WorkshopPLUS

Microsoft Azure Infrastructure as a Service

Microsoft Azure Automation Desired State Configuration (DSC) - ARM

Student Lab Manual

V1.3, October 12, 2016

Information in this document is subject to change without notice. The example companies, organizations, products, people, and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred. 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.

Microsoft may have patents, patent applications, trademarked, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any 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.

© 2014 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, MS, Windows, Windows NT, MSDN, Active Directory, BizTalk, SQL Server, SharePoint, Outlook, PowerPoint, FrontPage, Visual Basic, Visual C++, Visual J++, Visual InterDev, Visual SourceSafe, Visual C#, Visual J#,  and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries.

Other product and company names herein may be the trademarks of their respective owners.

Contents

[Microsoft Azure Automation DSC 4](#_Toc464032987)

[Prerequisites 5](#_Toc464032988)

[Task 1 – Create an Azure Automation Account 5](#_Toc464032989)

[Task 2 – Review the DSC script 11](#_Toc464032990)

[Task 3 – Confirm the node configuration 17](#_Toc464032991)

# Microsoft Azure Automation DSC

In this lab, you will use Azure Automation Desired State Configuration (DSC) to configure an existing VM with IIS and components.

You'll learn:

* How to create an Automation Account
* How to provision an Azure Active Directory user that can be used by Azure Automation to access resources in an Azure Subscription
* How to deploy and compile a DSC configuration to Azure Automation
* How to deploy the DSC configuration to a VM (node)

## Prerequisites

The following is required to complete this hands-on lab:

* A Microsoft Azure subscription
* One or more virtual machines running in the subscription that can be shutdown. You can create the VMs using the Azure portal or PowerShell. Please start this process **before** continuing on to Task 1. Put the VM(s) in to any resource group name of your choosing and the machine needs to have a public IP address so that you can RDP in to the machine. It is best for this lab to have a new clean VM with no other extensions applied; if you already have a DSC extension applied to a machine, you could see conflicting installation information.

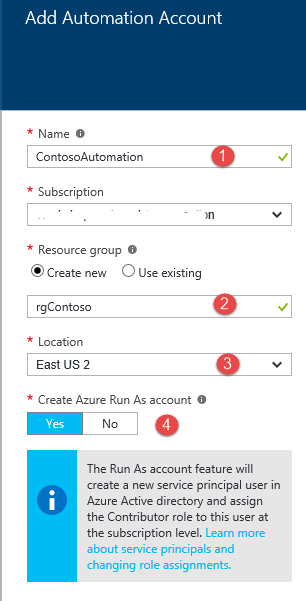
## Task 1 – Create an Azure Automation Account

Please see the prerequisites listed above and create the required VM before continuing.

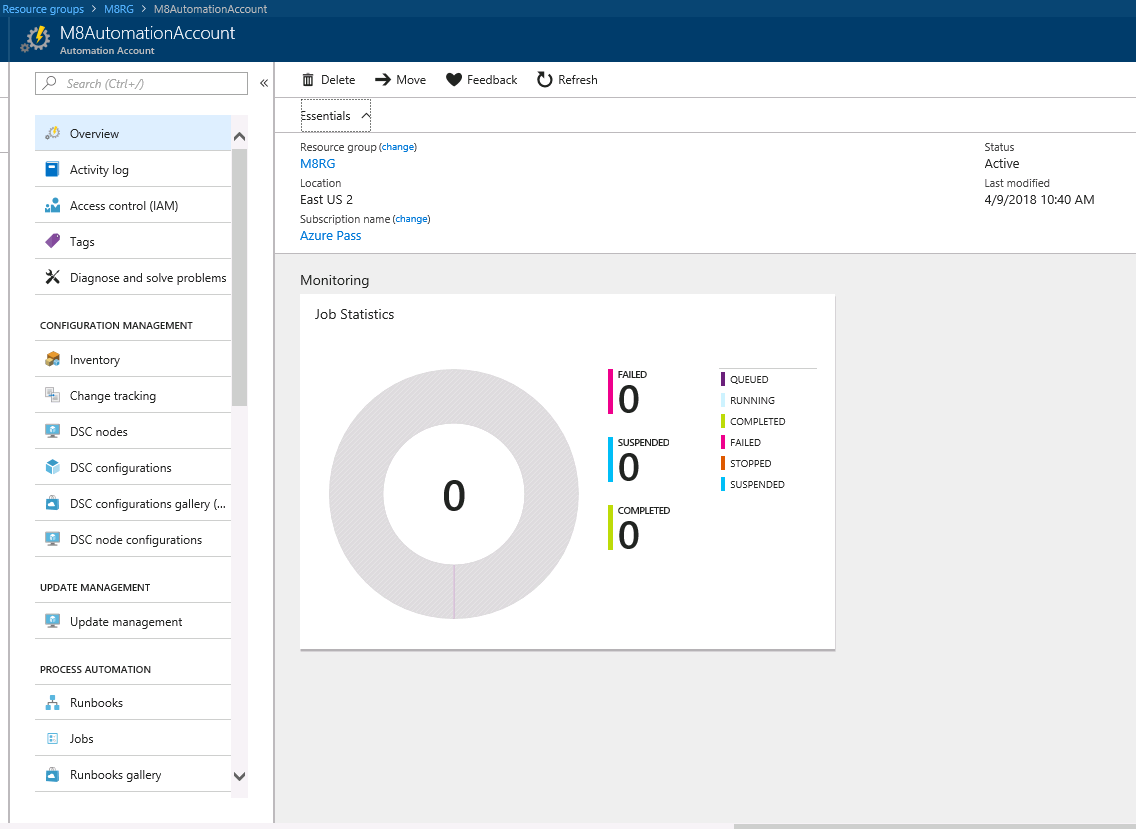
NOTE: If you have already created an Azure Automation account from previous exercises, you can use that same automation account for this lab. Skip to **Task 2** if you have already setup an automation account and want to use that account.

To begin using Azure Automation, the first thing you need is an Automation Account. Automation accounts securely contain the resources that configure and run your runbook jobs. An Automation Account is a logical container and security boundary for all automation assets (runbooks, connections, etc.). Assets within an Automation account are visible only to other assets within the account. An Automation Account also provides affinity to a region, which helps with data sovereignty concerns (if your runbook IP or data can't leave a region).

1. Log in to the Azure Management Portal at https://portal.azure.com.
2. Select **+ Create a resource | Monitoring + Management |** **Automation**.
3. Configure the *Automation Account* as follows:
   1. Enter a name for your automation account (unique to your subscription).
   2. Create a new resource group or use an existing one.
   3. Select a Location.
   4. Leave the *Create Azure Run As account* setting as is.



1. Click
2. Once the automation account has been created, you can use the *Resource groups* menu item to find your new or current resource group and then select your automation account.
3. The view of your automation account should look something similar to this:



## Task 2 – Review the DSC script

In this task, you will review the script that will be used to apply a configuration to your virtual machine.

1. Open **PowerShell ISE** as an administrator.
2. Open the file **C:\AzureIaaSWS\M8 - Azure Automation\Labs\AutomationDSC\WebRoleFull.ps1**.

* At the top of the file you will see:

Configuration webrolefull {

This is the ‘name’ of the configuration and you will use this name later to import the configuration into automation DSC. Note: The name of the ps1 file must match the configuration name in the file, otherwise, the configuration upload will fail.

* Next, you will see the features that will be applied to your VM (node). The Name parameters are standard Windows features that can be installed. The Ensure parameter setting of Present means that DSC should install the feature.

$features = @(

@{Name = "Web-Server"; Ensure = "Present"},

@{Name = "Web-WebServer"; Ensure = "Present"},

@{Name = "Web-Common-http"; Ensure = "Present"},

@{Name = "Web-Default-Doc"; Ensure = "Present"}

* Next, you will see the node name, localhost. This name is also important when we import the configuration in to Automation DSC. This script has a for loop that will loop through and apply all the features to the node

node localhost {

foreach ($feature in $features){

WindowsFeature ($feature.Name) {

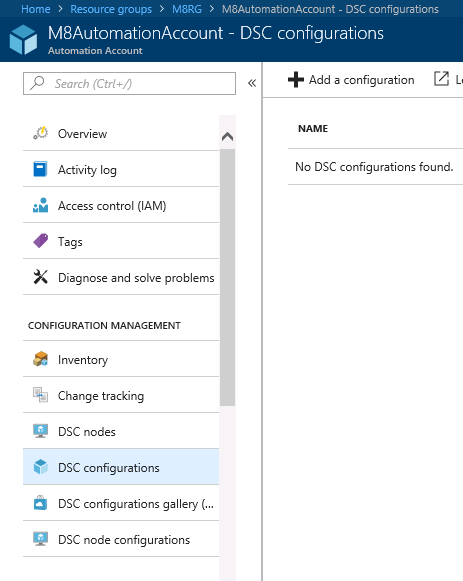
Name = $feature.Name

Ensure = $feature.Ensure

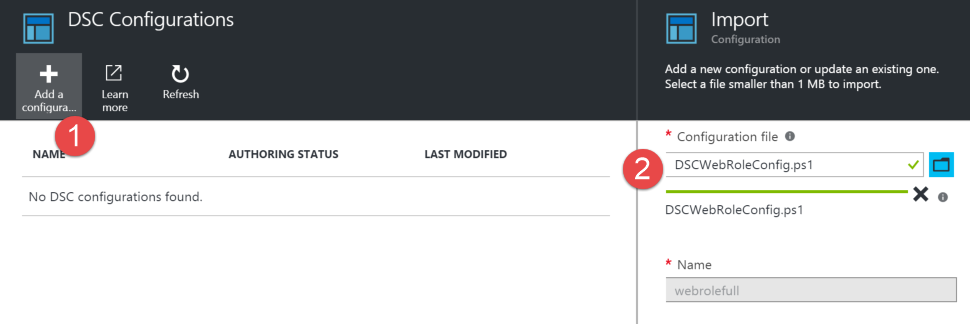
}

}

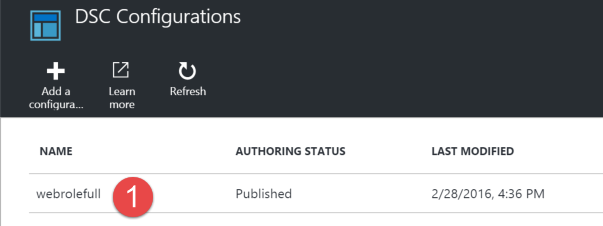
1. Open your automation account in the portal and click on the **DSC Configurations** node under **Configuration Management**.



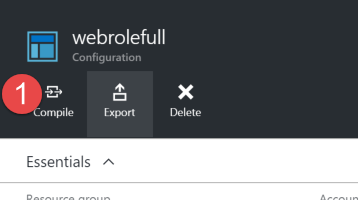
1. Click on the **+Add a Configuration** button and browse to the **WebRoleFull.ps1** file and select it. Click the **OK** button on the Import blade.



Result…notice that the name *WebRoleFull* is the name of the ***configuration***.

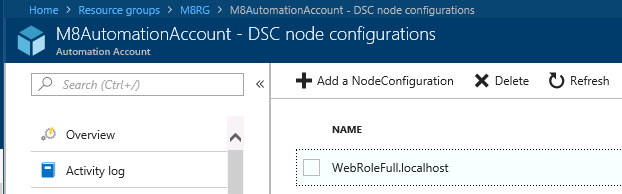


1. Click on the **webrolefull** DSC configuration and a new blade will appear. In this blade, select the **Compile** button. A configuration has to be compiled (into a MOF file) in order for you to apply the configuration to a node.

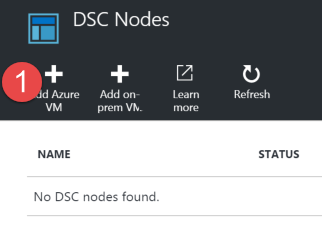


The compile request will be queued, started and then completed. Once the compile has completed, you can click on the compiled jobs and gather any status info.

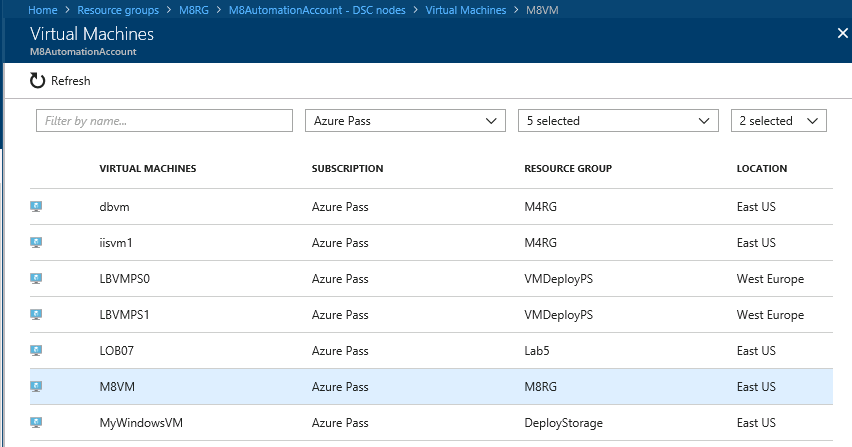
1. Once the compile has completed successfully, go back to the main automation account blade.
2. Under *Configuration Management*, select the **DSC node configurations**. Notice that there is a new DSC Node Configuration item. This is what appears after you do a compile of your DSC Configuration. If you do not have a DSC Node Configuration item present, you will not be able to apply a configuration to a node. (You may need to refresh the browser at this point).



1. Select the **DSC Nodes** icon. This is where you will choose what VM to apply the configuration to.
2. In the DSC Nodes blade, select **+Add Azure VM**.



1. In the Add Azure VMs blade, select the virtual machine that you want to apply the node configuration to.



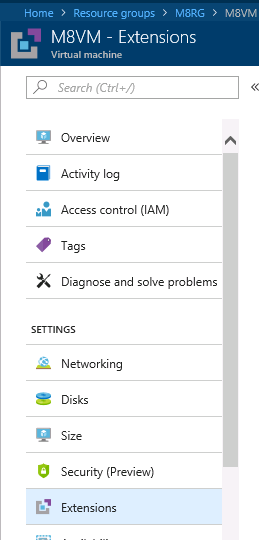
1. Click **+Connect**.
2. On the *Registration* blade, enter the information shown in the table below. The information entered here is very important. Select **OK** once you have finished the information entry.

|  |  |
| --- | --- |
| **RegistrationKey** – the automation account has a primary and secondary key for security. Leave this as the default setting.  **Node Configuration Name** – the name needs to be **<configurationname>.<nodename>.** Recall that this information is in the .ps1 file.  **Refresh Frequency** – Represents the frequency (in minutes) at which the PS DSC Local Configuration Manager contacts the Azure Automation DSC pull server to download the latest node configuration. Minimum 30 minutes.  **Configuration Mode Frequency** – Represents the frequency (in minutes) at which the background application of DSC attempts to implement the current node configuration on the target node.  **Configuration Mode** – how you want DSC to operate once the configuration has been applied to the node.  **Allow Module Override** - Controls whether new configurations downloaded from the Azure Automation DSC pull server are allowed to overwrite the old modules already on the target node. Check this box.  **Reboot Node if Needed** – if anything applied to the machine requires a machine reboot, allow it Check this box.  **Action after Reboot** – you can choose to continue with any missing configuration or stop any further configuration. |  |

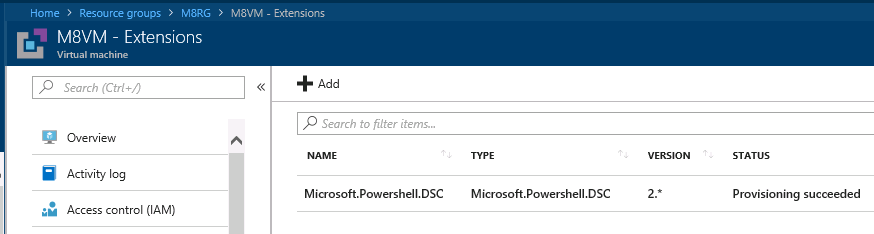
1. At this point, the configuration process begins. The configuration is sent to the DSC pull server. For the initial application of DSC to the node, this will take approximately **20 minutes**.

## Task 3 – Confirm the node configuration

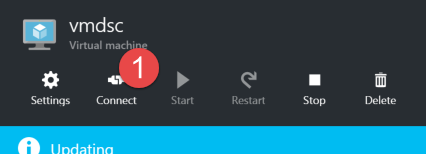
1. Within the Azure portal, go to the resource group that contains the VM that you applied the DSC configuration to.
2. Select the VM that you configured as a DSC node and then select Extensions.



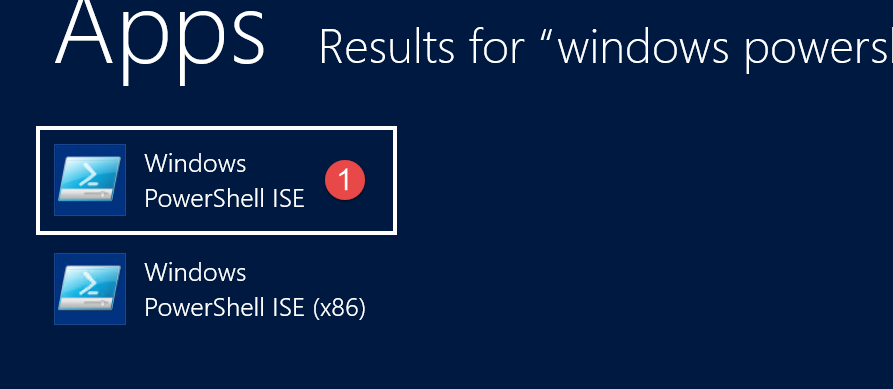
1. If you do not see the **Microsoft.Powershell.DSC** extension with a status of *Provisioning succeeded*, wait until you see that.



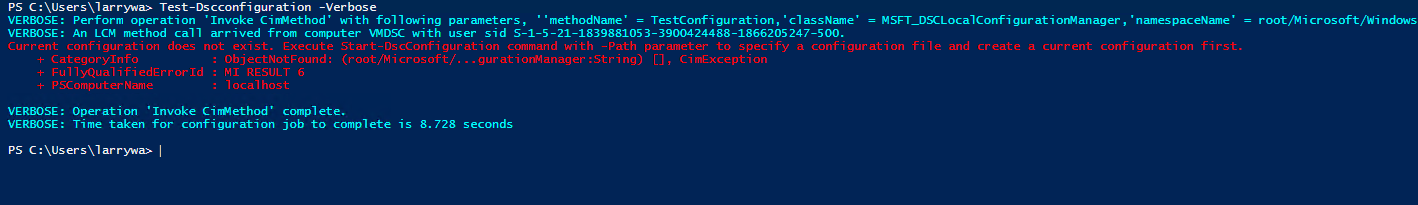
1. Go back to the VM blade and select the **Connect** button and RDP in to the VM.



1. Once you have RDPed in to the VM, you can open up **File Explorer** on the VM to confirm that **C:\inetpub** exists, which usually represents that IIS has been installed. If you do not see *C:\Inetpub*, this means that the configuration has not taken place yet.
2. On the VM, open **PowerShell ISE** as administrator by opening the tile window and typing in ‘Windows PowerShell ISE’. Right-click on the icon and choose **Run as Administrator**.



1. Once *PowerShell ISE* is open, type **Test-DscConfiguration –Verbose** into the command window and select **Enter**. You may receive path errors or an error that tells you that the DSC configuration is currently being applied. Remember, the minimum amount of time it will take to apply the configuration is **30 minutes**.



1. Once the configuration has been applied and you execute the **Test-DscConfiguration -Verbose** line of code, you should see something like this:

PS C:\Users\larrywa> Test-Dscconfiguration -Verbose

VERBOSE: Perform operation 'Invoke CimMethod' with following parameters, ''methodName' = TestConfiguration,'className' = MSFT\_DSCLocalConfigurationManage

r,'namespaceName' = root/Microsoft/Windows/DesiredStateConfiguration'.

VERBOSE: An LCM method call arrived from computer VMDSC with user sid S-1-5-21-1839881053-3900424488-1866205247-500.

VERBOSE: [VMDSC]: LCM: [ Start Test ]

VERBOSE: [VMDSC]: LCM: [ Start Resource ] [[WindowsFeature]Web-Server]

VERBOSE: [VMDSC]: LCM: [ Start Test ] [[WindowsFeature]Web-Server]

VERBOSE: [VMDSC]: [[WindowsFeature]Web-Server] The operation 'Get-WindowsFeature' started: Web-Server

VERBOSE: [VMDSC]: [[WindowsFeature]Web-Server] The operation 'Get-WindowsFeature' succeeded: Web-Server

VERBOSE: [VMDSC]: LCM: [ End Test ] [[WindowsFeature]Web-Server] True in 1.1100 seconds.

VERBOSE: [VMDSC]: LCM: [ End Resource ] [[WindowsFeature]Web-Server]

VERBOSE: [VMDSC]: LCM: [ Start Resource ] [[WindowsFeature]Web-WebServer]

VERBOSE: [VMDSC]: LCM: [ Start Test ] [[WindowsFeature]Web-WebServer]

VERBOSE: [VMDSC]: [[WindowsFeature]Web-WebServer] The operation 'Get-WindowsFeature' started: Web-WebServer

VERBOSE: [VMDSC]: [[WindowsFeature]Web-WebServer] The operation 'Get-WindowsFeature' succeeded: Web-WebServer

VERBOSE: [VMDSC]: LCM: [ End Test ] [[WindowsFeature]Web-WebServer] True in 0.4850 seconds.

VERBOSE: [VMDSC]: LCM: [ End Resource ] [[WindowsFeature]Web-WebServer]

VERBOSE: [VMDSC]: LCM: [ Start Resource ] [[WindowsFeature]Web-Common-http]

VERBOSE: [VMDSC]: LCM: [ Start Test ] [[WindowsFeature]Web-Common-http]

VERBOSE: [VMDSC]: [[WindowsFeature]Web-Common-http] The operation 'Get-WindowsFeature' started: Web-Common-http

VERBOSE: [VMDSC]: [[WindowsFeature]Web-Common-http] The operation 'Get-WindowsFeature' succeeded: Web-Common-Http

VERBOSE: [VMDSC]: LCM: [ End Test ] [[WindowsFeature]Web-Common-http] True in 0.5150 seconds.

VERBOSE: [VMDSC]: LCM: [ End Resource ] [[WindowsFeature]Web-Common-http]

VERBOSE: [VMDSC]: LCM: [ Start Resource ] [[WindowsFeature]Web-Default-Doc]

VERBOSE: [VMDSC]: LCM: [ Start Test ] [[WindowsFeature]Web-Default-Doc]

VERBOSE: [VMDSC]: [[WindowsFeature]Web-Default-Doc] The operation 'Get-WindowsFeature' started: Web-Default-Doc

VERBOSE: [VMDSC]: [[WindowsFeature]Web-Default-Doc] The operation 'Get-WindowsFeature' succeeded: Web-Default-Doc

VERBOSE: [VMDSC]: LCM: [ End Test ] [[WindowsFeature]Web-Default-Doc] True in 0.4380 seconds.

VERBOSE: [VMDSC]: LCM: [ End Resource ] [[WindowsFeature]Web-Default-Doc]

VERBOSE: [VMDSC]: LCM: [ Start Resource ] [[WindowsFeature]Web-Dir-Browsing]

VERBOSE: [VMDSC]: LCM: [ Start Test ] [[WindowsFeature]Web-Dir-Browsing]

VERBOSE: [VMDSC]: [[WindowsFeature]Web-Dir-Browsing] The operation 'Get-WindowsFeature' started: Web-Dir-Browsing

