Vision Scope



Azure Automation and Continious Delivery

Prepared for

Microsoft

26-Jan-16

Version 1.0 Draft

Prepared by

**Stanislav Zhelyazkov**

Principal Consultant

stanislav.zhelyazkov@lumagate.com

Contributors

**Add Contributors to Doc Properties**

Table of Contents

[1 Before you begin 3](#_Toc441594865)

[2 Lab 1: Deploy Azure VM via ARM Template 3](#_Toc441594866)

[3 Lab 2: Deploy Automation account via ARM Tempalte 6](#_Toc441594867)

[4 Lab 3: Trigger deployment via Runbook from GitHub commit 8](#_Toc441594868)

[5 Lab 4: Deploy Web Site with Automation DSC 14](#_Toc441594869)

1. Before you begin

The objective of this Lab is to learn how to use Azure Automation trough examples.

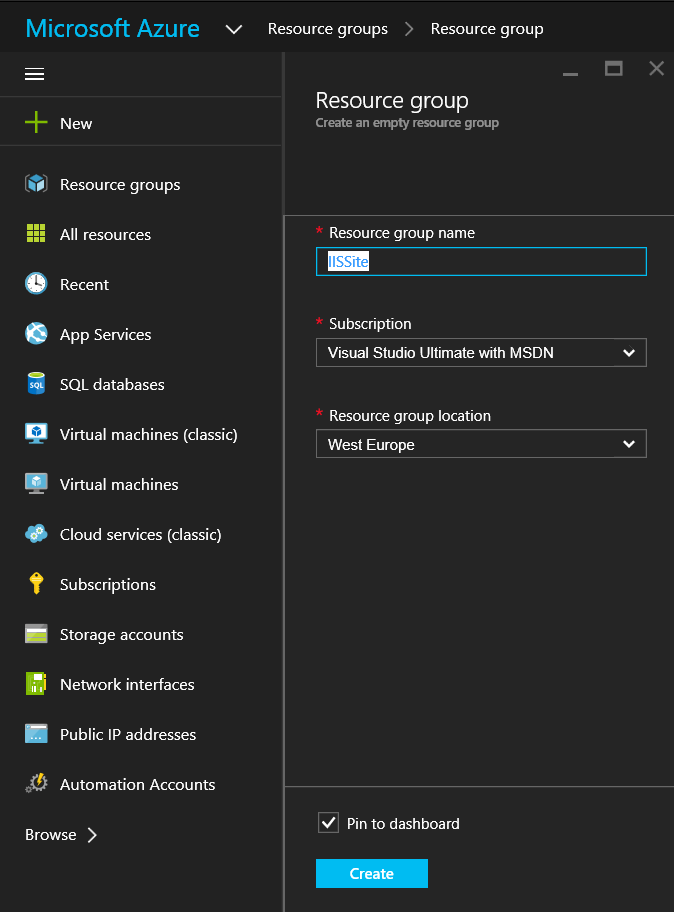
In order to complete these labs, you need to ensure you have the following access and tools properly configured:

* Admin access to an Azure Subscription (minimum Trial Subscription)
* Microsoft Azure PowerShell (1.0.2 or higher)
* PowerShell ISE or Console

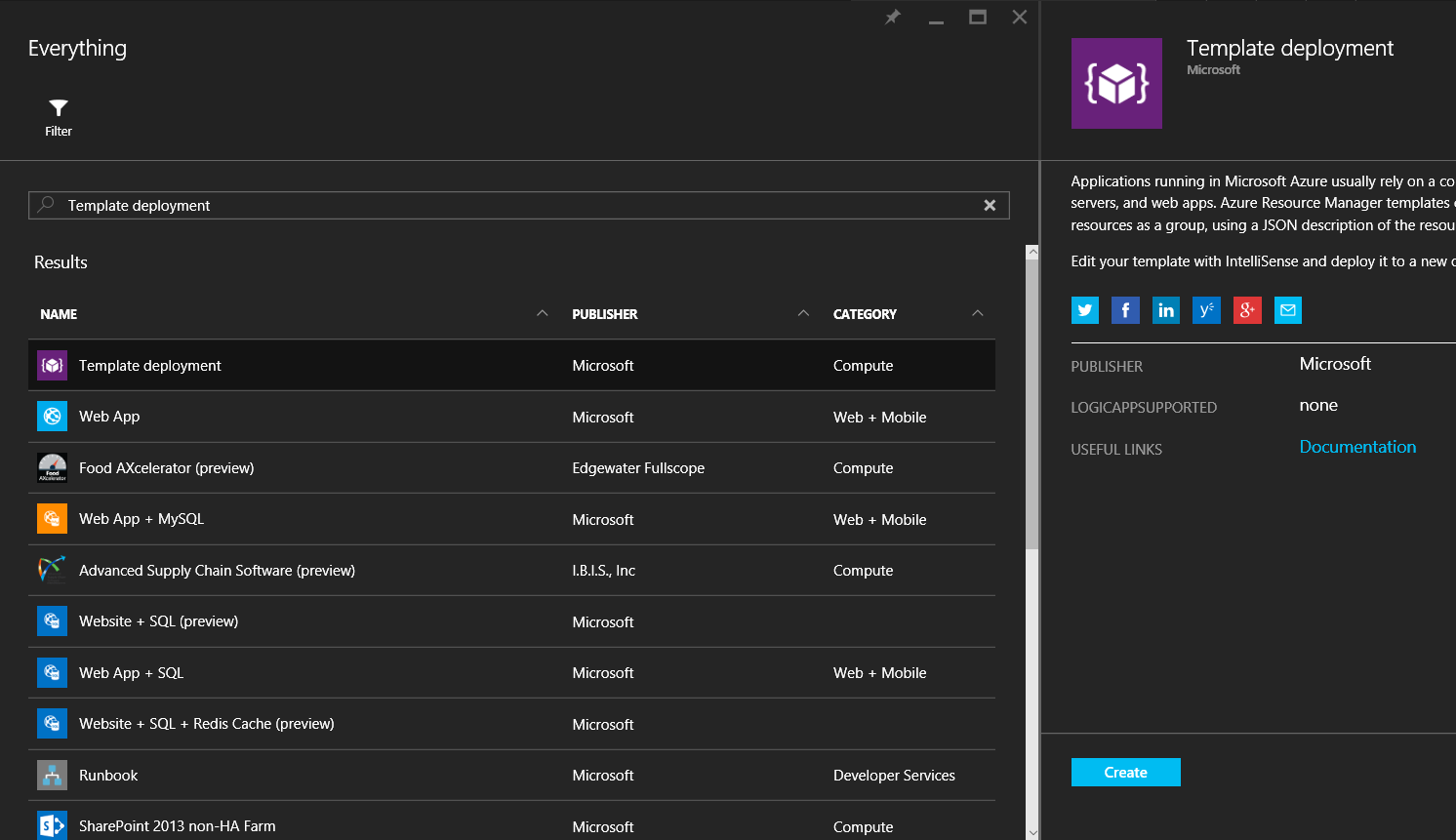
1. Lab 1: Deploy Azure VM via ARM Template

In this lab we will deploy Azure VM that we can use as IIS server.

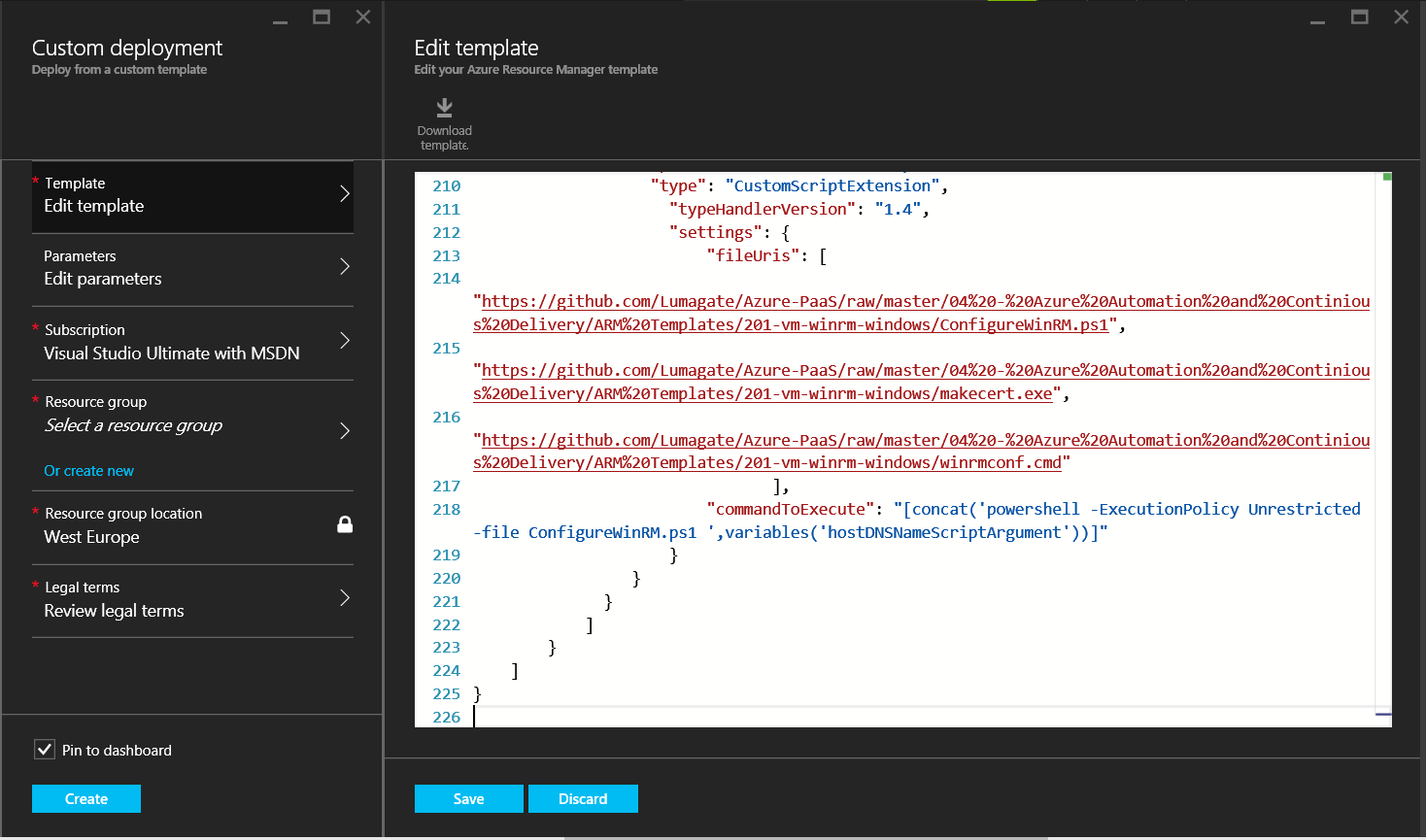
1. From your computer, open your preferred web browser and navigate to <portal.azure.com>.
2. From the hamburger menu on the right click on Resource Groups.
3. Create new resource group named ‘IISSite’ in West Europe region.



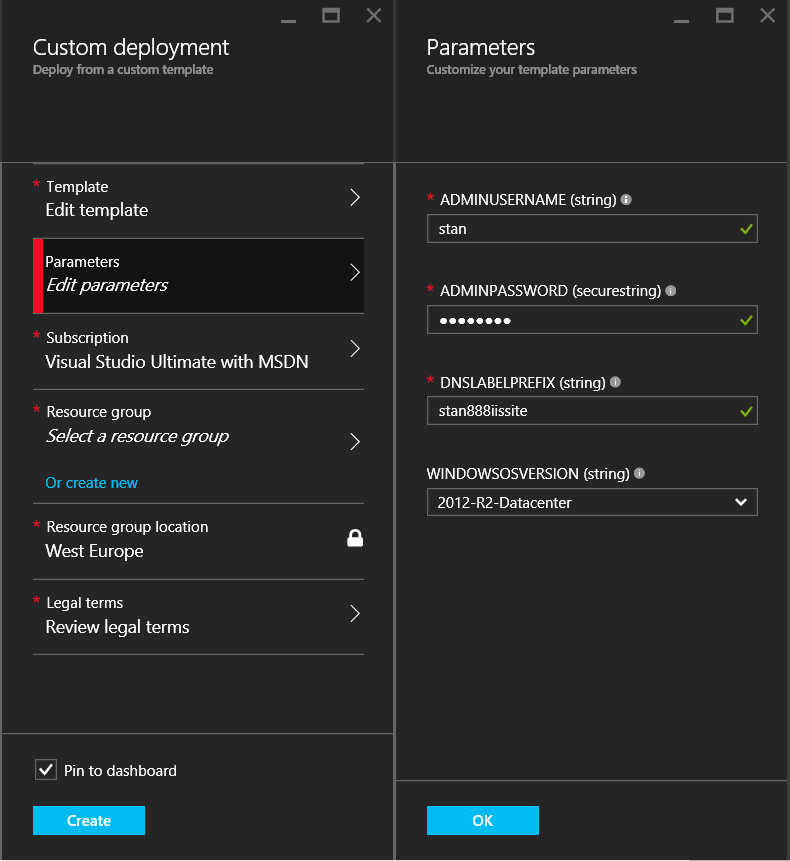
1. From the hamburger menu on the right click New.
2. In the search box type ‘Tempalte deployment’ and click enter.
3. Select the first result and click Create.



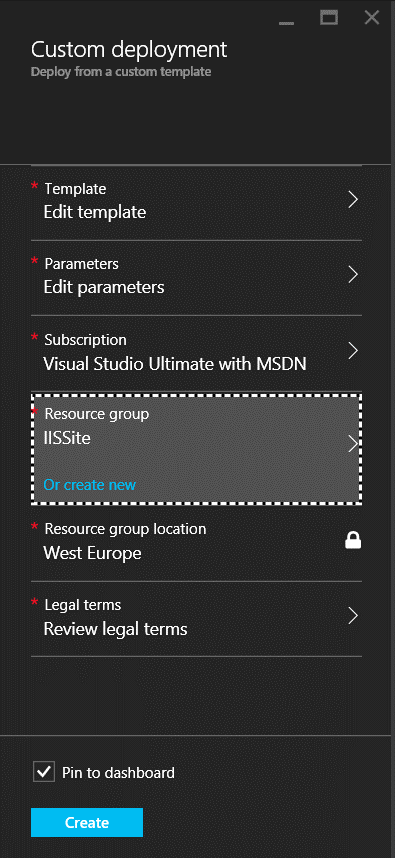
1. Click on Edit template and paste the ARM template from “04 - Azure Automation and Continious Delivery\ARM Templates\201-vm-winrm-windows”. Click Save.



1. In edit Paramaters enter credentials and unqie dns name for the VM. Click OK.



1. From Select a resource Group choose IISSite.



1. On Review Legal Terms click Create.
2. In Custom deployment click Create.
3. Wait until the deployment completes successfully.

**Summary**

You have now successfully completed Lab 1, where you have done the following:

* Deployed Azure VM
* The Azure VM has IIS installed
* The Azure VM has WinRM over HTTPS configured and enabled

1. Lab 2: Deploy Automation account via ARM Tempalte

In this lab, you will deploy automation account with all components needed for the rest of the labs.

1. From your computer, open your preferred web browser and navigate to <portal.azure.com>.
2. From the hamburger menu on the right click New.
3. In the search box type ‘Tempalte deployment’ and click enter.
4. Select the first result and click Create.
5. Click on Edit template and paste the ARM template automation-account.json from “04 - Azure Automation and Continious Delivery\ARM Templates\automation-account”. Click Save.
6. In edit Paramaters enter enter values for the following parameters:

AZURESUBSCRIPTIONIDVALUE : <your Azure Subscription ID>

AZURESERVICEACCOUNTUSERNAME: <Azure AD account username with administrator permissions on your Azure Subscription, ex. Stan@stasoutlook.onmicrosoft.com>

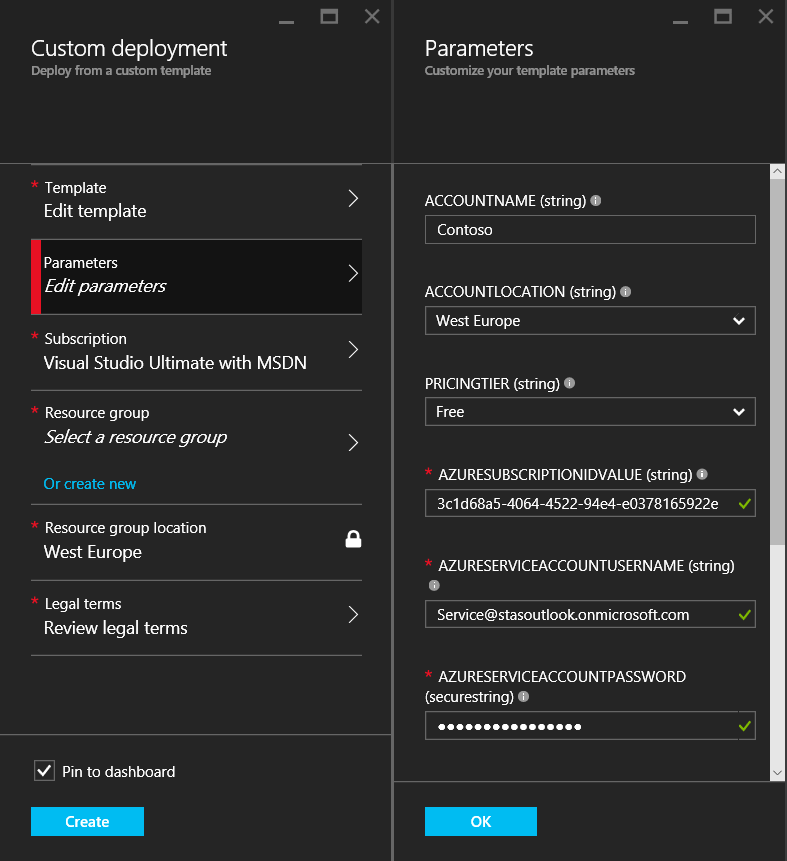
AZURESERVICEACCOUNTPASSWORD: <Password for the above Azure AD Account>

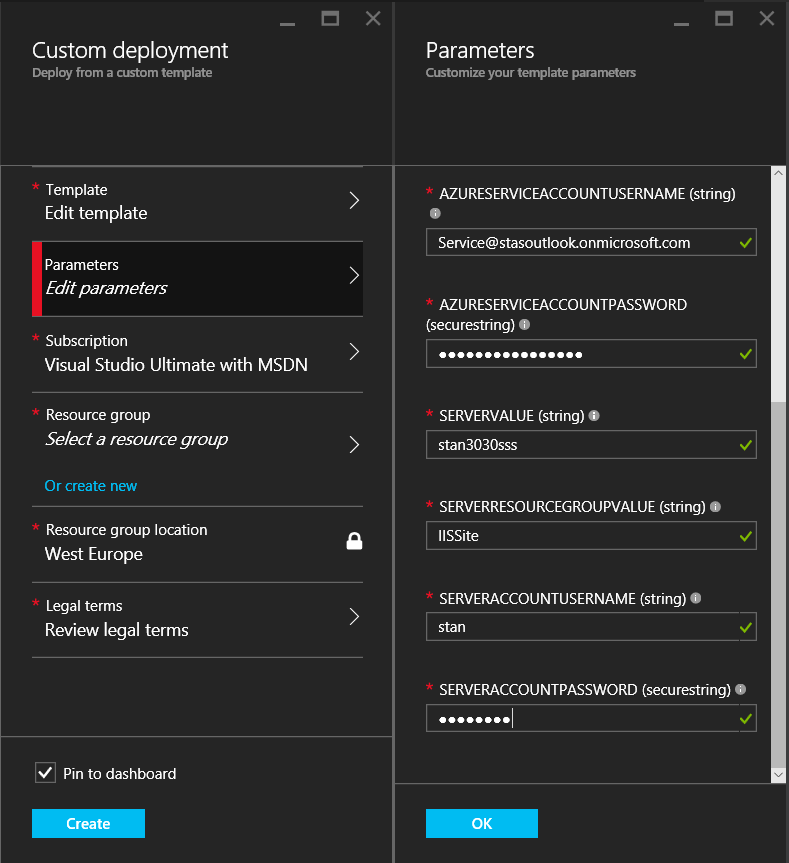
SERVERVALUE: <The name of the server deployed in lab 1>

SERVERRESOURCEGROUPVALUE: <The name of the resource group where the server from lab 1 was deployed>

SERVERACCOUNTUSERNAME: <username entered during the deployment of the server in lab 1>

SERVERACCOUNTPASSWORD: <password for the above username>





1. From Select a resource Group choose IISSite.
2. On Review Legal Terms click Create.
3. In Custom deployment click Create.
4. Wait until the deployment completes successfully.
5. From the righ hamburger menu choose Automation accounts.
6. Select the newly created automation account and look what runbooks and assets were created.

**Summary**

You have now successfully completed Lab 2, where you have done the following:

* Deployed Automation account
* Deployed assets like modules, variables and credentials for automation account
* Depoyed runbooks for automation account

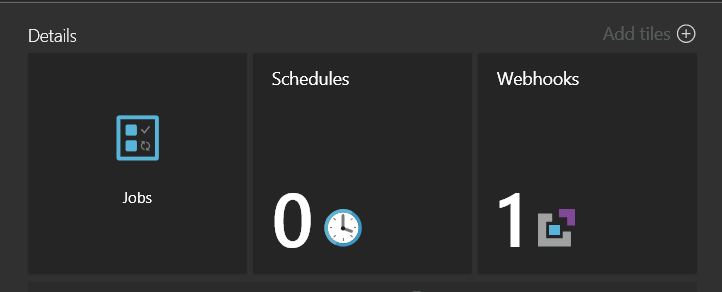
1. Lab 3: Trigger deployment via Runbook from GitHub commit

In this lab, you will create Runbook webhook, create GitHub account and repository, add webhook for commit action and triger deployment by commiting change to repository.

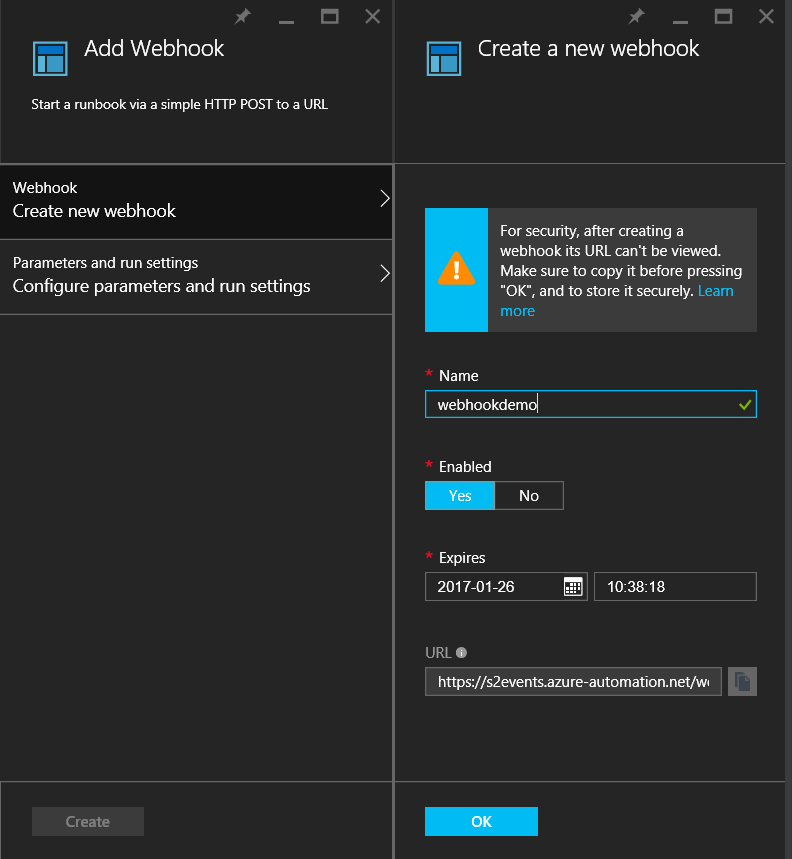
1. From your computer, open your preferred web browser and navigate to <portal.azure.com>.
2. From the hamburger menu on the right click on Automation Accounts.
3. Select the name of your Automation account.
4. From Automation account blade click on Runbooks.



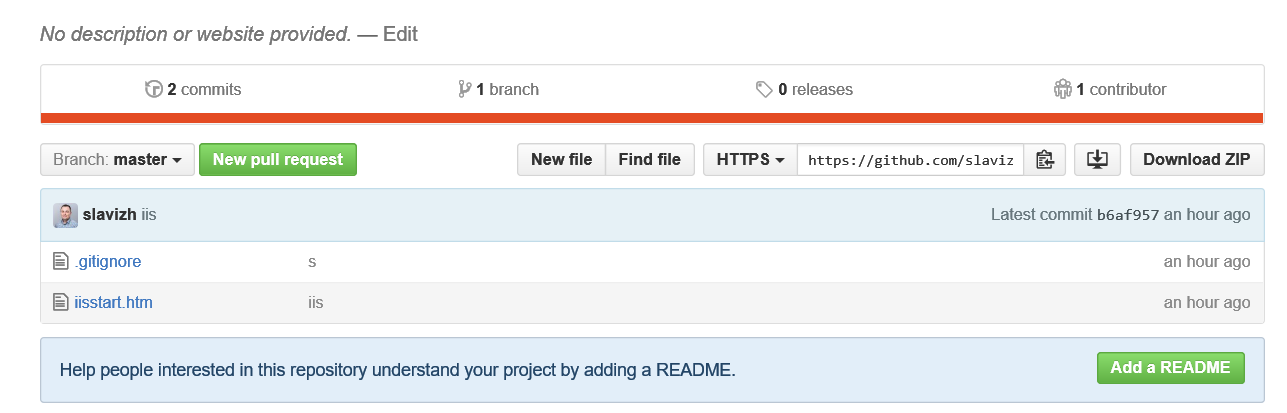
1. From the listed runbooks click on Start-IISDeployment runbook.
2. From the tiles click on Webhooks tile.



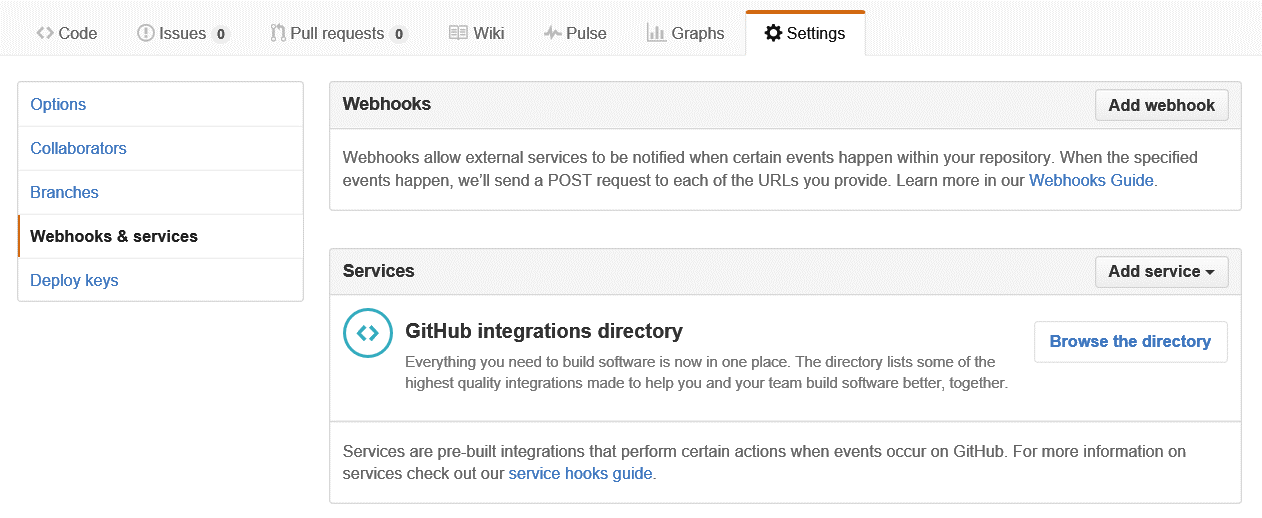
1. From Webhooks blade click on Add Webhook.
2. Click on Create Webhook from Add Webhook blade.
3. Give name for the webhook, copy the URL somewhere. It will be used later. Click OK



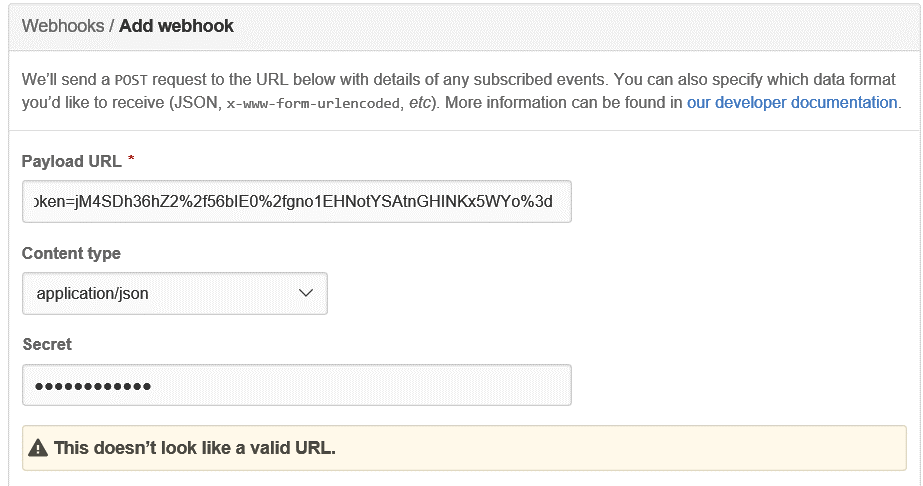
1. Click on Configure Paramaters and run settings. Click OK.
2. On Add Webhook blade click Create. Wait until webhook is created.
3. Naviate to <GitHub.com>, if you have account sign if you do not have create one.
4. Create new Public repository with name of your choice.
5. Clone the repository on your computer with GitHub Desktop.
6. Copy file iisstart.htm from 04 - Azure Automation and Continious Delivery\IISExamplePage to the GitHub Repository. Comit change and sync.



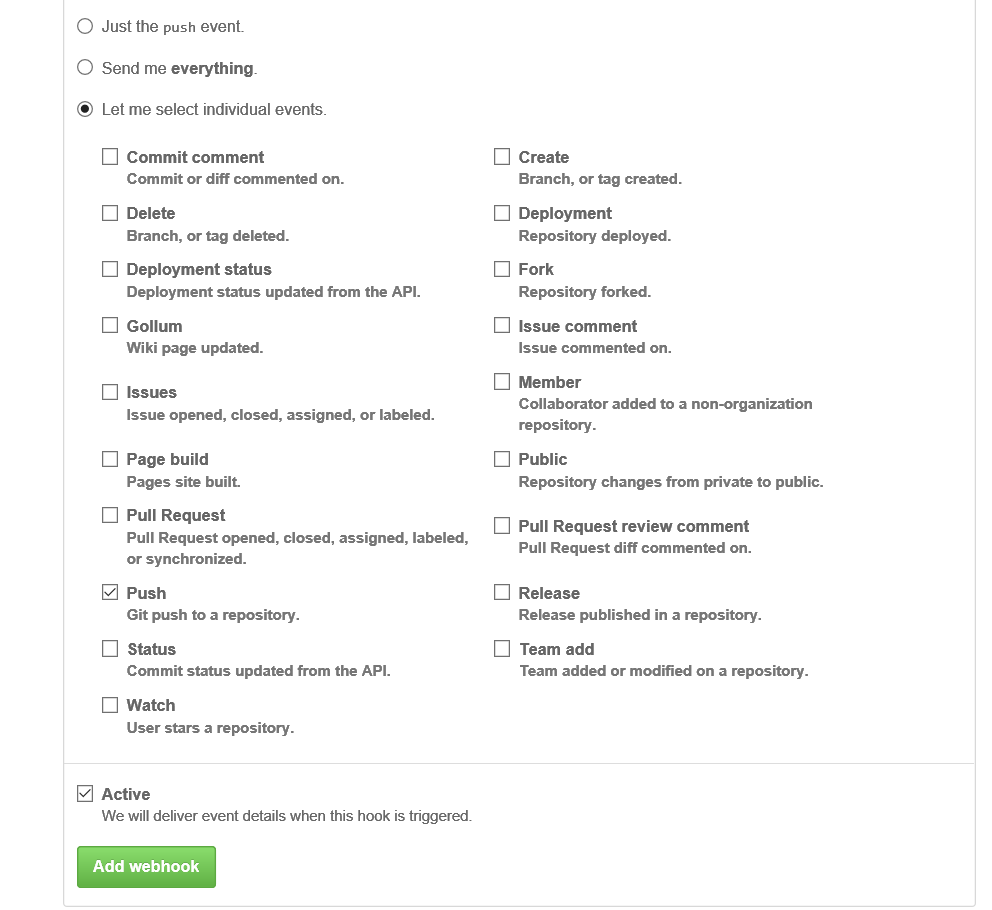
1. On the GitHub repository web page click Settings.
2. From the right menu select Webhooks & services.
3. Click on Add webhook.



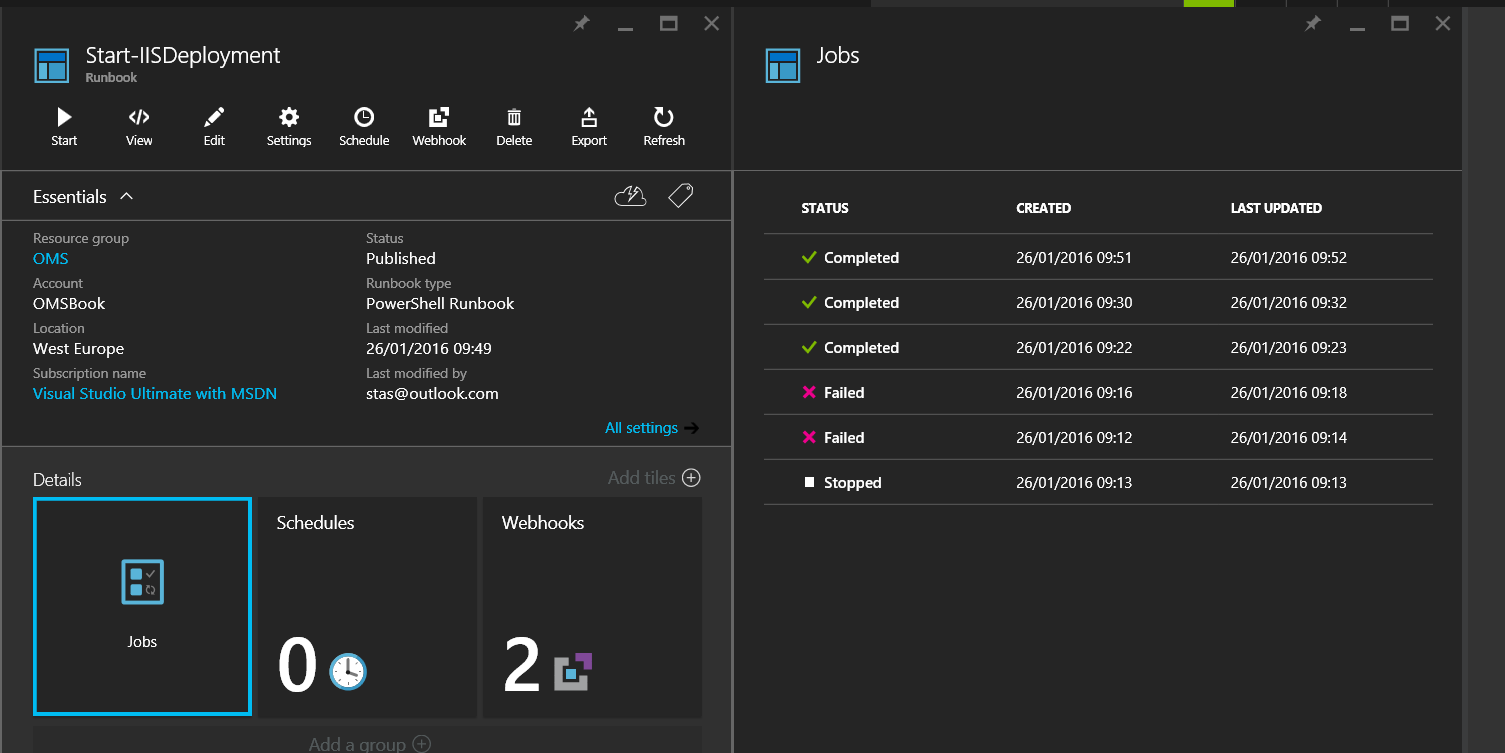
1. In Payload URL paste the URL for the webhook created in Azure Automation from step 9.
2. For Content type choose application/json.
3. Leave Secret field as it is.



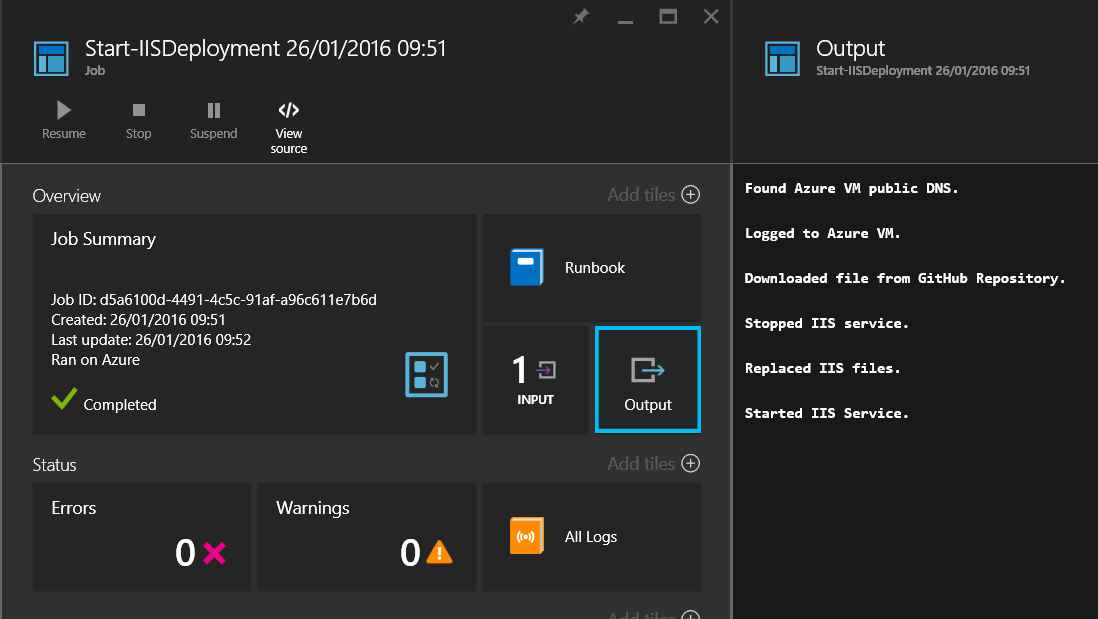
1. For triggering events choose Let me select individual events.
2. From all choice select only Commit comment.



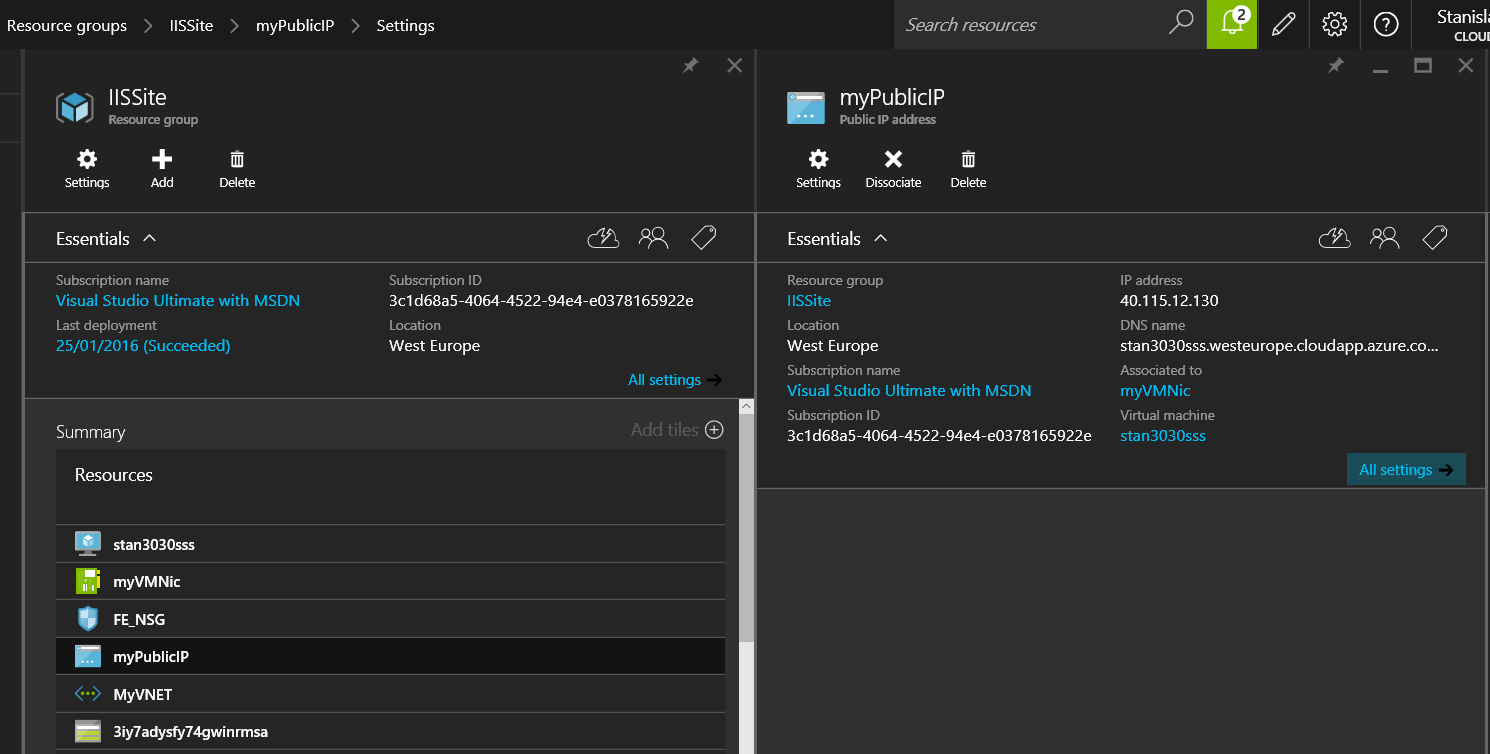
1. Check Active option.
2. Click on Add webhook. This will trigger the runbook in Azure Automation for first time.
3. Navigate to <portal.azure.com>.
4. From the hamburger menu on the right click on Automation Accounts.
5. Select the name of your Automation account.
6. From Automation account blade click on Runbooks.
7. From the listed runbooks click on Start-IISDeployment runbook.
8. Click on Jobs tile.



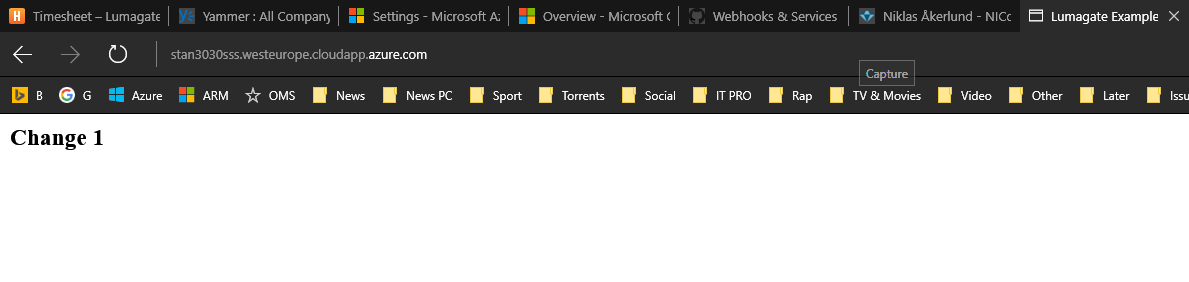
1. Select the first job which is probably in running or queued state.
2. Clicking on Output of the job you can see more details of the status.



1. When job is completed that means it passed successfully.
2. In Azure portal Browse IISSite resource group -> myPublicIP Public IP Address. Copy the value from DNS Name and paste in address bar of a browser.



1. When you open the URL in browser you will see default IIS website page changed.



1. Try editing the iisstart.htm file on GitHub and commitng the change to see another deployment being triggered.

**Summary**

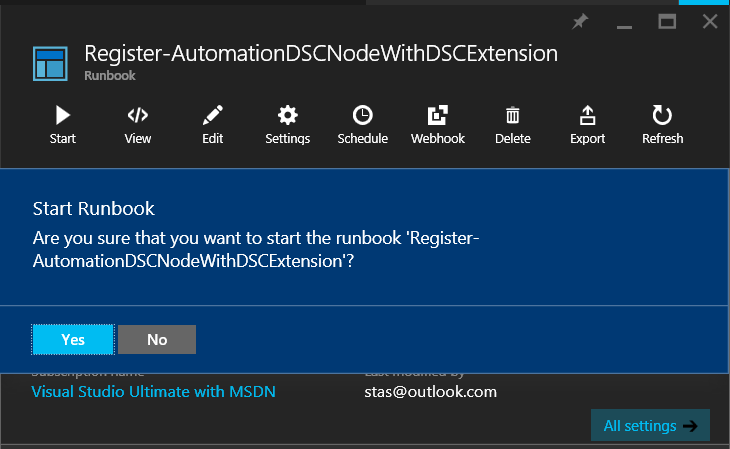
You have now successfully completed Lab 3, where you have done the following:

* Created webhook for Runbook in Azure Automation
* Configured webhook in GitHub
* Initiated a deployment to web site with runbook by just commiting change to master repository

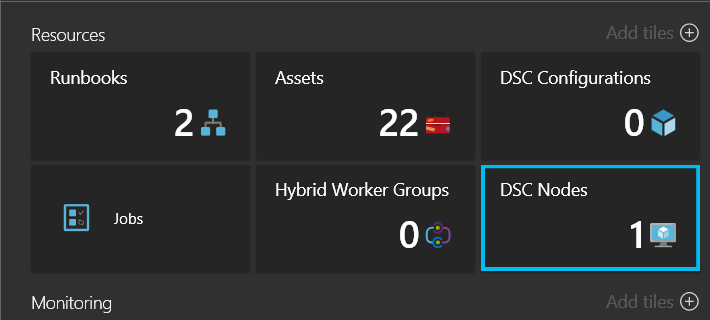
1. Lab 4: Deploy Web Site with Automation DSC

In this lab, you will deploy web site package with Automation DSC.

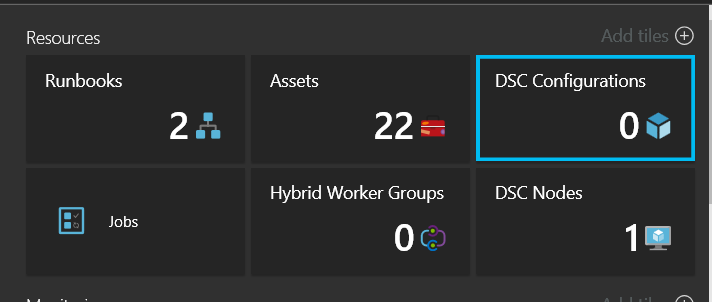
1. From your computer, open your preferred web browser and navigate to <portal.azure.com>.
2. From the hamburger menu on the right click on Automation Accounts.
3. Select the name of your Automation account.
4. From Automation account blade click on Runbooks.
5. Click on Register-AutomationDSCNodeWithDSCExtension runbook.
6. Start Register-AutomationDSCNodeWithDSCExtension runbook.



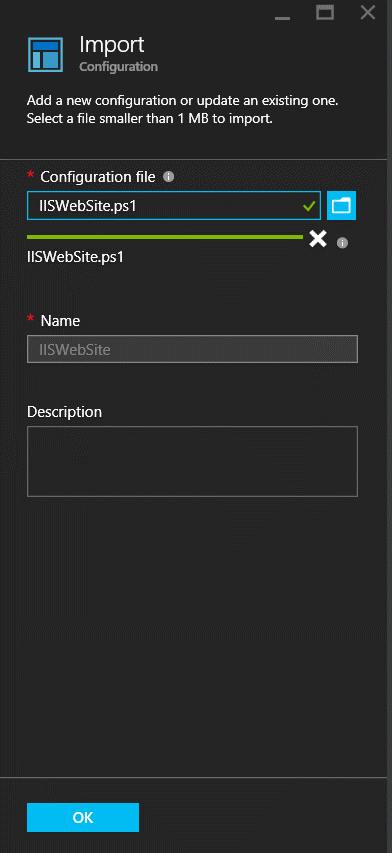
1. Wait until the job is completed successfully.
2. This runbook installs PowerShell v5 and connects the VM to Automation DSC.
3. When job is completed successful go to Automation account blade click on DSC Nodes.



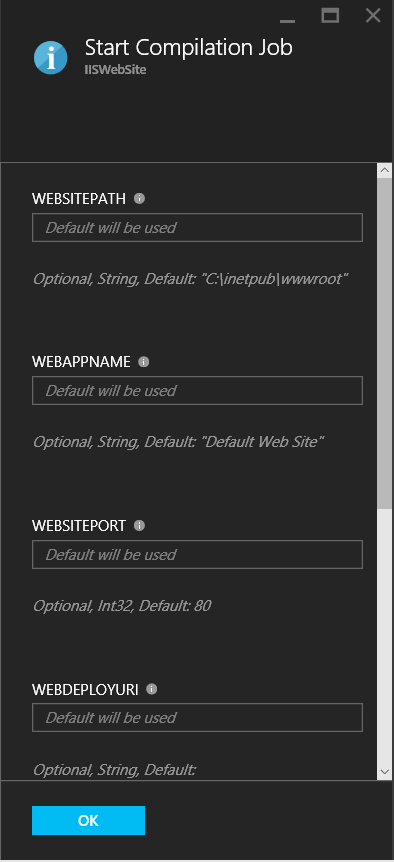
1. One Node should be registered successfully.
2. From Automation account blade click on DSC Configurations.



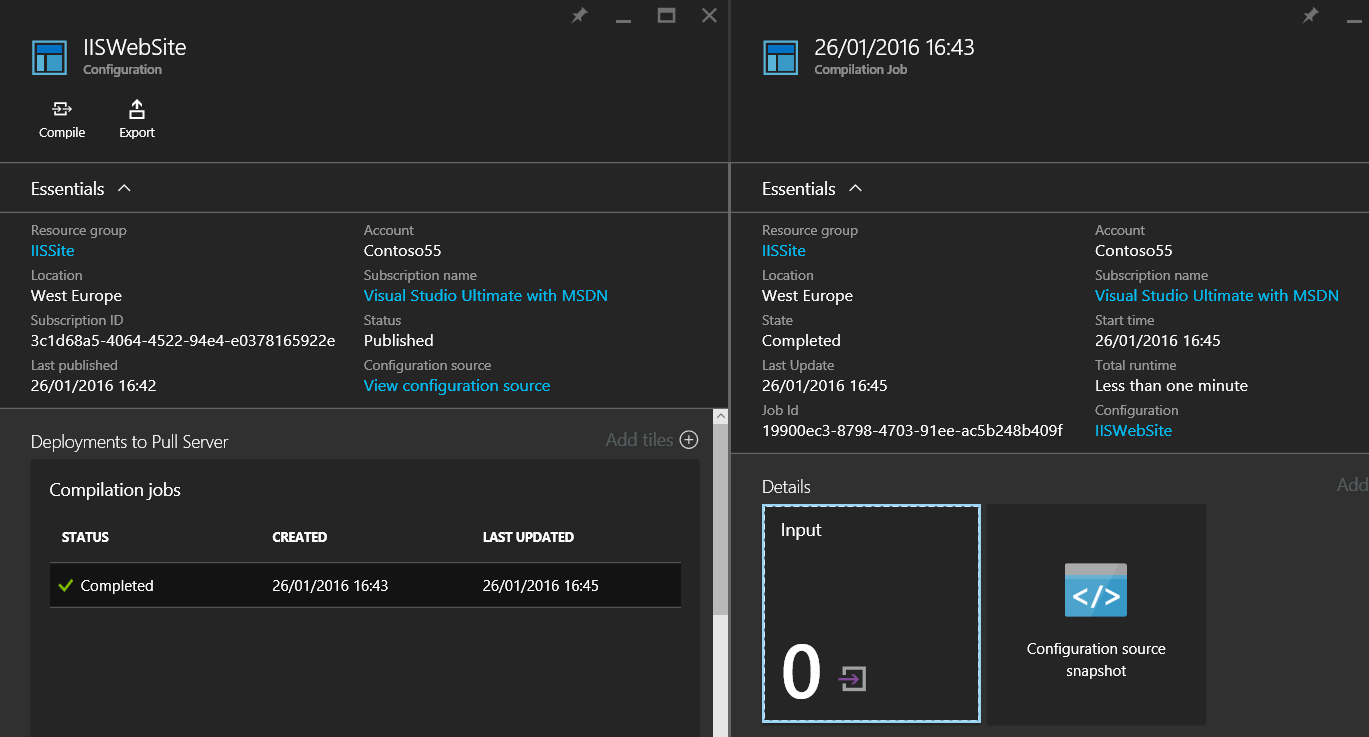
1. Click Add a configuration.
2. Import blade select configuration IISWebSite.ps1 from 04 - Azure Automation and Continious Delivery\Configurations and click OK.



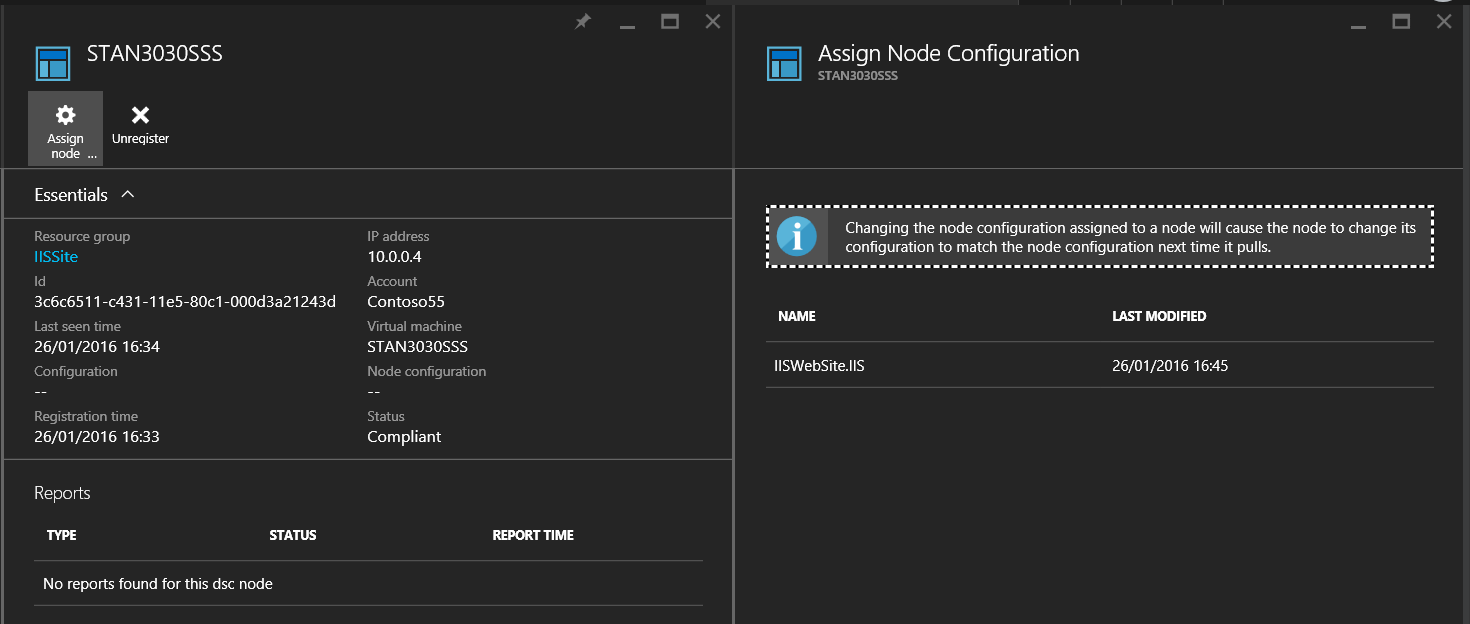
1. Once imported select the configuration and from the top menu choose Compile. When prompted to enter values do not enter them and click OK.



1. Wait until the compilation job completes successfully.

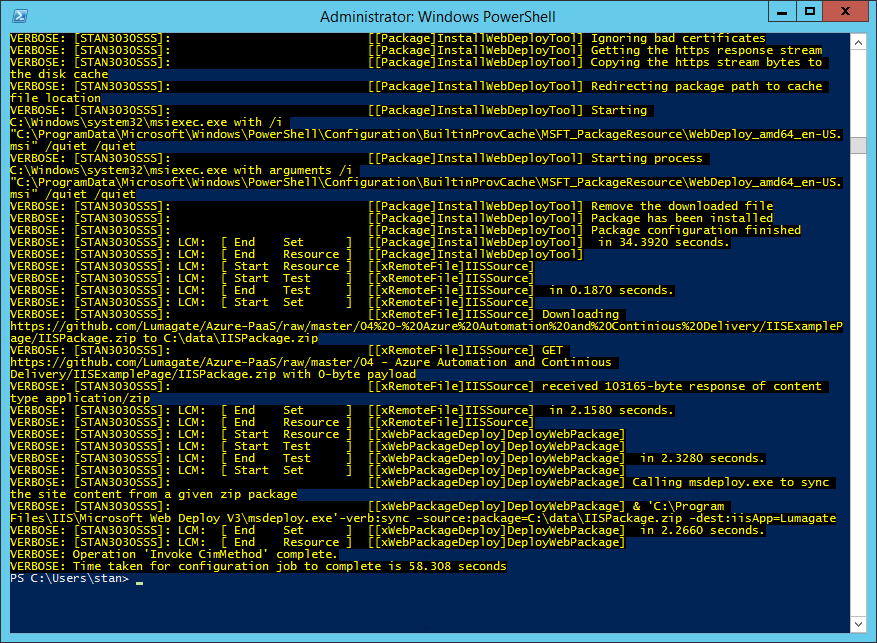


1. Navigate to Automation Account blade and click on DSC Nodes tile.
2. Select the node that was added earlier.
3. From the nodes blade click Assign node configuration. Select the compiled configuration and click OK

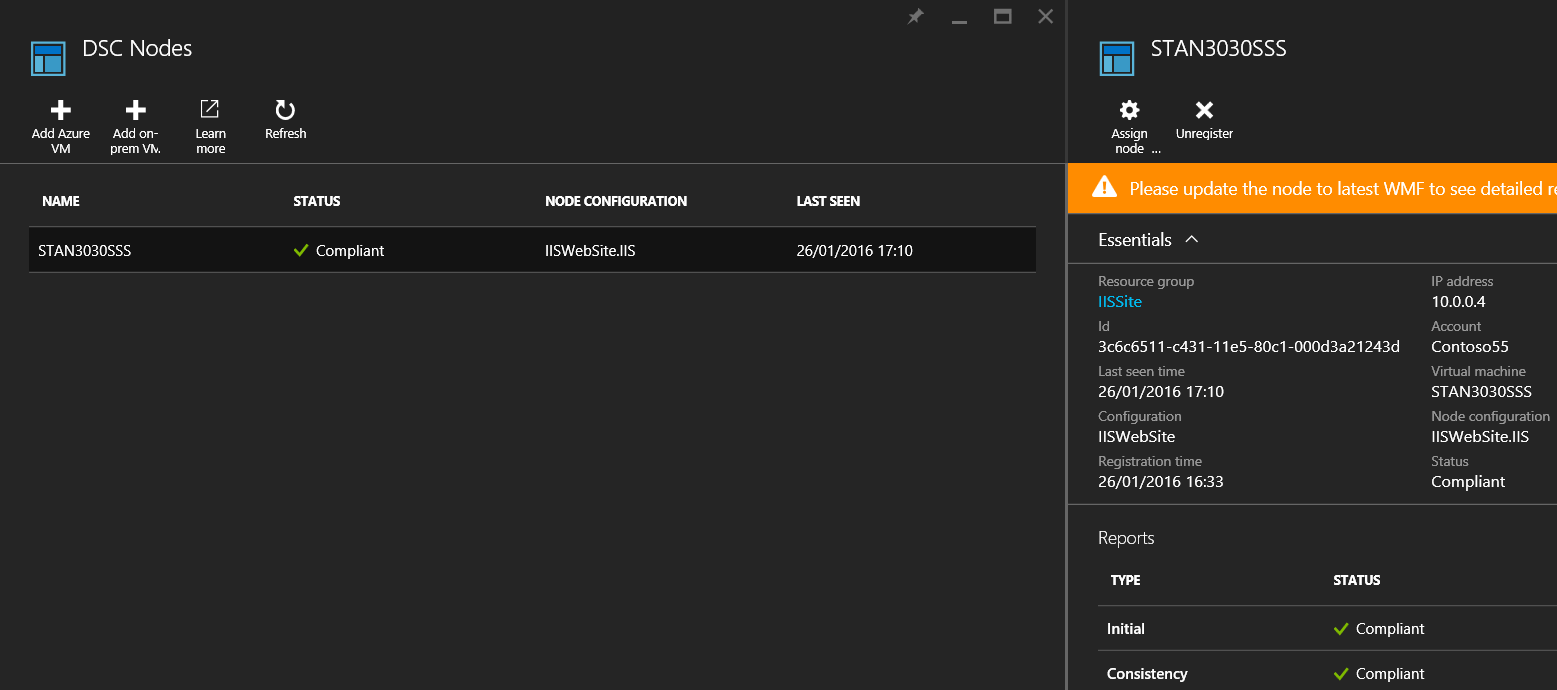


1. To trigger imidiate compliance logon to the server via PowerShell or RDP and in PowerShell Administrator console execute:

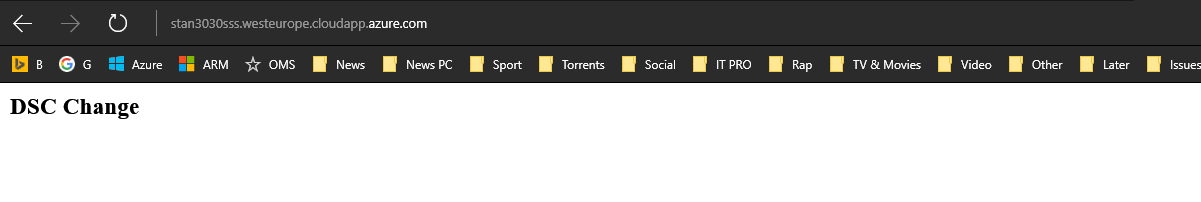
Update-DscConfiguration -Wait -Verbose



1. Once configuration is completed node should be compliant in Azure Automation.



1. Opening the Public DNS name of the VM will also show updated site.



**Summary**

You have now successfully completed Lab 4, where you have done the following:

* Added server to Automation DSC
* Added Configuration to Automation DSC
* Compiled Configuration to Node Configuration
* Assigned Node configuration to Node
* Deployed web site trough DSC Configuration

If the customer does not have a properly configured VPN or Azure Subscription in place, please have them consider the Azure IaaS Foundations IP at http://aka.ms/MCS\_EPG\_Azure\_Iaas-Foundation.

These are just examples; update the requirements adapting to the specific customer situation if necessary. Points to consider might include: tool usage (MAP versus customer provided), server locations, availability of test lab, and so on.