# Lab Answer Key: Module 3: Implementing virtual machines

# Lab A: Creating IaaS v2 virtual machines in Azure

## Exercise 1: Creating virtual machines by using the Azure portal and Azure PowerShell

#### Task 1: Use the Azure portal to create a virtual machine

1. On MIA-CL1, in Internet Explorer, in the address bar, type **https://portal.azure.com**, and then press Enter.
2. Sign in using the Microsoft account that is either the Service Administrator or Co-administrator of your subscription.
3. On the Hub menu, click **Virtual machines**, and then, on the Virtual machines blade, click **Add**.
4. On the **Compute** blade, click **Windows Server**.
5. On the Windows Server blade, click **Windows Server 2012 R2 Datacenter,** and then on the Windows Server 2012 R2 Datacenter blade, ensure that **Resource Manager** appears in the **Select a deployment model** list box, and then click **Create**.
6. On the Basics blade, specify the following:

* **Name:** ResDevDB1
* **VM disk type:** HDD
* **User name:** Student
* **Password:** Pa$$w0rd1234
* **Confirm password:** Pa$$w0rd1234
* **Subscription:** *Your subscription*

1. In the **Resource group** section, click **Use existing** and then, in the drop-down list and click **ResDevRG**.
2. Accept the default Location value and click **OK**.
3. On the Choose a size blade, click **A1 Standard** and then click **Select**.
4. On the Settings blade, ensure that **HQ-VNET** is selected as the **Virtual network**.
5. Click **Subnet**.
6. On the Choose subnet blade, click **Database**.
7. On the Settings blade, click **OK**
8. On the Summary blade, click **OK**.

**Note:** You can monitor the virtual machine's deployment progress on the Dashboard page.

#### Task 2: Use Azure PowerShell to create a virtual machine

1. On MIA-CL1, click the **Start** button, type **ISE**, and then click **Windows PowerShell ISE**.
2. In the Windows PowerShell Integrated Scripting Environment (ISE) window, open the **CreateRmVM.ps1**script at \*\*D:03\*\*
3. In the Windows PowerShell ISE window, review the content of the script.
4. In the Windows PowerShell ISE, click the **Run Script** icon or press F5.
5. In the Sign into your account window, type the name and password of an account that is either the Service Administrator or Co-administrator of your Azure subscription, and then click **Sign in**.
6. If you have multiple subscriptions, select the one to use in the labs in this module.
7. When the script is complete, leave the Windows PowerShell ISE window open.

**Result**: After completing this exercise, you will have created virtual machines by using the Azure portal and Azure PowerShell.

## Exercise 2: Validating virtual-machine creation

#### Task 1: Use Azure PowerShell to validate virtual-machine deployment

1. In the Windows PowerShell ISE window, at the command prompt, type the following command, and then press Enter:

Get-AzureRmResource | where {$\_.ResourceType -like "\*VirtualMachines"}

1. Confirm that the ResDevDB1 and the ResDevDB2 virtual machines are listed. Note that both virtual machines belong to the ResDevRG resource group.

#### Task 2: Use the Azure portal to validate virtual-machine deployment

1. On MIA-CL1, switch to Internet Explorer.
2. In the Internet Explorer window, in the Azure portal, on the Hub menu, click **All resources**. > **Note:** Note that both ResDevDB1 and ResDevDB2 are listed, along with the network and storage resources that you created in the previous exercise.
3. On the Hub menu, click **Virtual machines**.
4. On the Virtual machines blade, click **ResDevDB1**.
5. On the ResDevDB1 blade, confirm the following values:

* **Resource group:** **ResDevRG**
* **Virtual network/subnet: HQ-VNET/Database**

1. Repeat steps 4 and 5 for the **ResDevDB2** virtual machine.

**Result**: After completing this exercise, you will have validated the creation and configuration of Azure Virtual Machines.

# Lab B: Deploying IaaS v2 virtual machines by using Azure Resource Manager templates

## Exercise 1: Using Visual Studio and an Azure Resource Manager template to deploy IaaS v2 virtual machines

#### Task 1: Use Visual Studio to deploy the Linux app server's virtual machines

1. On MIA-CL1, in the Windows PowerShell ISE window, type in the following and press Enter (replace [Azure\_region] with the name of the Azure region that you selected when running the Setup-Azure script at the beginning of this module):

foreach ($sku in (Get-AzureRmVMImageSku -Location "[Azure\_region]" -PublisherName "Canonical" -Offer "UbuntuServer")) {if ((Get-AzureRmVMImage -Location "[Azure\_region]" -PublisherName "Canonical" -Offer "UbuntuServer" -Skus $sku.Skus -ErrorAction SilentlyContinue).Count -gt 0) {Write-Host $($sku).Skus}}

1. In the output displayed in the Windows PowerShell ISE window, note the list of the Ubuntu images avaialble in the Azure region you selected.
2. On MIA-CL1, on the taskbar, click the **Visual Studio** icon.
3. In Visual Studio, click **File**, click **Open**, click **Project/Solution**, and then browse to \*\*D:03\*.
4. In the Open Project window, click **ResDevLinuxDeploy.sln**, and then click **Open**.
5. In Visual Studio, in the Solution Explorer pane, expand **Templates**, and then click **azuredeploy.json**.
6. View the parameters and variables section of the template. In the list of parameters in the JSON Outline window, click **ubuntuOSVersion**.
7. In the template, identify the values of the **defaultValue** and **allowedValues** entries of the **ubuntuOSVersion** parameter. If these values do not include one of the entries you noted in step 2, perform the following tasks:

* identify the last entry from the list you noted in step 2 and replace the **defaultValue** entry of **"14.04.2-LTS"** in the template with that entry (in double quotes). For example, if the entry you noted was **16.04.0-LTS**, then you would replace **"defaultValue": "14.04.2-LTS",** with **"defaultValue": "16.04.0-LTS",**.
* identify the last entry from the list you noted in step 2 and replace the **allowedValues** entry of **"14.04.2-LTS"** in the template with that entry (in double quotes). For example, if the entry you noted was **16.04.0-LTS**, then you would replace the line **"14.04.2-LTS",** under **"allowedValues": [** with **"16.04.0-LTS",**.

1. In the Solution Explorer pane, right-click **ResDevLinuxDeploy**, click **Deploy**, and then click **New Deployment**.
2. If necessary, in the Deploy to Resource Group window, click **Add an account**. In the Sign in to your account window, sign in with an account that is either the Service Administrator or Co-administrator of your Azure subscription.
3. In the Deploy to Resource Group window, click the **Resource Group** drop-down box, and then click **ResDevRG**.
4. In the Deploy to Resource Group window, click **Edit Parameters**.
5. In the Edit Parameters window, populate the parameter values according to details in the following list:

* **vmName:** ResDevApp1
* **adminUsername:** Student
* **adminPassword:** Pa$$w0rd
* **virtualNetworkName:** HQ-VNET
* **resourceGroupName**: ResDevRG
* **subnetName:** App
* **vmSize:** Standard\_D1\_V2
* **ubuntuOSVersion:** 14.04.2-LTS
* **storageAccountType:** Standard\_LRS

1. In the Edit Parameters window, click the **Save passwords** check box, and then click **Save**.
2. In the Deploy to Resource Group window, click **Deploy**. > **Note:** Deployment will run with the output that appears in the Output pane, which is at the bottom of the window. When deployment is complete, you will receive a message stating the template was deployed successfully to resource group ResDevRG.
3. In the Solution Explorer pane, click **Azuredeploy.parameters.json**.In the main window pane, notice that the parameters that you entered in the first deployment are saved in this file. You can reuse these parameters for the deployment of the second app server.
4. In the Solution Explorer pane, right-click **ResDevLinuxDeploy**, click **Deploy**, and then click **ResDevRG**.
5. In the Deploy to Resource Group window, click **Edit Parameters**.
6. In the Edit Parameters window, in the **vmNameValue** box, type **ResDevApp2**, and then click **Save**.
7. In the Deploy to Resource Group window, click **Deploy**. > **Note:** Deployment will run with the output that appears in the Output pane, which is at the bottom of the window. When deployment is complete, you will receive a message stating the template was deployed successfully to resource group ResDevRG.
8. In the Visual Studio, click **File** and then on the drop-down menu, click **Close Solution**.
9. Leave Visual Studio open.

#### Task 2: Use Azure PowerShell to validate the deployment of the app server's virtual machines

1. On MIA-CL1, on the taskbar, right-click the **Windows PowerShell** icon and select **Run ISE as Administrator** from the pop-up menu.
2. In the Windows PowerShell ISE, at the command prompt, type the following cmdlet, and then press Enter:

Login-AzureRMAccount

1. When prompted, sign in to your Azure subscription with an account that is either the Service Administrator or Co-administrator of your Azure subscription.
2. If you have multiple subscriptions associated with your account, at the Windows PowerShell ISE prompt, type the following cmdlet, and then press Enter:

Get-AzureRmSubscription

1. Identify the name of the Azure subscription to which you deployed virtual machines in the previous task of this exercise, type in the following cmdlet, and then press Enter (replace *'Name of your subscription'* with the actual name of your subscription and make sure to enclose the name of your subscription in single quotes):

Set-AzureRmContext -SubscriptionName 'Name of your subscription'

1. Type the following cmdlet, and then press Enter:

Find-AzureRMResource -ResourceGroupNameContains ResDevRG | Format-Table -Property ResourceName, ResourceType

1. In the cmdlet output, note the resources created in this exercise: **ResDevApp1** and **ResDevApp2** virtual machines, and an NIC, public IP, and storage account for each virtual machine.
2. Leave the Windows PowerShell ISE window open for the next exercise.

**Result**: After completing this exercise, you will have deployed Azure Virtual Machines by using Visual Studio and an Azure Resource Manager template.

## Exercise 2: Using Azure PowerShell and an Azure Resource Manager template to deploy virtual machines

#### Task 1: Use Azure PowerShell to deploy the Windows virtual machines

1. In the Windows PowerShell ISE window that you opened in the previous exercise, click **File**, and then click **Open**.
2. In the **Open** dialog box, navigate to the \*\*D:03\* folder.
3. Click **ResDevWindowsDeploy.ps1**, and then click **Open**.
4. In the Windows PowerShell ISE window, review the script that will initiate the template. > **Note:** Note the $templateFile and $rgName variables. These represent the location of the Azure Resource Manager template file and the resource group to which you will deploy the virtual machines.
5. Switch to Visual Studio and click **File**, click **Open**, and then click **File**.
6. In the **Open File** dialog box, navigate to the \*\*D:03\* folder.
7. Click **ResDevWindowsDeployTemplate.json**, and then click **Open**. > **Note:** Note that the template has the same structure as the template for the Linux virtual machines in the previous exercise. The only difference between the two templates is the variables declaring the image and operating system details.
8. Close Visual Studio.
9. Switch back to the Windows PowerShell ISE window and run the **ResDevWindowsDeploy.ps1** script. When prompted, provide the following values for the parameter prompts, pressing Enter after each value:

* **vmName:** **ResDevWeb1**
* **adminUsername:** **Student**
* **adminPassword:** **Pa$$w0rd**
* **virtualNetworkName:** **HQ-VNET**
* **subnetName:** **Web**

1. When the script completes, repeat step 9, changing only the value of the vmName parameter to **ResDevWeb2**

#### Task 2: Use the Azure portal to validate deployment of the Windows virtual machines

1. In Internet Explorer, on the address bar, type **https://portal.azure.com**, and then press Enter.
2. Sign in using the Microsoft account that is either the Service Administrator or Co-administrator of your subscription.
3. On the Hub menu, click **Resource groups**.
4. On the Resource groups blade, click **ResDevRG**.
5. On the ResDevRG blade, in the Summary section, view the **Resource** list. Click the ellipsis (...) in the lower-right corner to see all of the resources in this resource group. > **Note:** Note the virtual machines and the NIC and public IP resources for each virtual machine.
6. On the Resources blade, click **ResDevWeb1**.
7. On the ResDevWeb1 blade, in the Essentials section, note that ResDevWeb1 has been assigned to the **HQ-VNet/Web** virtual network/subnet, and the operating system is **Windows**.
8. Close Internet Explorer.

#### Task 3: Reset the environment

1. Close all open applications without saving any files.
2. On the taskbar, right-click **Windows PowerShell**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**.
4. At the command prompt, type the following command, and then press Enter:

Reset-Azure

1. When prompted, sign in by using the Microsoft account associated with your Azure subscription.
2. If you have multiple Azure subscriptions, select the one you want to target with the script.
3. When prompted for confirmation, type **y**.

**Note:** This script might remove Azure services in your subscription. Therefore, we recommend that you use an Azure trial pass that was provisioned specifically for this course, and not your own Azure account. The script will take 5-10 minutes to reset your Microsoft Azure environment, and prepare it for the next lab. The script removes all storage, VMs, virtual networks and gateways, cloud services, and resource groups. **Important**: The script might not be able to get exclusive access to a storage account to delete it (you will see an error, if this occurs). If you still find objects after the reset script is complete, you can rerun the **Reset-Azure** script, or use the full Azure portal to manually delete all the objects in your Azure subscription, with the exception of the default directory.

**Result**: After completing this exercise, you will have deployed Azure Virtual Machines by using Windows PowerShell and a Resource Manager template.

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/legalcode), additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.