Automation using Powershell DSC

Building ~~Windows~~ Servers

Raj

JUST EAT

Agenda

- Quick intro
- Setup (if needed)
- Windows image on AWS
- LINUX image on AWS
- Deployment using Docker
- Windows 10 and Azure
- Pull based deployments

Get the code and instructions from:

https://github.com/rajwilkhu/prognet-2015.git

JUST EAT

Setup - Setting up the local machine

- Install WMF 4.0 if you do not have Windows 8.1
- Install Powershell tools for AWS
- Set up a local AWS profile (we'll do this at the start)
- Install Windows Azure Powershell (if you have an Azure account)
- Set up the local Azure environment (we'll do this at the start)

Why?

- Infrastructure as code
- Phoenix Servers
- Automate everything - production servers including build agents, dev vms
- Standardise a way to automate deployments - Lots of tools eg Chef, Puppet, Ansible, etc.

A bit of history

1999 Unix Services for Windows

https://technet.microsoft.com/en-us/library/bb496506.aspx

2002 Monad Manifesto

http://www.jsnover.com/Docs/MonadManifesto.pdf

2007 System Center 2007 Desired Configuration Manager

https://technet.microsoft.com/en-gb/library/bb680553.aspx

Built on System Center Configuration Manager - Desktop

2014 Desired State Configuration

https://technet.microsoft.com/en-gb/library/dn249912.aspx

Built on Powershell - Windows Management Framework 4.0

Windows 7/Windows Server 2008 R2 - Desktop and Server

What is DSC?

- Declarative configuration technology
- Final destination of the Monad Manifesto
- You don’t really need to learn Powershell
- Based on DMTF standards - CIM and WS-MAN
- DSC will be the primary administrative interface for Microsoft products and services
- Remoting is a prerequisite

What we get over traditional scripts

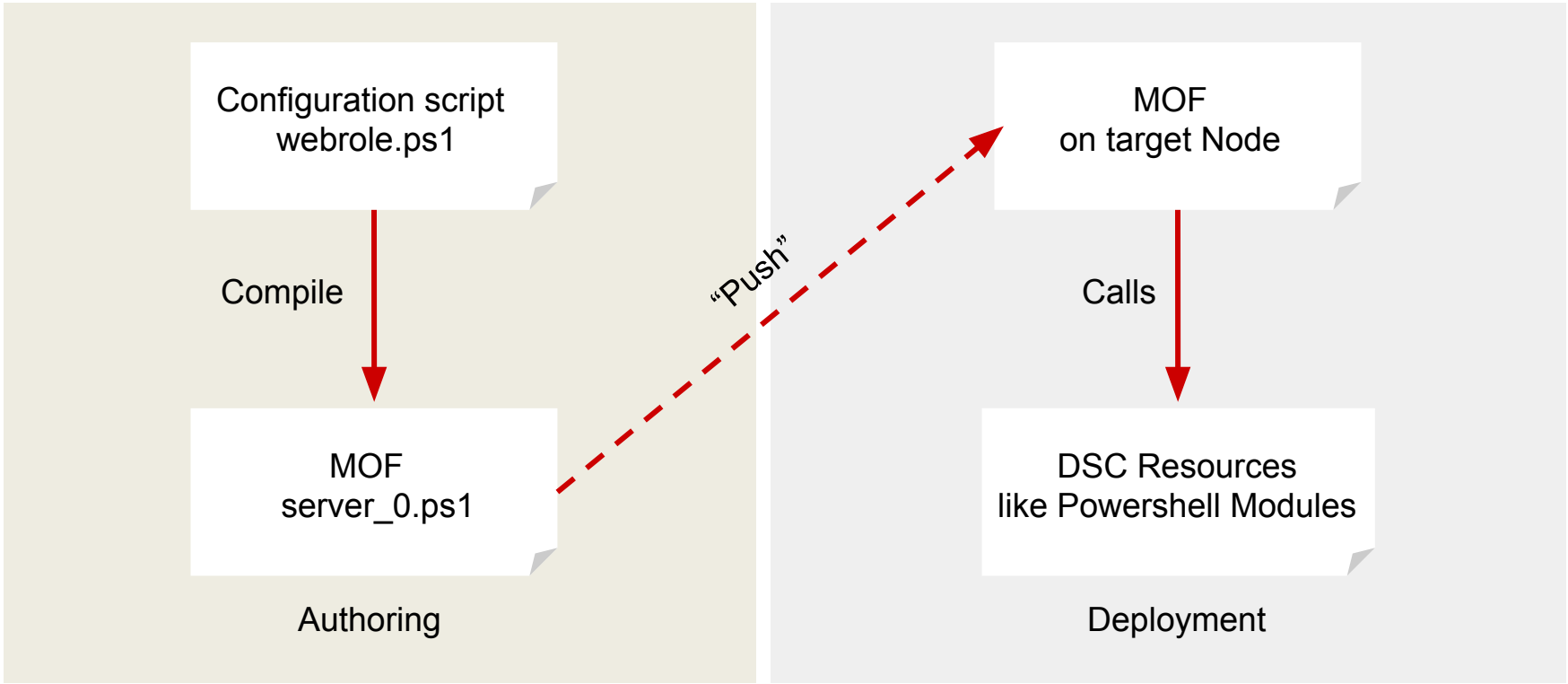
- Repeatable automation
- Logging and Error handling
- Reboot resiliency
- Dependency resolution
- Intent
- Technology specific

JUST EAT

Deployment modes

- Push
  Use *Start-DscConfiguration* to push MOF to target nodes
  Evaluate configuration immediately (and every 15 mins)
- Pull
  Push mode used to configure target nodes
  Target nodes check against pull server every 30 minutes
  Generated MOF files copied to pull server with checksum
  Zipped DSC Resources copied from the pull server

Overview



JUST EAT

Imperative vs Declarative

```powershell
Import-Module ServerManager

if (-not (Get-WindowsFeature "Application-Server").Installed) {
    try {
        Add-WindowsFeature Application-Server
    }
    catch {
        Write-Error $_
    }
}


if (-not (Get-Service "LOB Application Service").StartupType -eq "Automatic") {
    try {
        Set-Service "LOB Application Service" -StartupType Automatic
    }
    catch {
        Write-Error $_
    }
}

if (-not (Get-Service "LOB Application Service").Status -eq "Running") {
    try {
        Start-Service "LOB Application Service"
    }
    catch {
        Write-Error $_
    }
}
```

```powershell
Configuration LOBConfig
{
    Node LOBServer
    {
        WindowsFeature ApplicationServer
        {
            Ensure = "Present"
            Name = "Application-Server"
        }

        Service LOBService
        {
            Name = "LOB Application Service"
            StartupType = "Automatic"
            State = "Running"

        }
    }
}
```

JUST EAT

Local Configuration Manager

- Just an agent/service
  Part of WMF 4.0
  Has it's own meta-configuration
    pull mode, refresh interval, application mode, etc..

JUST EAT

Resources

- Script, composite or compiled modules
  You can write your own - it's easy

How do I get help?

Lots of Microsoft documentation and a free book:

https://www.penflip.com/powershellorg/the-dsc-book
Microsoft's DSC Resource Kit (quite a few resources)

JUST EAT

What I really like about DSC

- It’s just text!!

If you configure everything on your server using DSC, then all your configurations are just text files!

You can use any text editor!
Searchable, indexable and to some extent human-readable
Everything you need lives in Git!
Ideally all in one repo!

JUST EAT

When things go wrong

- -Verbose
- Windows Event log
- Write-Verbose

JUST EAT

Azure, VMM and Windows 10

- Inject MOFs into the VM on creation

Powershell Resource Gallery and OneGet
https://msconfiggallery.cloudapp.net/