Introduction to Desired State Configuration (DSC)

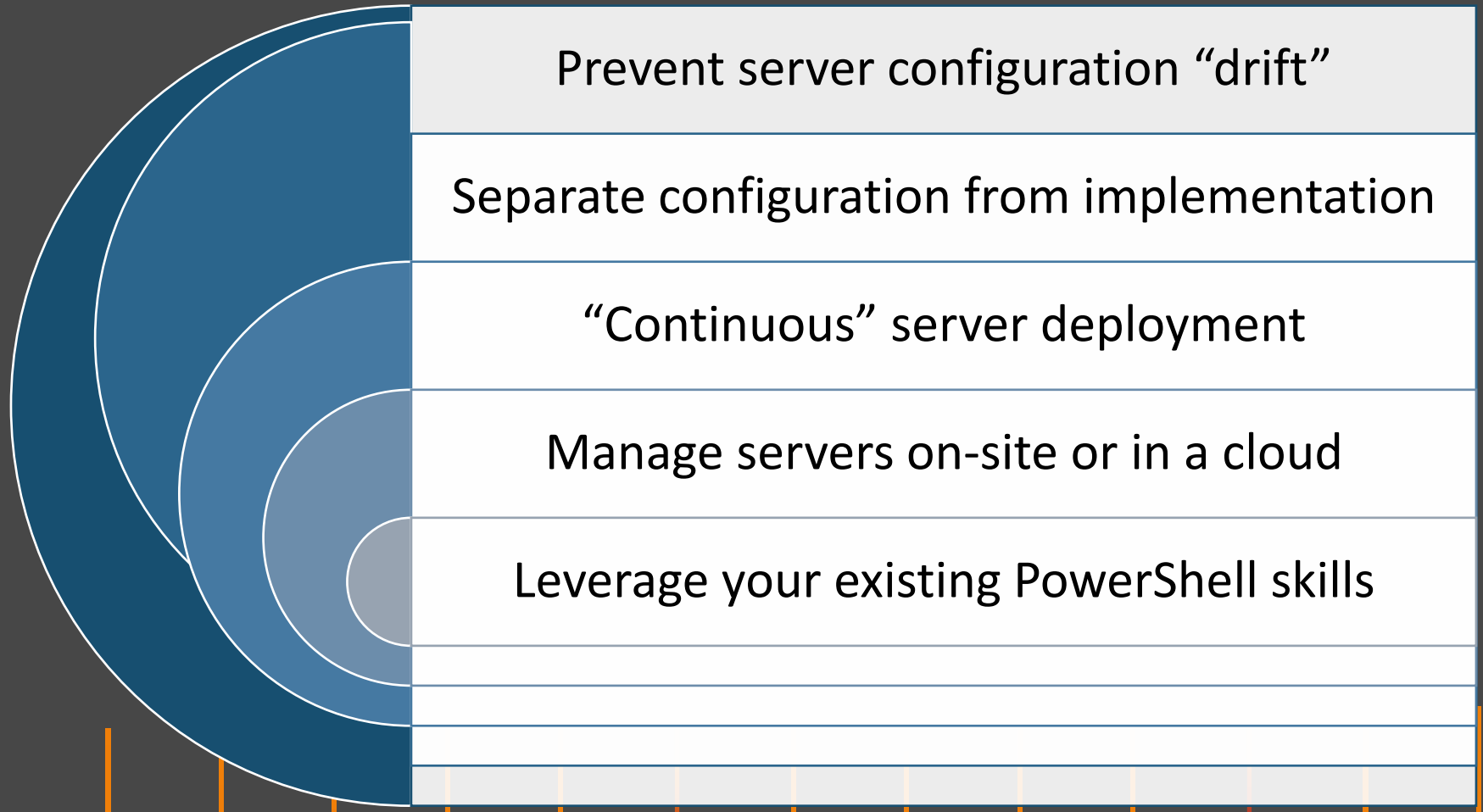
JEFFERY HICKS, MVP

JHICKS@JDHITSOLUTIONS.COM

What is DSC?

An extension to the PowerShell language	• Uses PowerShell syntax • Create configuration scripts
Create and manage server configuration files	• Use PowerShell language and cmdlets to create and deploy configurations
Ensures servers are always configured the way you need	• A local configuration manager does the heavy lifting (V2 only)

Why DSC?



V2 Requirements

- Windows PowerShell 5.1
- PowerShell Remoting
- Access to the PowerShell Gallery
- DSC-aware editor
- Optional: PKI for certificates

DSC Phases

Authoring Phase

- Can include imperative and declarative commands
- Create MOF definitions

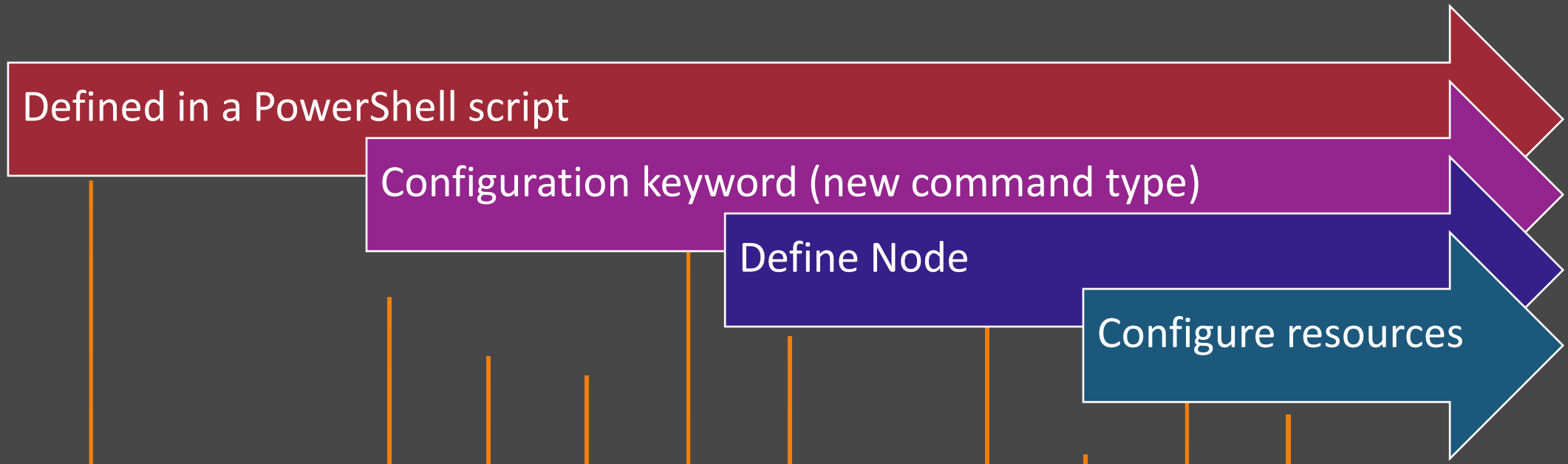
Staging Phase

- Declarative MOFs staged
- Configuration calculated per node

“Make It So” Phase

- Declarative configurations implemented through imperative providers

Creating a Configuration



DSC Resources

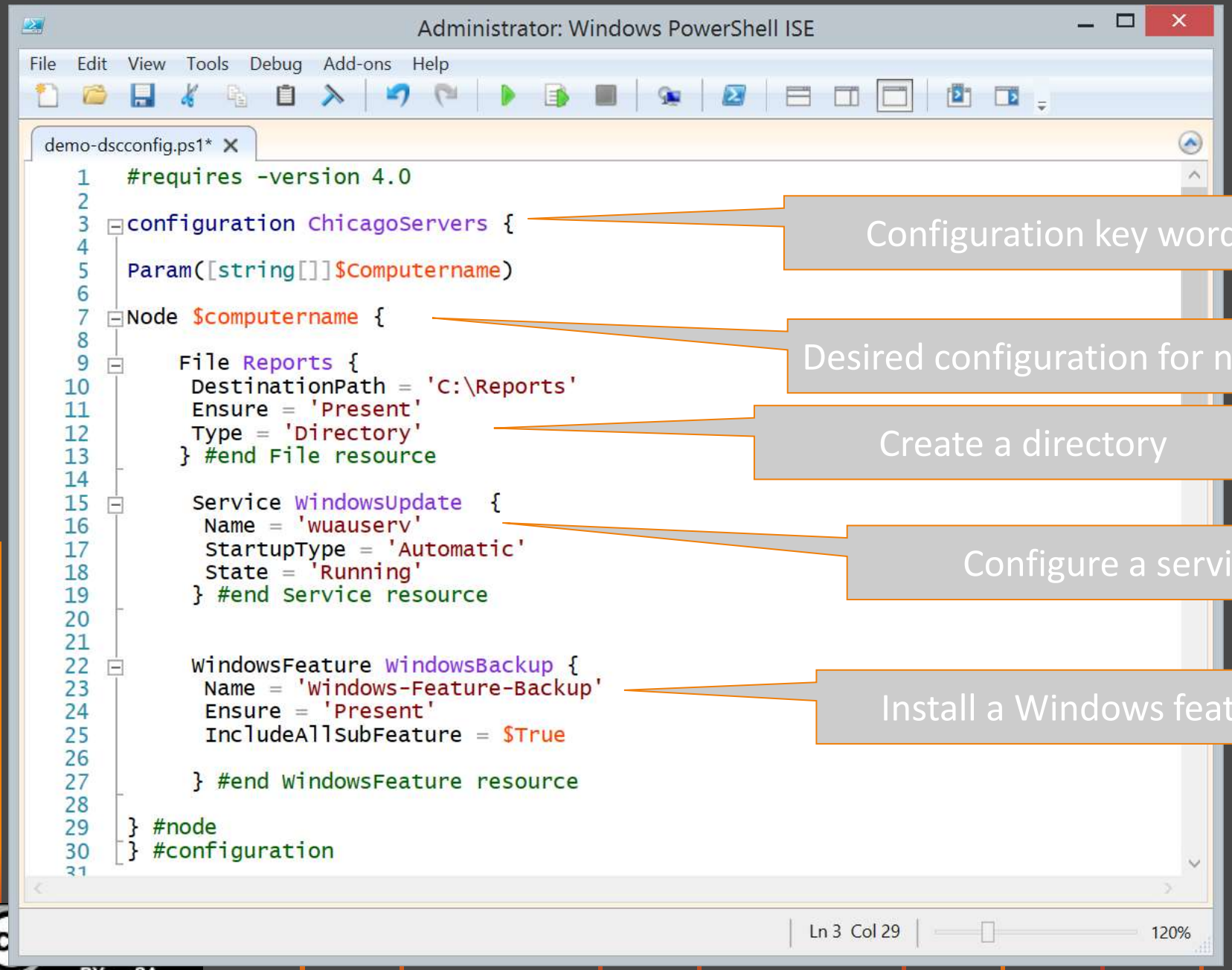
Managed element you define in your configuration

Core resources shipped “out of the box”

Additional “experimental” resources shipped from Microsoft

Community developed resources

You can write your own



Configuration key word

Desired configuration for nodes

Create a directory

Configure a service

Install a Windows feature

Define configuration and load into PowerShell

- `PS C:\Scripts> . .\ChicagoCoreConfig.ps1`

Defines a configuration called ‘ChicagoCore’

Invoke the configuration to create MOF

- `PS C:\Scripts> ChicagoCore`

Configuration has hard code node names

Start the configuration on the computer

- `PS C:\Scripts> Start-DscConfiguration -Path .\ChicagoCore`

Configuration pushed to every defined node

Configuration Management is a concept. DSC is a framework. You will need to provide tooling, practices and policies.