

Microsoft Azure: Infrastructure as a Service (IaaS)

Conditions and Terms of Use

Microsoft Confidential

This training package is proprietary and confidential, and is intended only for uses described in the training materials. Content and software is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or software included in such packages is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URLs and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Copyright and Trademarks

© 2013 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see Use of Microsoft Copyrighted Content at

<http://www.microsoft.com/about/legal/permissions/>

Active Directory, DirectX, Internet Explorer, Microsoft, Outlook, SkyDrive, Windows, Windows Server, Windows Vista, Windows PowerShell, Xbox 360 and Zune are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

Module 5: Managing Virtual Machines (VMs) from Windows PowerShell

Introduction

Management mechanisms

Microsoft Azure Subscription

- Unique user account in Microsoft Azure.
- All resources are scoped under the Azure Subscription
- Uniquely identified by a subscription ID
- Security boundary for administration
- Unit of billing

settings

SUBSCRIPTIONS MANAGEMENT CERTIFICATES ADMINISTRATORS AFFINITY GROUPS USAGE

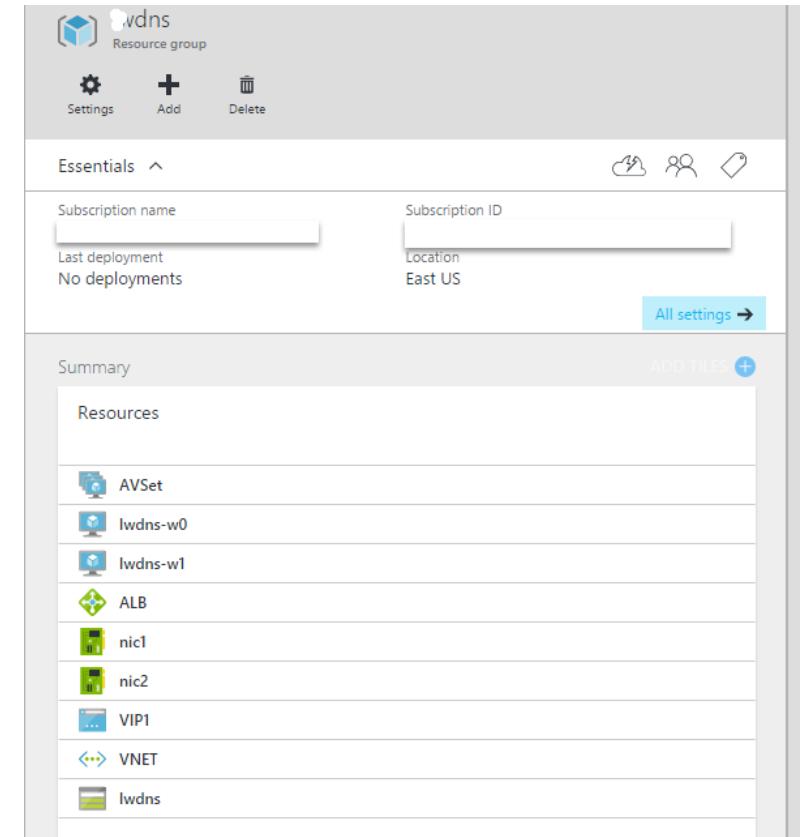
SUBSCRIPTI... SUBSCRIPTION ID ACCOUNT ADMINISTRATOR DIRECTORY

Azurepfe-fab... a6b20923-96bb-47eb-9333-806fb41... azurepfe@microsoft.com

Microsoft (microsoft.onmicrosoft.com)

Microsoft Azure Resource Group

- Resource Groups exist within a subscription
- A resource group can be secured via Role Based Access Control
- Resource groups are created via the Portal <https://portal.azure.com> or through RM PowerShell cmdlets
- Security boundary for administration of individual or group resources
- Unit of billing



Service Management API (SMAPI) - Classic

The Service Management API provides programmatic access to much of the functionality available through the Management Portal.

The Service Management API is a REST API. All API operations are performed over SSL and mutually authenticated using X.509 v3 certificates.

The subscription ID forms part of the URI for every call made to the Service Management API.

`https://management.core.windows.net/<subscriptionId>/...`

Azure Resource Manager API (ARMAPI) – V2

The Azure Resource Manager API provides programmatic access to much of the functionality available through the Management Portal
<https://portal.azure.com> .

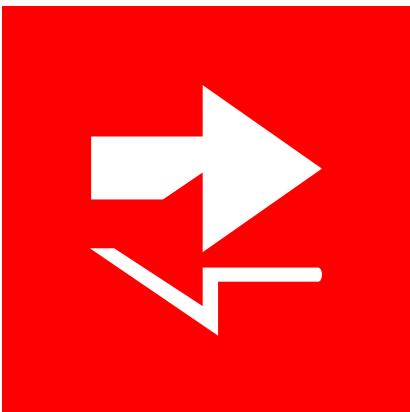
The ARMAPI is a REST API. All API operations are performed over SSL and mutually authenticated using Azure Active Directory

The subscription ID forms part of the URI for every call made to the ARMAPI
`https://management.azure.com/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/...`

Module 5: PowerShell

V2 (ARM)

Logging in to Azure



Login-AzureRmAccount

- With no parameters, will ask you to login, then will automatically select first Azure subscription it finds
- With *–SubscriptionName* or *–SubscriptionId* parameter, will ask you to log in, then select specific subscription
- Use *–Credential* if you already have credentials file



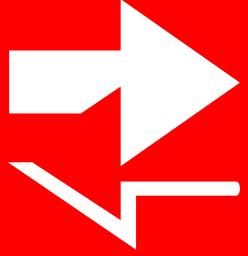
Before executing any other 'RM' commands, you need to first log in

How do I create an Azure Credential?

- An Azure Credential will allow you run a PowerShell script without a login prompt
- Does not use an .X509 Certificate (although you can)
- Uses an Azure Service Principal for an Azure AD Application
- You don't need to write a physical application, you just need to register an application name in Azure AD
- You must use an organizational account as the service principal identity, Microsoft accounts will not work (ie @Hotmail.com).

<https://azure.microsoft.com/en-in/documentation/articles/resource-group-authenticate-service-principal/>

Retrieving your Azure Subscription



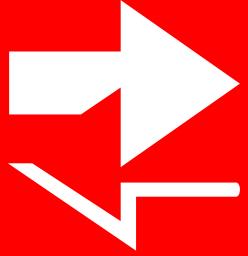
`Get-AzureRmSubscription | Select SubscriptionName, SubscriptionId`

- Returns all subscriptions related to previous `Login-AzureRMAccount` command
- Provides subscription name and ID
- With `-SubscriptionName` or `-SubscriptionId` parameter, will return information about this particular subscription



Generally used to gather list of subscriptions or a particular subscription

Selecting your Azure Subscription



Select-AzureRmSubscription

- With `-SubscriptionName` or `-SubscriptionId` parameter, will select this subscription into the running PowerShell session



Generally used to gather list of subscriptions or a particular subscription

Creating a new Resource Group

An empty resource group

```
New-AzureRmResourceGroup -Name $resourceGroupName -Location $location
```

A resource group from a custom template

```
New-AzureRmResourceGroup -Name $ResourceGroupName -Location $Location -TemplateFile  
‘.\ContosoHosting.json’ -DeploymentName $deploymentName -TemplateParameterFile  
‘.\ContosoHostingParms.json’
```

Creating a new V2 Storage account

An new locally redundant V2 storage account

```
New-AzureRmStorageAccount -ResourceGroupName $resourceGroupName -Name $storageAcctName -  
Type Standard_LRS -Location $location
```

Creating a Virtual Network

Configure a Subnet to put the VM in

```
$subnet = New-AzureRmVirtualNetworkSubnetConfig  
    -Name $subnetName  
    -AddressPrefix "10.0.64.0/24"
```

Create the Virtual Network to put the subnet in

```
$vnet = New-AzureRmVirtualNetwork -Name $vnetName  
    -ResourceGroupName $resourceGroupName  
    -Location $location  
    -AddressPrefix "10.0.0.0/16" -Subnet $subnet
```

Confirm Subnet configuration

```
$subnet = Get-AzureRmVirtualNetworkSubnetConfig  
    -Name $subnetName -VirtualNetwork $vnet
```

Create IL Public IP address and NIC

Create a new instance level public IP address

```
$pip = New-AzureRmPublicIpAddress  
    -ResourceGroupName $resourceGroupName  
    -Name $vipName -Location $location  
    -AllocationMethod Dynamic  
    -DomainNameLabel $domainLabel
```

Create a new Network Interface

```
$nic = New-AzureRmNetworkInterface  
    -ResourceGroupName $resourceGroupName  
    -Name "nic1" -Subnet $subnet  
    -Location $location -PublicIpAddress $pip  
    -PrivateIpAddress "10.0.64.4"
```

VM image retrieval

Image identification (example)

- Publisher - MicrosoftWindowsServer
- Offer - WindowsServer
- SKU – 2012-R2-Datacenter
- Version – 4.0.201506
- Location – westus

PowerShell

- \$publisher = **Get-AzureRmVMImagePublisher** –Location \$location
- \$offer = **Get-AzureRmVMImageOffer** –Location *\$location* –PublisherName \$publisher
- \$sku = **Get-AzureRmVMImageSku** –Location *\$location* –PublisherName \$publisher –Offer \$offer
- \$imageName = **Get-AzureRmVMImage** –Location *\$location* –Offer \$offer –PublisherName \$publisher –SKUs \$sku
- **Save-AzureRmVMImage**

General Output of Get-AzureRmVMImage...

Windows	SQL
Get-AzureRmVMImagePublisher <ul style="list-style-type: none">MicrosoftWindowsServer	Get-AzureRmVMImagePublisher <ul style="list-style-type: none">MicrosoftSQLServer
Get-AzureRmVMImageOffer <ul style="list-style-type: none">WindowsServer	Get-AzureRmVMImageOffer <ul style="list-style-type: none">SQL2008R2SP3-WS2008R2SP1SQL2012SP2-WS2012SQL2012SP2-WS2012R2SQL2014-WS2012R2SQL2014SP1-WS2012R2SQL2016CTP2-WS2012R2
Get-AzureRmVMImageSKU <ul style="list-style-type: none">2008-R2-SP12012-Datacenter2012-R2-Datacenter2016-Technical-Preview-3-with-ContainersWindows-Server-Technical-Preview	Get-AzureRmVMImageSKU (SQL2012SP2-WS2012R2) <ul style="list-style-type: none">EnterpriseEnterprise-Optimized-for-DWEnterprise-Optimized-for-OLTPStandardWeb

Configuring a VM prior to creation

Setup a new VM configuration

```
$vmConfig = New-AzureRmVMConfig -VMName $vmName  
    -VMSize $vmSize |
```

Set the Operating System Parameters

```
Set-AzureRmVMOperatingSystem -Windows  
    -ComputerName $vmName -Credential $cred  
    -ProvisionVMAgent -EnableAutoUpdate |
```

Set the VM Image Source location

```
Set-AzureRmVMSourceImage  
    -PublisherName $publisher -Offer $offer  
    -Skus $sku -Version $version |
```

Deploying a VM

Setup OS Disk caching parameters (optional)

```
Set-AzureRmVMOSDisk -Name $vmName  
    -VhdUri $vhURI -Caching ReadWrite  
    -CreateOption fromImage |
```

Add the network interface to the VM configuration

```
Add-AzureRmVMNetworkInterface -Id $nic.Id
```

Create the new VM

```
New-AzureRmVM -ResourceGroupName $resourceGroupName  
    -Location $location -VM $vmConfig -Name $vmName
```

Complete Command for VM configuration

```
$vmConfig = New-AzureRmVMConfig -VMName $vmName -VMSize "Standard_D1" | Set-AzureRmVMOperatingSystem -Windows -  
ComputerName $vmName -Credential $cred -ProvisionVMAgent -EnableAutoUpdate | Set-AzureRmVMSourceImage -PublisherName  
$publisher -Offer $offer -Skus $sku -Version "4.0.201506" | Set-AzureRmVMOSDisk -Name $diskName -VhdUri $osDiskUri -  
Caching ReadWrite -CreateOption fromImage | Add-AzureRmVMNetworkInterface -Id $nic.Id  
  
New-AzureRmVM -ResourceGroupName $resourceGroupName -Location $location -VM $vmConfig -Name $vmName
```

Create a Static Public IP address



New-AzureRmPublicIpAddress Options

```
$vip = New-AzureRmPublicIpAddress -ResourceGroupName $resourceGroupName -Name "VIP1" -Location $location -AllocationMethod Static -DomainNameLabel $domainName
```

- Name – the name to be applied to the VIP
- Allocation method – static or dynamic, depends on if you want a reserved IP address
- DomainNameLabel – provides a DNS name like 'contoso.eastus.cloudapp.azure.com'

Get information about IP address

```
Get-AzureRMPublicIPAddress –Name 'VIP1' –ResourceGroupName $resourceGroupName
```

- Public static IP addresses, 20 per subscription
- Public dynamic IP addresses, 60 per subscription
- Public front end IP per load balancer ~ 5
- Private front end IP per load balancer ~ 1

Create a Static Private IP



New-AzureRmVMNetworkInterface Options

```
$nic1 = New-AzureRmNetworkInterface -ResourceGroupName $resourceGroupName `  
-Name "nic1" -Subnet $subnet -Location $location -PrivateIpAddress '10.0.0.54' `  
-LoadBalancerInboundNatRule $alb.InboundNatRules[0] `  
-LoadBalancerBackendAddressPool $alb.BackendAddressPools[0]
```

- ARM VMs are associated with NICs
- NICs are connected to a subnet
- You do not need to specify 'Static' to have a static IP address, that is the default when you specify -PrivateIpAddress

Configuring Load Balancing (example)

```
# establish load balancer configuration using pre-recreated public IP address
$felpConfig = New-AzureRmLoadBalancerFrontendIpConfig -Name "FEIP" `

# establish a NAT rule to allow RDP access
$inboundNATRule1 = New-AzureRmLoadBalancerInboundNatRuleConfig -Name "RDP1" `

# establish the backend pool configuration
$beAddressPool = New-AzureRmLoadBalancerBackendAddressPoolConfig -Name "LBBE"

# establish a health probe
$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name "HealthProbe" `

# establish a load balancer rule for Http access
$lbrule = New-AzureRmLoadBalancerRuleConfig -Name "HTTP" `

# Configure the load balancer
$alb = New-AzureRmLoadBalancer -ResourceGroupName "SomeResourceGroup" `
```

User Defined Routes



New-AzureRmRouteTable Options

```
$myTable = New-AzureRmRouteTable –Name FrontEndSubnetRouteTable –Location 'West US' –  
ResourceGroupName $resourceGroupName
```

Add a Route to a Route Table

```
$myTable | Add-AzureRmRouteConfig –Name FirewallRoute –AddressPrefix 10.2.0.0/16 –  
NextHopType VirtualAppliance –NetHopIpAddress 10.1.1.10 | Set-AzureRmRouteTable
```

Apply to Subnet

```
Set-AzureRmVirtualNetworkSubnetConfig -VirtualNetwork $vnet -Name $subnetName -AddressPrefix  
$subnet.AddressPrefix -RouteTableId $myTable.Id | Set-AzureRmVirtualNetwork
```

View Applied Routes

```
Get-AzureRmRouteTable -ResourceGroupName $rgName -Name $routeTableName
```

Demo: V2 PowerShell Script

Module 5: PowerShell

Classic

What Can You Do with Azure PowerShell?



Automation

- Query, manage and configure VMs across multiple subscriptions, cloud services, and storage accounts.



Provision Fully Configured VMs

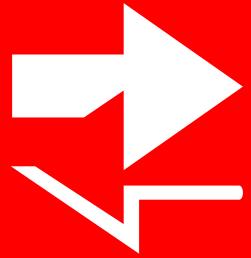
- Domain-joined
- Storage and networking configured



Virtual Networking

- Completely configure virtual networks from a script

Getting Subscription Information



Use `Add-AzureAccount`
to import subscription information into the PowerShell session window



Automatically pulls in all subscription information and configures one of
the subscriptions as the default subscription. You can change this by
calling `Get-AzureSubscription` and then `Select-AzureSubscription`
Allows non-interactive mode by using the `-Credentials` parameter

Subscription Management



Subscription Settings Persisted

C:\Users\user\AppData\Roaming\Microsoft Azure Powershell



Subscription Example

```
<Subscription name="somesub1">
  <SubscriptionId>13d83b03-6d06-4770-943c-3d46766c3a35</SubscriptionId>
  <Thumbprint>2AC8112B34CC840A30B9C2716AE840D5DC107510</Thumbprint>
  <ServiceEndpoint>https://management.core.windows.net/</ServiceEndpoint>
</Subscription>
```

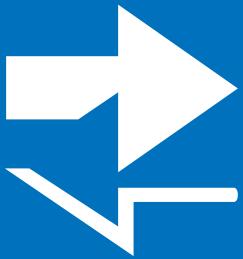
Manual Subscription Configuration



Associate Certificate and Subscription ID

```
$cert = Get-Item cert:\CurrentUser\My\CERTTHUMBPRINT  
Set-AzureSubscription 'mysub' -Certificate $cert -SubscriptionID $id
```

Switching Between Subscription Settings



Multiple Subscription Support

```
Get-AzureSubscription | foreach {  
    Select-AzureSubscription $_.SubscriptionName  
    # Perform Management Operation Against Each Subscription  
}
```

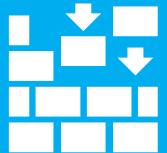
Setting the Current Storage Account



Returns Storage Account

Get-AzureStorageAccount | Select StorageAccountName

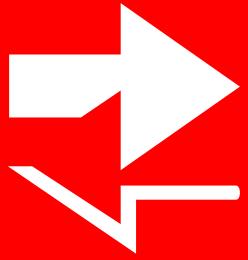
*returns only the storage account(s) in the currently selected subscription



Sets the Current Storage Account

Set-AzureSubscription 'somesub1' -CurrentStorageAccount 'mystorage'

Getting Subscription Information



Use `Get-AzurePublishSettingsFile` and `Import-AzurePublishSettingsFile` to import downloaded Publish Profile (.publishsettings)
<http://windows.azure.com/download/publishprofile.aspx>



Automatically configures:

- Subscription ID
- Certificate
- Service Endpoint
- Subscription Name

Classic

Information Needed to Create a VM



Image Name

```
Get-AzureVMImage | select ImageName
```

Disk Name

```
Get-AzureDisk | select DiskName
```

Get Image name ~ latest WS2012R2

```
$imageName = (Get-AzureVMImage | Where { $_.ImageFamily -eq "Windows Server 2012 R2 Datacenter" } | sort PublishedDate -Descending | Select-Object -First 1).ImageName
```



Data Center Location

```
Get-AzureLocation
```

Classic

Virtual Machine Management



Quick VM Provisioning Mode

Supports VM creation in a single cmdlet (New-AzureQuickVM)



Advanced Provisioning Configuration Mode

Provision with: Endpoints, Data Disks

Configure: Cache settings for OS/Data Disks and Subnet Names



Create Multiple Pre-Defined VMs in a Batch

New-AzureVM -VMs \$vm1, \$vm2, \$vm3

Classic

Simple VM Creation

- First VM in a *new* Cloud Service (-Location parameter used)

```
New-AzureQuickVM -Windows -ServiceName $svc -Name $vm1 -ImageName $wimg -Location  
$location -Password $pwd -AdminUserName $adminName
```

- New VM in an existing Cloud Service (-Location parameter not used)

```
New-AzureQuickVM -Windows -ServiceName $svc -Name $vm2 -ImageName $wimg -  
AdminUserName $adminName Password $pwd
```

- Creating a Linux VM in an existing Cloud Service

```
New-AzureQuickVM -Linux -ServiceName $svc -Name $vm3 -ImageName $limg -  
LinuxUser $lu -Password $pwd
```

Configuring VM at Provisioning

- Create a Configuration Object with New-AzureVMConfig
- Modify with Add-* cmdlets
- Add with New-AzureVM

```
New-AzureVMConfig -Name $vm1 -InstanceSize Medium -ImageName $img |  
Add-AzureProvisioningConfig -Windows -AdminUserName $adminName -Password $pwd  
|  
Add-AzureDataDisk -CreateNew -DiskLabel 'data' -DiskSizeInGB 10 -LUN 0 |  
Add-AzureEndpoint -Name 'web' -PublicPort 80 -LocalPort 80 -Protocol tcp |  
New-AzureVM -ServiceName $newSvc -Location $location
```

VM Batch Creation

- Create multiple configured VMs and pass them to New-AzureVM

```
$vm1 = New-AzureVMConfig -Name 'myvm1' -InstanceSize 'Small' -ImageName $img | Add-AzureProvisioningConfig -Windows -AdminUserName $adminName -Password $pwd

$vm2 = New-AzureVMConfig -Name 'myvm1' -InstanceSize 'Small' -ImageName $img | Add-AzureProvisioningConfig -Windows -AdminUserName $adminName -Password $pwd

$vm3 = New-AzureVMConfig -Name 'myvm1' -InstanceSize 'Small' -ImageName $img | Add-AzureProvisioningConfig -Windows -AdminUserName $adminName -Password $pwd

New-AzureVM -CreateService -ServiceName $cloudSvcName -VMs $vm1,$vm2,$vm3 -Location $dc
```

Classic

VM Batch Creation - Using an Array

- Create multiple configured VMs and pass them to New-AzureVM

```
$vmcount = 5
$vms = @()
for($i = 0; $i -lt 5; $i++)
{
    $vmn = 'myvm' + $i
    $vms += New-AzureVMConfig -Name $vmn -InstanceSize 'Small' -ImageName
$img |
    Add-AzureProvisioningConfig -Windows -AdminUserName $adminName -Password
$pwd |
    Add-AzureDataDisk -CreateNew -DiskLabel 'data' -DiskSizeInGB 10 -LUN 0 |
    Add-AzureDataDisk -CreateNew -DiskLabel 'logs' -DiskSizeInGB 10 -LUN 1
}

New-AzureVM -ServiceName $cloudSvcName -VMs $vms -Location $dc
```

Common Settings



Name

The name of the VM



AvailabilitySetName

The availability set (used for high availability)



InstanceSize

A0 – A11, D1 – 4, D11 – D14, G1 – G5 + 'S' Series machines

Classic

Windows Provisioning Options



Add-AzureProvisioningConfig Options

- Windows -AdminUserName \$adminName -Password \$pwd
- WindowsDomain -Password \$pwd
 - Domain \$dom, -JoinDomain \$fqdn, -DomainUser \$domUser
 - DomainPassword \$domPwd -MachineObjectOU \$ou
- DisableAutomaticUpdates
- NoRDPEndpoint, -TimeZone, Certificates

Classic

Setting a Static Internal IP



New-AzureVMConfig Options

```
New-AzureVMConfig -Name "myNewVM" -InstanceSize "Small" -ImageName $imageName | Add-AzureProvisioningConfig -Windows -AdminUsername $user -Password $pwd | Set-AzureSubnet -SubnetNames "AppSubnet" | Set-AzureStaticVNetIP -IPAddress "10.0.0.15" | New-AzureVM -ServiceName "vnetsvc"
```

- Cloud Service already exists (vnetsvc) in a virtual network
- Virtual network has a subdomain named 'AppSubnet'
- Subnet address range must be within the –IPAddress range
- If a machine in the subnet already has this address, you will receive an error when you run Set-AzureStaticVNetIP
- If all machines in the subnet are shut down and you restart this VM first, it will go grab the static IP address listed above

Classic

Setting an Instance Level Public IP Address for a VM



New-AzureVMConfig Options

```
New-AzureVMConfig -Name "WebAppVM" -InstanceSize Small -ImageName $imageName | Add-AzureProvisioningConfig -Windows -AdminUsername $username -Password $password | Set-PublicIP -PublicIPName "ftpip" | New-AzureVM -ServiceName "MyWebAppService" -ReservedIPName "MyWebSiteIP" -Location "East US"
```

Get information about VM

```
Get-AzureRole -ServiceName FTPInAzure -Slot Production -InstanceDetails
```

- Instance level IP address is public and is PER VM
- Does not replace the VIP of the Cloud Service that contains the VM
- 5 instance level IP addresses allowed, per subscription

Classic

Setting a Reserved Public IP Address



New-AzureReservedIP Options

```
New-AzureReservedIP – ReservedIPName "MyWebsiteIP" –Label "WebsiteIP" –Location "East US"
```

New-AzureVMConfig Options

```
New-AzureVMConfig -Name "WebAppVM" -InstanceSize Small -ImageName $imageName | Add-AzureProvisioningConfig -Windows -AdminUsername $username -Password $password | New-AzureVM -ServiceName "MyWebAppService" -ReservedIPName "MyWebSiteIP" -Location "East US"
```

Get-AzureReservedIP

```
Get-AzureReservedIP – ReservedIPName "MyWebsiteIP"
```

- Note that the –ReservedIPName does not refer to an IP address
- The Cloud Service with VMs can not already exist, they need to be created new via PowerShell
- To retrieve the reserved address information for confirmation of the IP address, use Get-AzureReservedIP

Configuring Azure Load Balancing (ILB)



Add-AzureInternalLoadBalancer Options

```
Add-AzureInternalLoadBalancer -ServiceName $svc -InternalLoadBalancerName $ilb –SubnetName  
$subnet –StaticVNetIPAddress $IP
```

Add-AzureEndpoint Options

```
Get-AzureVM –ServiceName $svc –Name $vmname | Add-AzureEndpoint -Name $epname -  
Protocol $prot -LocalPort $locport -PublicPort $pubport –DefaultProbe -  
InternalLoadBalancerName $ilb -LBSetName $exILBSet | Update-AzureVM
```

Get-AzureService

```
Get-AzureService -ServiceName $svc | Get-AzureInternalLoadBalancer
```

- The virtual network that contains the Cloud Service and VM must be a regional level network
- A Static internal IP address can also be requested (optional)
- Default, None or Custom load balance probes can be specified
- Get-AzureService confirms the load balanced IP address to use for incoming traffic

User Defined Routes



New-AzureRouteTable Options

```
New-AzureRouteTable –Name FrontEndSubnetRouteTable –Location 'West US' –Label "Route table for front end subnet"
```

Add a Route to a route table

```
Get-AzureRouteTable FrontEndSubnetRouteTable | Set-AzureRoute –RouteName FirewallRoute –AddressPrefix 10.2.0.0/16 –NextHopType VirtualAppliance –NetHopIpAddress 10.1.1.10
```

View Applied Routes

```
Get-AzureVM –Name FWAppliance1 –ServiceName ProductionVMs | Get-AzureEffectiveRouteTable
```

RDP Changes



Updating RDP Username/Password

```
Get-AzureVM –ServiceName $cloudSvcName –Name $vmName | Set-AzureVMAccessExtension –  
UserName $adminUsername –Password $adminPassword | Update-AzureVM
```



Fixing RDP Issues

```
Get-AzureVM –ServiceName $cloudSvcName –Name $vmName | Set-AzureVMAccessExtension |  
Update-AzureVM
```

Classic

Linux Provisioning Options



Add-AzureProvisioningConfig Options

Linux

- LinuxUser \$user -Password \$pwd
- DisableSSH , -NoSSHEndpoint
- SSHKeyPairs, -SSHPublicKeys
 - installed from certificates deployed in cloud service

Classic

Deploying into a Virtual Network



Virtual Machine Settings

Set Subnet on VM with Set-AzureSubnet



Deployment Settings

Set Virtual Network -VNetName

Set DNS Servers - New-AzureDns and -DNSSettings

Classic

Provisioning into a Virtual Network and Active Directory

```
$dom = 'contoso'  
$jdom = 'contoso.com'  
$onPremDNS = New-AzureDns -IPAddress '192.168.1.4' -Name 'OnPremDNS'  
$cloudDNS = New-AzureDns -IPAddress '10.1.1.4' -Name 'CloudDNS'  
$computerOU = $advmou = 'OU=AzureVMs,DC=contoso,DC=com'  
  
New-AzureVMConfig -Name 'myvm1' -InstanceSize 'Small' -ImageName $img |  
Add-AzureProvisioningConfig -WindowsDomain –AdminUserName $adminName -Password $pwd -Domain  
$dom`  
-DomainUserName $domUser -DomainPassword $dpwd -JoinDomain $jdom `  
-MachineObjectOU 'AzureVMs' |  
Set-AzureSubnet -SubnetNames 'AppSubnet' |  
New-AzureVM –ServiceName $svc -AffinityGroup 'adag' `  
-VNetName 'ADVNet' -DnsSettings $onPremDNS, $cloudDNS
```

Classic

VM Storage



Data Disks

Add/Remove data disks at boot or while running
Create a blank data disk or attach an existing disk



Modify Cache Settings of OS Disk or Data Disk

Modifying OS Disk while running requires reboot

Classic

Microsoft Confidential

Data Disk Creation

- Creating a new VM with a Data Disk

```
New-AzureVMConfig -Name 'myvm1' -InstanceSize 'Small' -ImageName $img |  
    Add-AzureProvisioningConfig -Windows -AdminUserName $adminName -Password  
    $pwd |  
    Add-AzureDataDisk -CreateNew -DiskSizeInGB 10 -DiskLabel 'myddisk' -LUN 0 |  
    New-AzureVM -ServiceName $cloudSvcName
```

- Add a new Data Disk to an existing VM

```
Get-AzureVM -ServiceName 'myvm1' |  
    Add-AzureDataDisk -CreateNew -DiskSizeInGB 10 -DiskLabel 'myddisk' -LUN 1 |  
    Update-AzureVM
```

Modifying Cache Settings

- Set host caching on an OS Disk during provisioning

```
(if no service exists) New-AzureService –ServiceName $svc –Location $location  
$myVM = New-AzureVMConfig -Name 'myvm1' -InstanceSize 'Small' -ImageName $img |  
    Add-AzureProvisioningConfig -Windows -AdminUserName $adminName -Password  
        $pwd  
Set-AzureOSDisk -HostCaching 'ReadOnly' –VM $myVM  
New-AzureVM -ServiceName $cloudSvcName –VMs $myVM
```

- Set host caching on an existing Data Disk in a running VM

```
Get-AzureVM -ServiceName $cloudSvcName -Name 'myvm1' |  
    Set-AzureDataDisk -HostCaching 'ReadWrite' -LUN 0 |  
Update-AzureVM
```

Configuring Endpoints

- Add endpoints at creation

```
New-AzureVMConfig -Name 'myvm1' -InstanceSize 'Small' -ImageName $img |  
    Add-AzureProvisioningConfig -Windows -AdminUserName $adminName -Password  
    $pwd |  
    Add-AzureEndpoint -LocalPort 80 -PublicPort 80 -Name http -Protocol tcp |  
    Add-AzureEndpoint -LocalPort 443 -PublicPort 443 -Name https -Protocol tcp |  
    New-AzureVM -ServiceDescription $cloudSvcName
```

- Modify endpoints at runtime

```
Get-AzureVM -ServiceName $cloudSvcName -Name 'myvm1' |  
    Add-AzureEndpoint -LocalPort 53 -PublicPort 53 -Name dns -Protocol udp |  
    Remove-AzureEndpoint -Name https |  
    Update-AzureVM -ServiceDescription $cloudSvcName
```

Disk and Image Repository



Microsoft, Partner and User
OS Images



OS Disks or Data Disks
Disks

Classic

```
Get-AzureVMImage # Return all
Get-AzureVMImage | Where { $_.Category -eq 'Microsoft' } # Return Microsoft
Get-AzureVMImage | Where { $_.Category -eq 'User' } # Return Custom
Get-AzureVMImage | Where { $_.Category -eq 'Partner' } # Return Partner Images
Get-AzureVMImage | Where { $_.OS -eq 'Windows' } # Return only Windows OS images
Remove-AzureVMImage -ImageName 'myimg' -DeleteVHD # Delete image and storage
Add-AzureVMImage -OS 'Windows' -ImageName 'MyWinImage' -MediaLocation
'http://storageaccount/vhds/winimage.vhd' # Add Existing VM Image from Storage

Get-AzureDisk # Return all
Get-AzureDisk | Where { $_.AttachedTo -eq $null } # Return all not attached to a VM
Get-AzureDisk | Where { $_.OS -eq $null } # Return only data disks
Get-AzureDisk | Where { $_.OS -eq 'Windows' } # Return only Windows OS disks
Remove-AzureDisk -DiskName 'mydisk' -DeleteVHD # Delete disk and storage
Add-AzureDisk -OS 'Windows' -DiskName 'MyWinDisk' -MediaLocation
'http://storageaccount/vhds/winosdisk.vhd' # Add Existing OS Disk from Storage
Add-AzureDisk -DiskName 'MyDataDisk' -MediaLocation 'http://storageaccount/vhds/datadisk.vhd'
# Add Existing Data Disk from Storage
```

Virtual Network Operations

- View and set virtual network configuration

```
Get-AzureVNetConfig | Select -Expand XMLConfiguration
```

```
Set-AzureVNetConfig -ConfigurationPath 'c:\Network\MyNetCFG.xml'
```

- Start and stop virtual network gateway

```
Set-AzureVNetGateway -Disconnect -VNetName 'MyVNet'
```

```
    -LocalNetworkSiteName 'MySite'
```

```
Set-AzureVNetGateway -Connect -VNetName 'MyVNet'
```

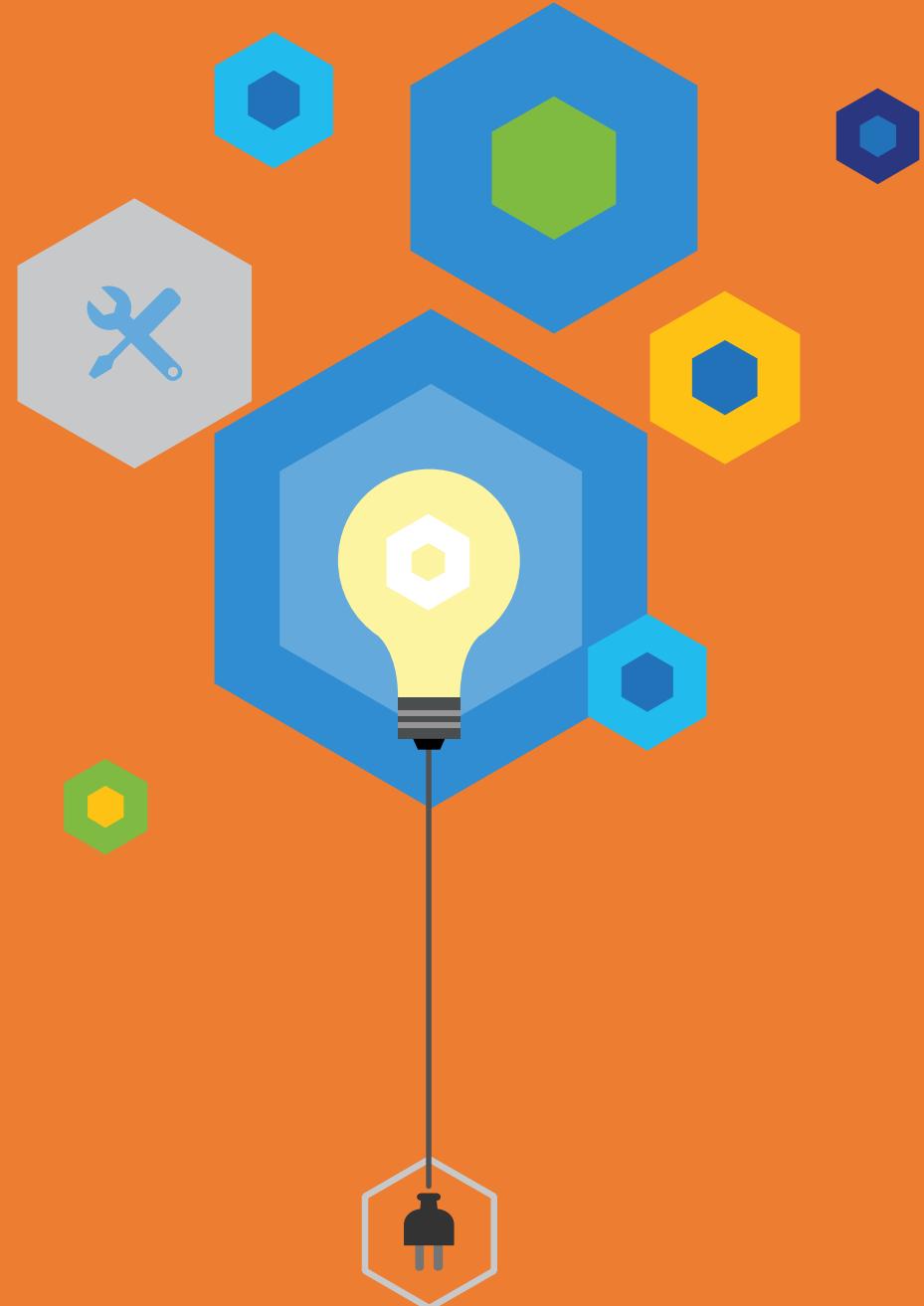
```
    -LocalNetworkSiteName 'MySite'
```

- View virtual network status

```
Get-AzureVNetConnection -VNetName 'MyVNet'
```

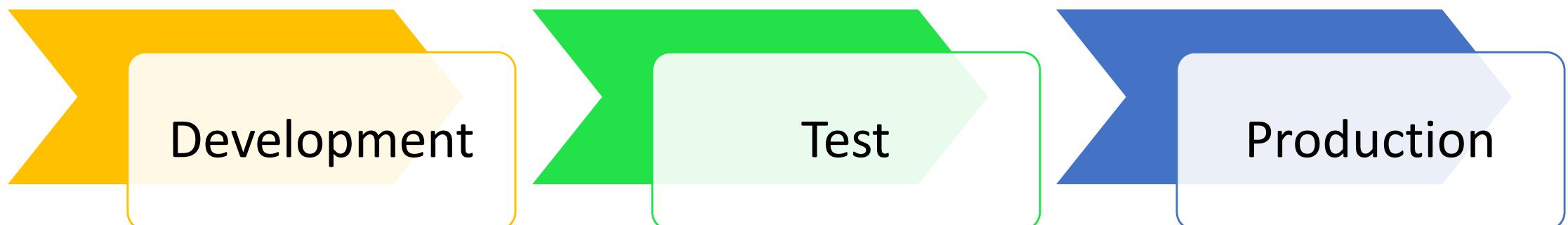
Classic

PowerShell Progression ...

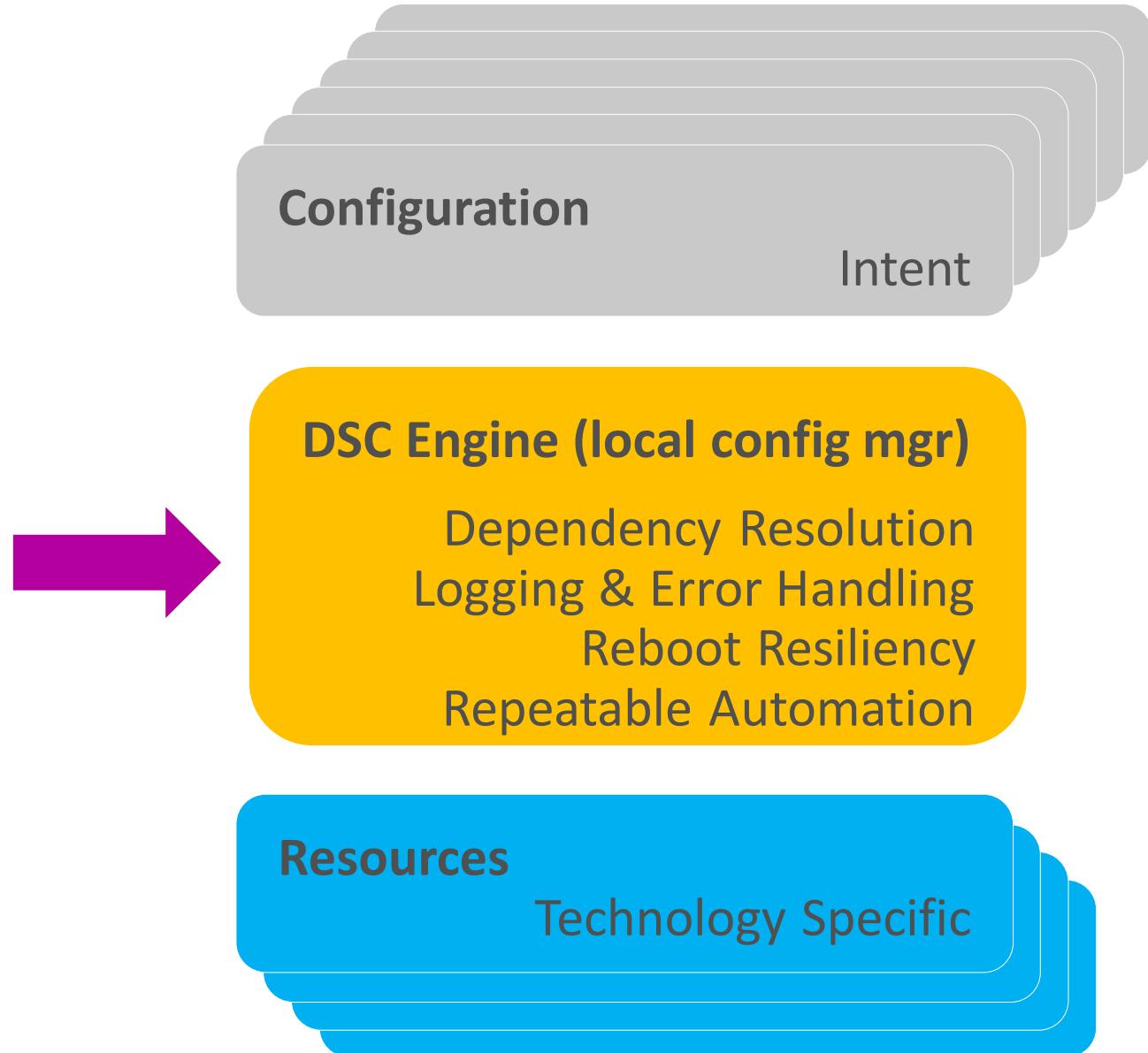
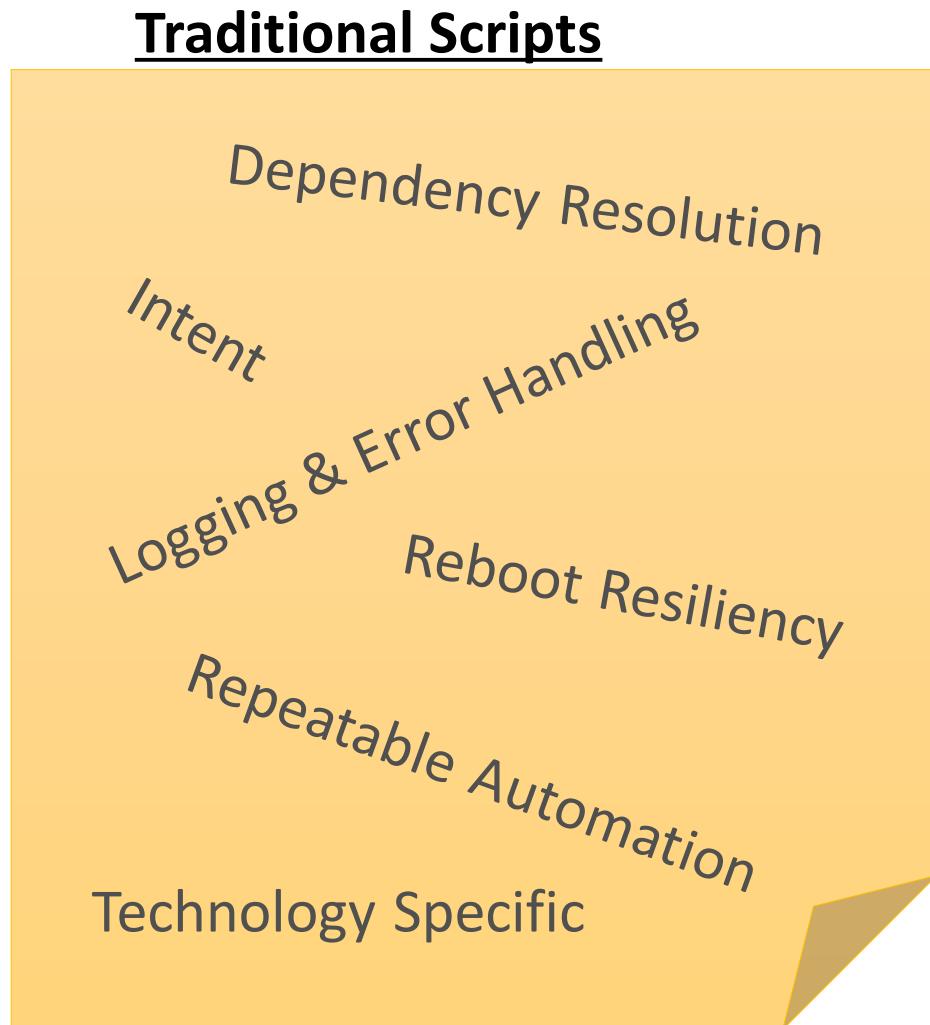


PowerShell Desired State Configuration (DSC)...

- Simplifies configuration
- Prevents configuration drift
- Flexible deployment options
- Enables continuous deployment



DSC Leveraged

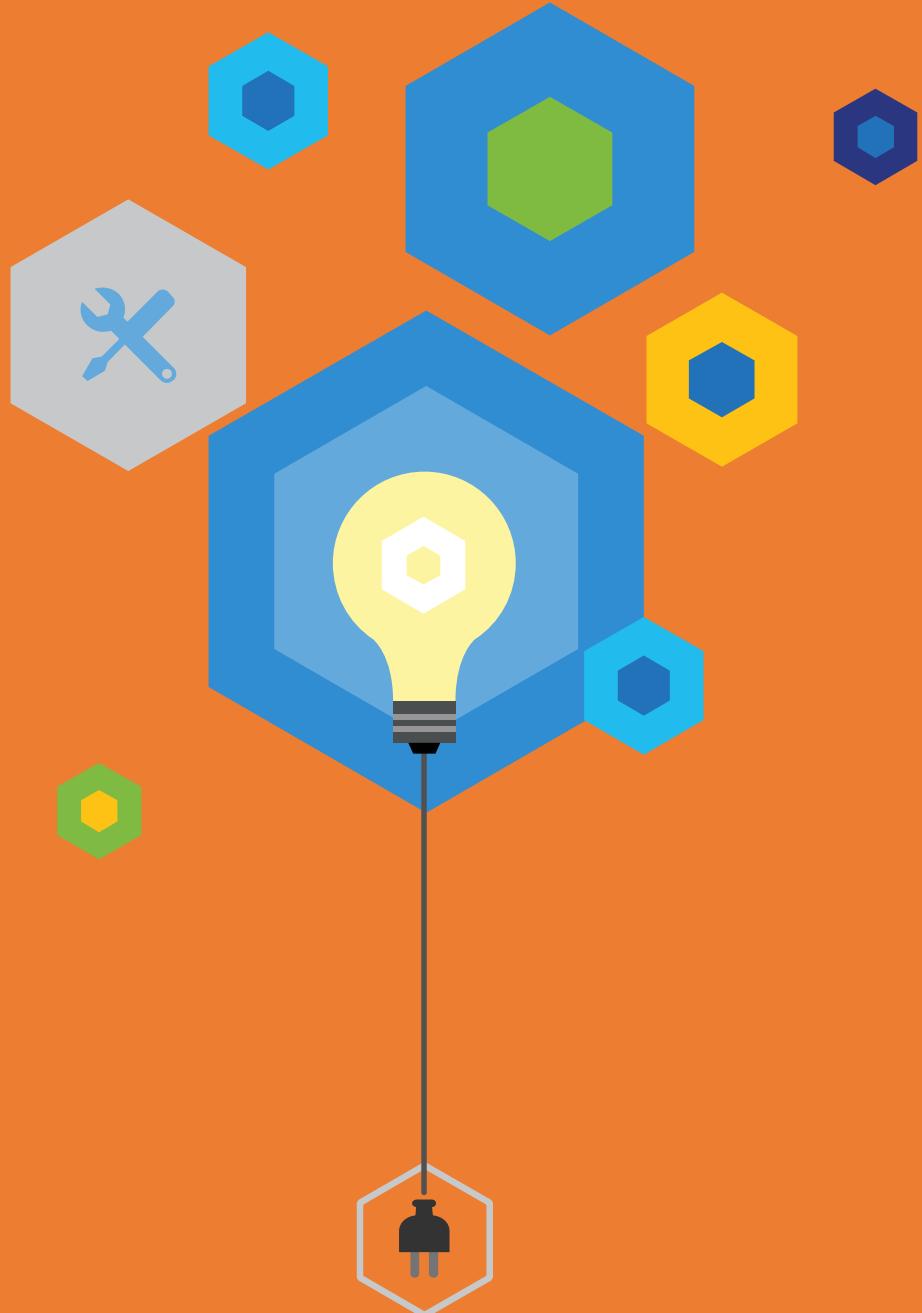


DSC Terminology

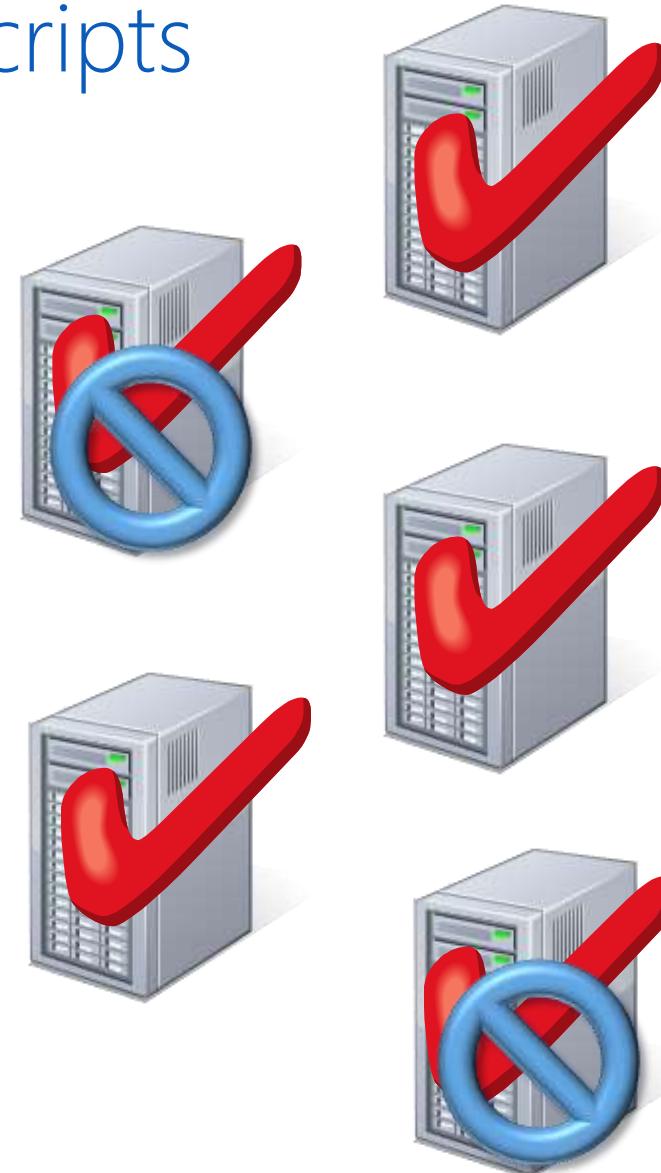
- **Configuration** – this is a new PowerShell keyword used to collect a block of configuration information for the VM
- **Node** – this is the name of the target computer, which can be a variable
- **Resource** – DSC comes with a set of built in system configuration resources that will be deployed to the VM for configuration
- **MOF file** - DSC tells the target nodes what configuration they should have by sending a MOF file with that information to each node, where the Local Configuration Manager implements the desired configuration
- **Local Configuration Manager (LCM)**– DSC engine that runs on all target nodes. Calls configuration resources that are included in the configuration script
- **DSC Pull Server** – LCM on node performs compliance check and if necessary pulls script from another server

Simplify Configuration

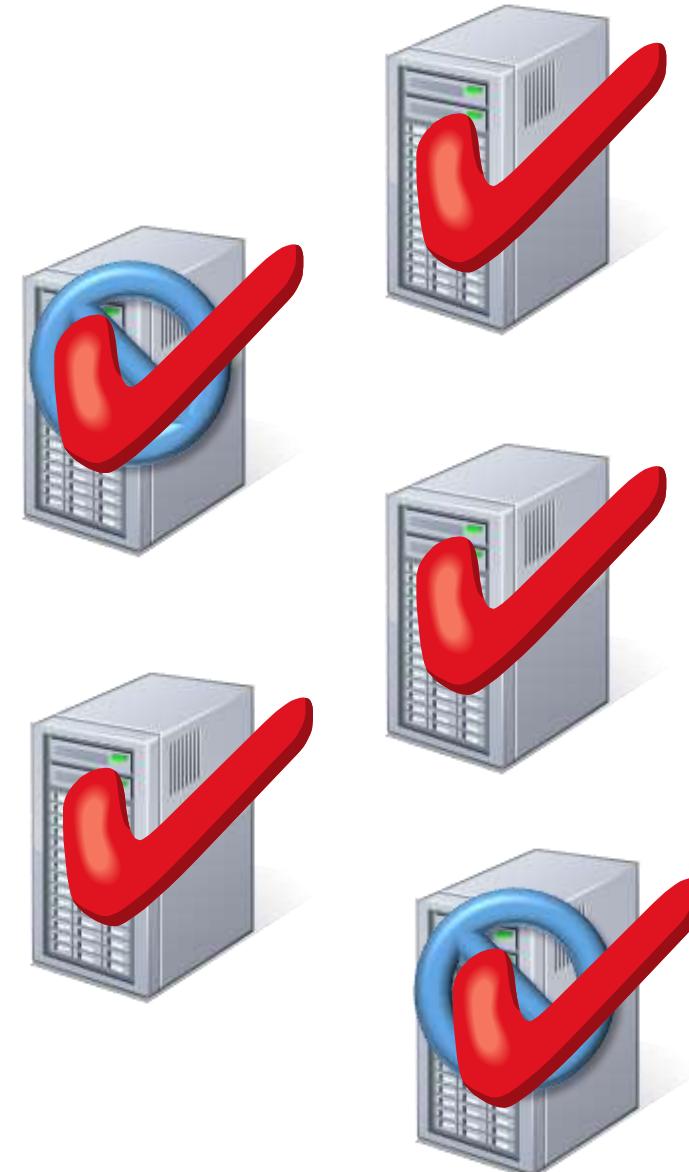
Set up web servers



Configuration Drift with standard scripts



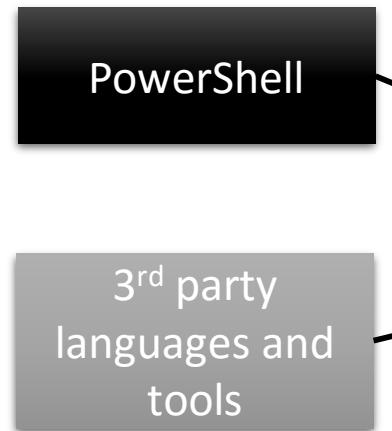
Configuration Drift with DSC



Push Model Components & Phases

Authoring Phase

(May include imperative as well as declarative code)



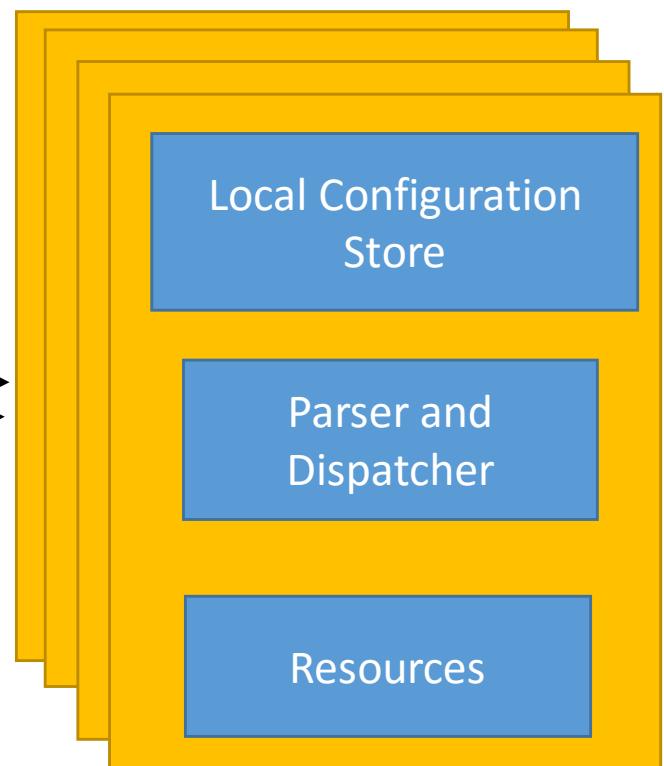
When authoring in PowerShell:

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

Staging Phase

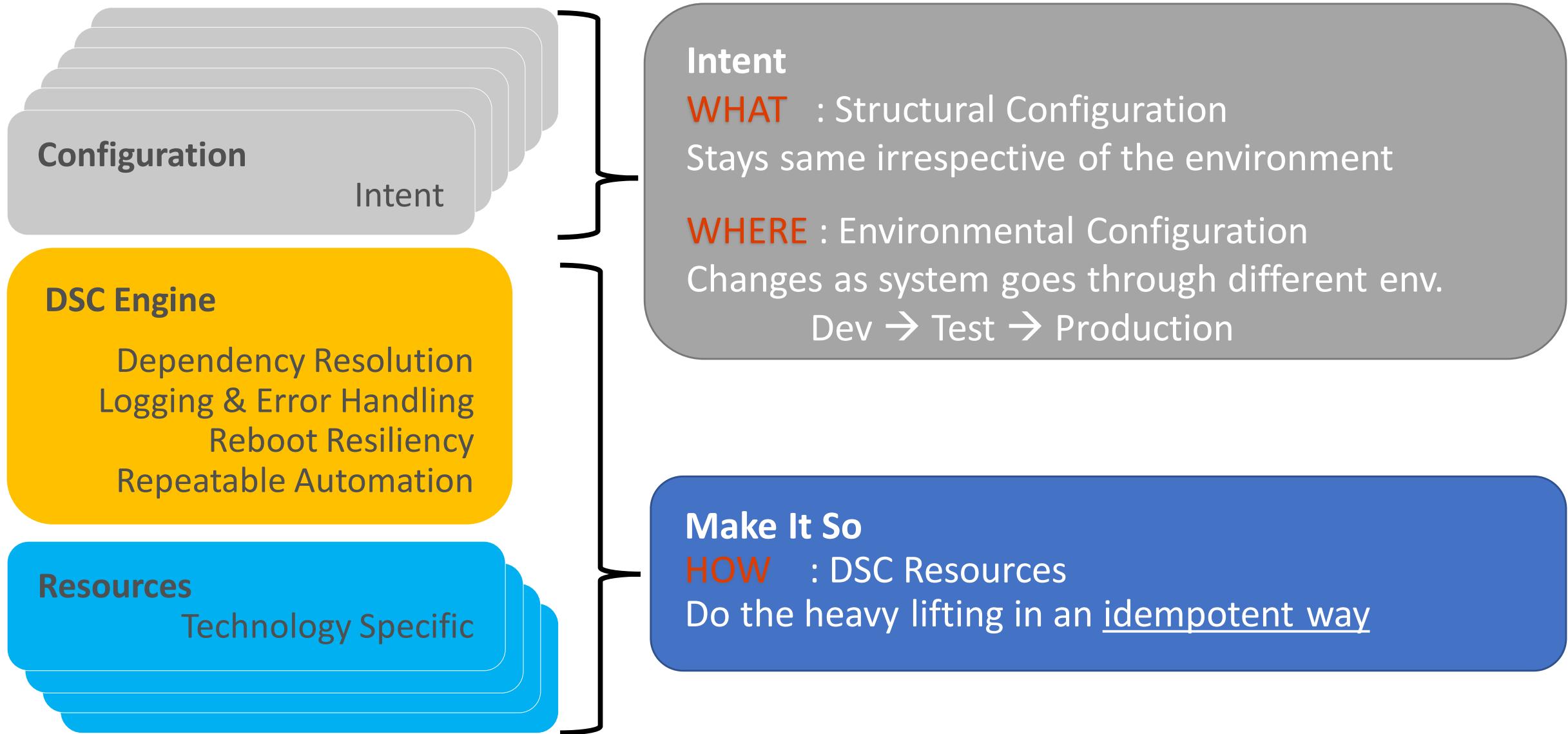
- Fully declarative configuration representation using DMTF standard MOF instances
- Configuration is calculated for all nodes

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



Resources implement changes:
• Monotonic
• Imperative
• Idempotent

DSC Decouples ...



Configuration and Continuous Deployment

Intent



Environment Configuration (Dev -> Test -> Production)

```
$SystemDrive = "C:"  
$DemoFolder = "$SystemDrive\Demo"  
$global:WebServerCount = 3
```

...

Structural Configuration

```
WindowsFeature IIS {  
    Name = "Web-Server"  
    Ensure = "Present"  
}  
...
```

Make It So



Idempotent Automation

```
foreach -parallel ($featureName in $Name)  
{  
    $feature = Get-WindowsFeature -Name $featureName  
    if(($Ensure -eq "Present") -and (!$feature.Installed))  
    {  
        Install-WindowsFeature -Name $featureName  
    }  
    ....  
}  
...
```

DSC Enables ...

Configuration as Code

- Conflict detection
- Single source, multiple environments
- Composable, common components

DevOps

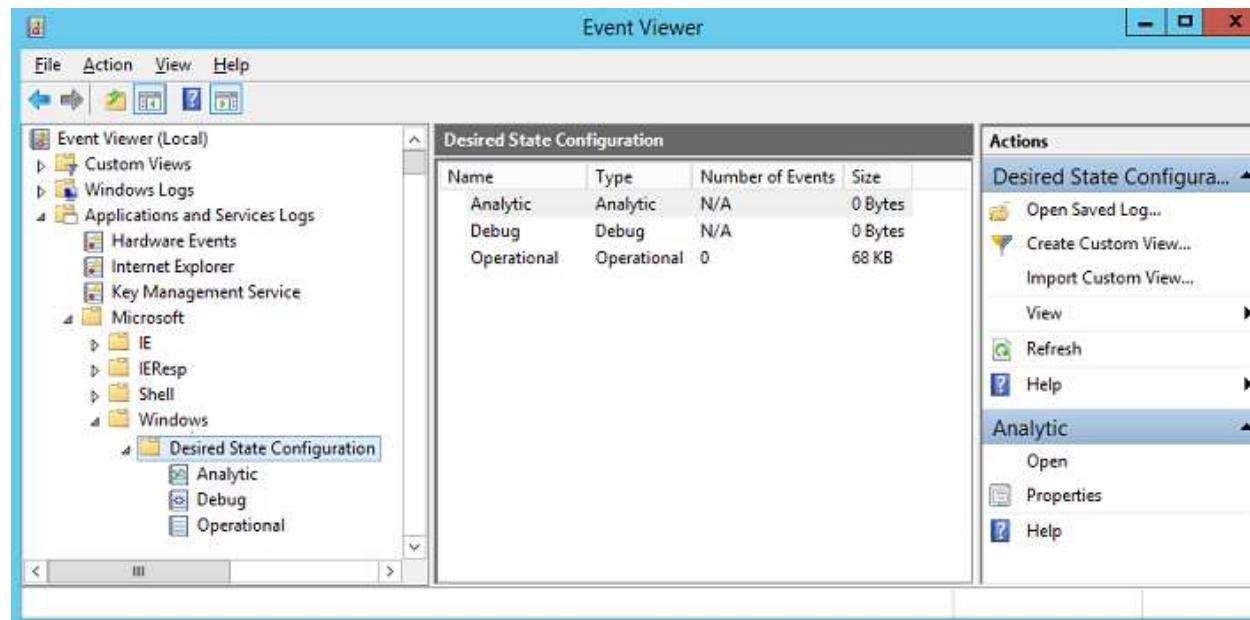
- Common toolset for Dev & Ops
- Apply Dev practices to Ops
- Continuous deployment

Cloud Scale

- Reduce complexity from within
- On-demand system creation and tear down

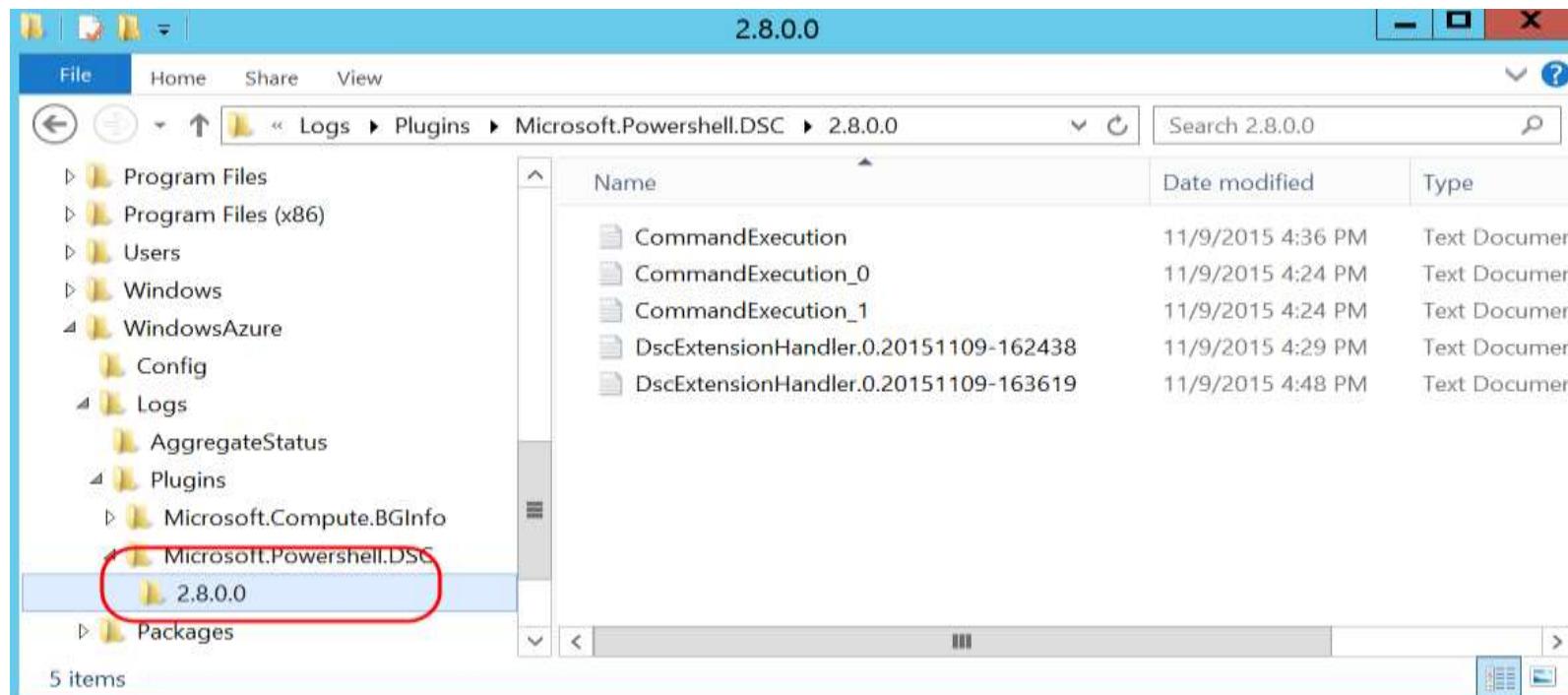
Troubleshooting DSC

- Locating the DSC Event logs
 - Operation Log – contains all error messages and is used to identify problems
 - Analytic Log – shows a higher volume of events and can be used to identify *where* a problem occurred
 - Debug Log - contains logs that can help you understand *how* the errors occurred



Troubleshooting DSC (con't)

- Locating the log files – located at C:\WindowsAzure\Logs\Plugins\Microsoft.PowerShell.DSC\dscversion#



DSC Log Files

