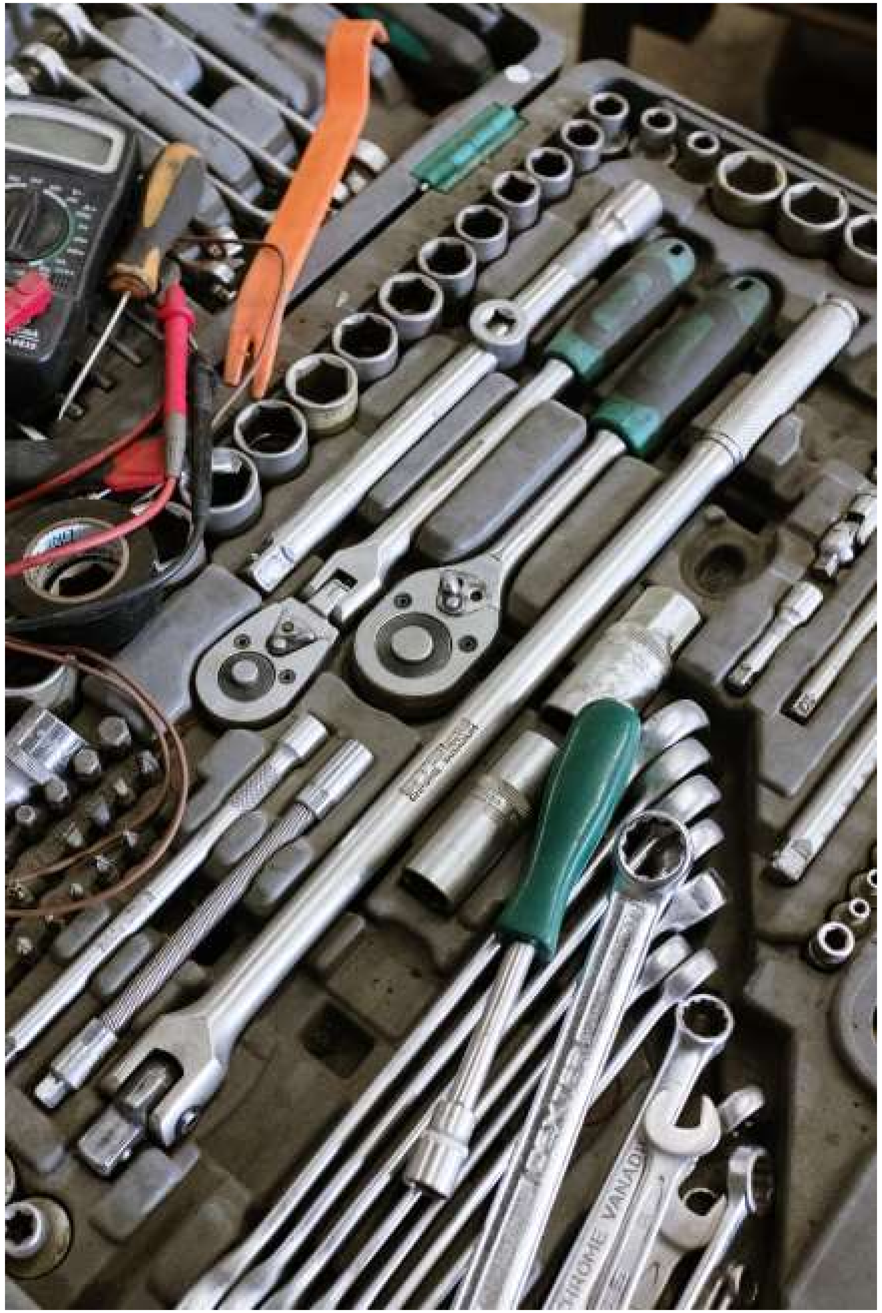
**Configuration management can be imperative**

- Run a script and do something

**Configuration management can be idempotent**

- Apply a defined configuration document

- Infrastructure from code



“Treat your servers like cattle not pets”

# DSC Past, Present, and Future

DSC introduced with Windows PowerShell 4.0

Improved with Windows PowerShell 5.0

Development subsides

# DSC Past, Present, and Future

DSC components installed by default

You can use DSC today to manage Windows servers using Windows PowerShell 5.1

DSC is not a product, it is a *framework*

# DSC Past, Present, and Future

DSC is being re-architected for PowerShell 7.x

It will require PowerShell 7 installations

Fundamental DSC concepts should remain the same

Techniques and technologies will most likely change

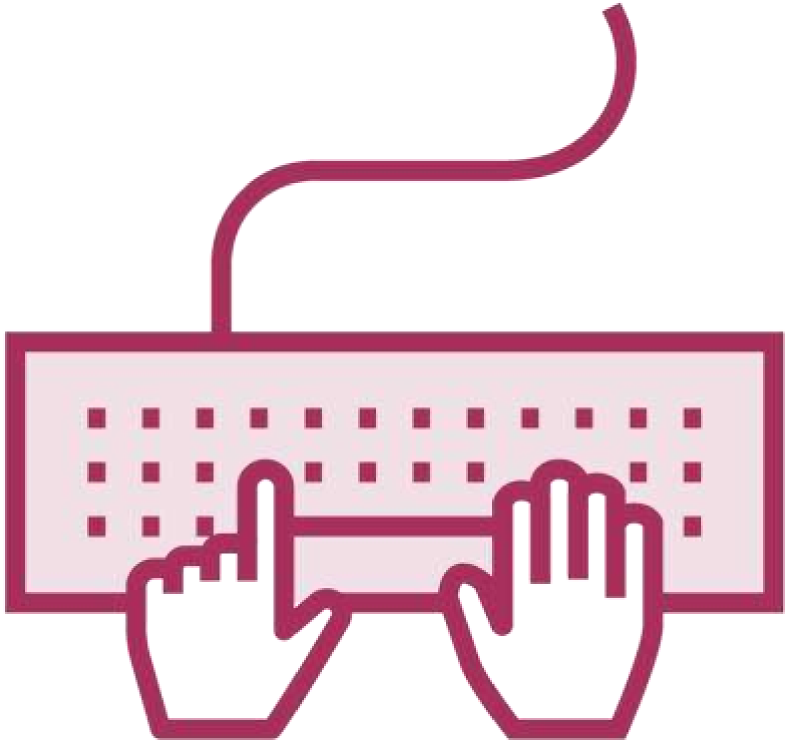
# Course Requirements

You are an experienced IT Pro

Familiar with Windows server setup and configuration

PowerShell scripting experience

# Setting Up a Lab Environment

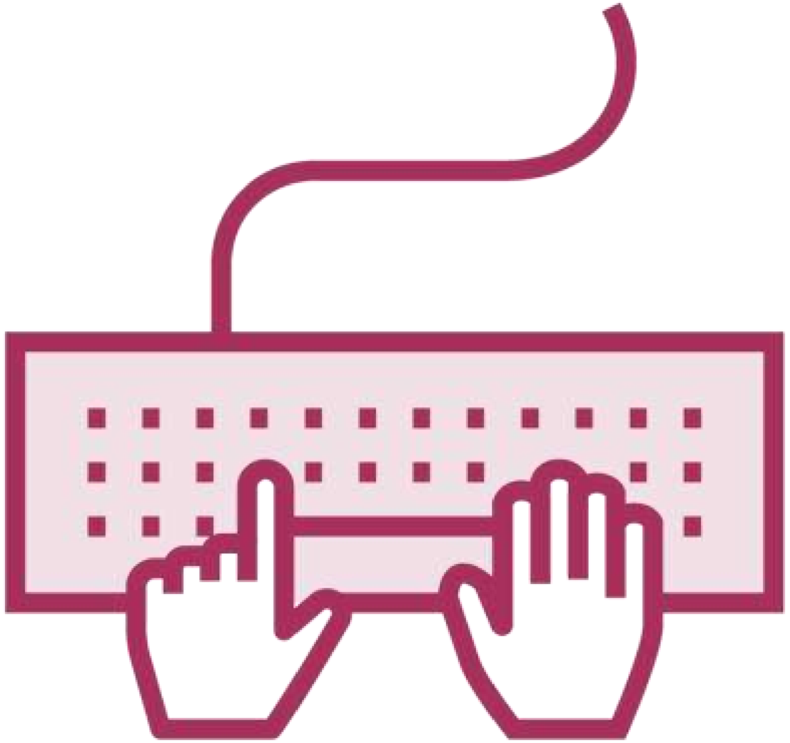


Expect a lot of demonstration

Download the course files

Setup a test lab to try yourself

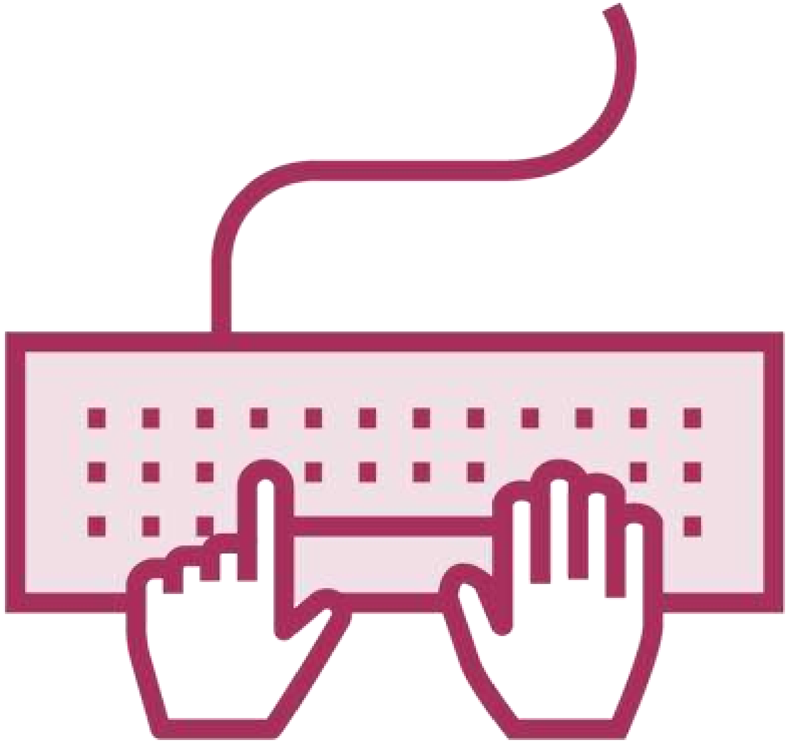
# Setting Up a Lab Environment



**You can use any test environment you want**

- Active Directory domain
- Windows 2016 domain member servers
- Windows 10 domain member client
- Windows PowerShell 5.1
- Windows PowerShell Remoting enabled
- Script execution allowed

# Setting Up a Lab Environment



**PSAutoLab**

- Windows PowerShell module
- Uses Windows 10 Hyper-V
- i7 processor (preferred)
- 16GB of memory
- 100GB of free disk space
- Use the PowerShellLab configuration
- http://bit.ly/PSLabSetup

# Additional Resources

**microsoft.com/powershell**

**powershell.org**

**PowerShell Scripting and Toolmaking**

- https://leanpub.com/powershell-scripting-toolmaking/

**The DSC Book**

- https://leanpub.com/the-dsc-book

Get yourself and your lab ready

Think about how you will want to use DSC