# Module 3: Implementing virtual machines

# Lab A: Creating IaaS v2 virtual machines in Azure

### Scenario

As part of the planning for deployment of IaaS v2 virtual machines to Azure, A. Datum has evaluated its deployment options. You must use the Azure portal and Azure PowerShell to deploy two Windows virtual machines for the Database tier of the Research and Development application. Additionally, to facilitate resource tracking, you should ensure that the virtual machines are part of the same resource group.

### Objectives

After completing this lab, you will be able to:

* Create virtual machines by using the Azure portal and Azure PowerShell.
* Validate virtual-machine creation.

### Lab Setup

Estimated Time: 35 minutes

**Virtual machine**: **20533C-MIA-CL1**

**User** **name**: \*\* Student\*\*

**Password**: \*\* Pa$$w0rd\*\*

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

### Scenario

You must deploy two virtual machines that are running Windows Server 2012 R2 Datacenter. Name these machines ResDevDB1 and ResDevDB2, and use them as database servers for the Research and Development app, ResDev. You will use the Azure portal to deploy one of the virtual machines, and you will use Azure PowerShell to deploy the other VM. You must deploy both virtual machines into the ResDevRG resource group, and you must configure the virtual machines to use the database subnet of the HQ-VNET virtual network. After deploying the virtual machines, you will confirm that the virtual machines are deployed to the correct resource group and are on the database subnet.

The main tasks for this exercise are as follows:

1. Use the Azure portal to create a virtual machine
2. Use Azure PowerShell to create a virtual machine

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

1. **On** **MIA-CL1,** **in Internet Explorer, navigate to the Azure portal.**
2. **Sign in using the Microsoft account that is either the Service Administrator or Co-administrator of your subscription.**
3. **From the Azure portal, create an IaaS v2 virtual machine with the following parameters:**

* **Name:** ResDevDB1
* **User name:** Student
* **Password:** Pa$$w0rd1234
* **Subscription:***Your subscription*
* **Resource group:** ResDevRG
* **Location:***accept the default location, which should match the location of the resource group.*
* **Size:** **A1 Standard**
* **Network** **name:** **HQ-VNET**
* **Subnet name:** **Database**

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

1. On MIA-CL1, open a Windows PowerShell Integrated Scripting Environment (ISE) window. In the Windows PowerShell ISE, open the script **D:03.ps1** **and review its content.**
2. Run the script.
3. When prompted to sign in, type the name of the account that is either the Service Administrator or Co-administrator of your Azure subscription.
4. If you have multiple subscriptions, select the one to use in the labs in this module.
5. 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

### Scenario

You now must validate the creation and configuration of the virtual machines that you created to ensure that they function properly.

The main tasks for this exercise are as follows:

1. Use Azure PowerShell to validate virtual-machine deployment
2. Use the Azure portal to validate virtual-machine deployment

#### 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, in the Internet Explorer window, in the Azure portal, view the virtual machine resources.
2. Confirm that both ResDevDB1 and ResDevDB2 have been created, that they belong to the **ResDevRG** resource group, and that they reside on the **Database** subnet of the **HQ-VNET** virtual network.

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

**Question** What storage-related differences did you notice when you created a virtual machine in the Azure portal versus in Azure PowerShell?

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

### Scenario

You must use an Azure Resource Manager template to deploy two additional Linux virtual machines and two additional Windows virtual machines that the ResDev application will use. The virtual machines should be part of the ResDevRG resource group, to facilitate resource tracking. Linux virtual machines should reside on the app subnet of the HQ-VNET virtual network, and Windows virtual machines should reside on the web subnet of the HQ-VNET virtual network.

### Objectives

After completing this lab, you must be able to:

* Use Visual Studio and an Azure Resource Manager template to deploy IaaS v2 virtual machines.
* Use Azure PowerShell and an Azure Resource Manager template to deploy virtual machines.

### Lab Setup

Estimated Time: 25 minutes

**Virtual machine**: **20533C-MIA-CL1**

**User name**: \*\* Student\*\*

**Password**: \*\* Pa$$w0rd1234\*\*

The virtual machine should be running from the previous lab.

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

### Scenario

You must use Visual Studio to deploy two Linux IaaS v2 virtual machines for use as app servers in the ResDev app. You should name the servers ResDevApp1 and ResDevApp2. You have a deployment-template solution and the deployment details for both virtual machines. You must deploy the two virtual machines from Visual Studio, and then confirm that the virtual machines have been deployed successfully by using Azure PowerShell.

The main tasks for this exercise are as follows:

1. Use Visual Studio to deploy the Linux app server's virtual machines
2. Use Azure PowerShell to validate the deployment of the app server's 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, open the Visual Studio solution located at **D:03.**
3. View the contents of the **azuredeploy.json** template. In the list of parameters in the JSON Outline window, click **ubuntuOSVersion**.
4. If necessary, update the values of **defaultValue** and **allowedValues** entries of the **ubuntuOSVersion** parameter with the values you identified in step 2.
5. In the Solution Explorer, right-click **ResDevLinuxDeploy**, and then start a new deployment process for the first virtual machine.
6. Deploy a new virtual machine into the ResDevRG resource group, by using the following parameter values:

* **vmName:** ResDevApp1
* **adminUsername:** Student
* **adminPassword:** Pa$$w0rd1234
* **virtualNetworkName:** HQ-VNET
* **resourceGroupName**: ResDevRG
* **subnetName:** App
* **vmSize:** Standard\_D1\_v2
* **ubuntuOSVersion:** 14.04.2-LTS
* **storageAccountType:** Standard\_LRS
* **Save password**: enabled

**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 that the template was deployed successfully to resource group ResDevRG.

1. View the contents of the Azuredeploy.parameters.json file to see that the parameters that you entered have been saved in this file.
2. Start another deployment process by using the deployment that you used for the first virtual machine.
3. Repeat step 4, changing only the following parameter:

* **vmName:** ResDevApp2

**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.

1. Close the solution but leave Visual Studio open.

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

1. On MIA-CL1, launch Windows PowerShell ISE as Administrator.
2. With the Service Administrator or Co-administrator credentials, sign in to the subscription to which you deployed virtual machines in the previous task of this exercise by using the following cmdlet:

Login-AzureRMAccount

1. 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. In the Windows PowerShell ISE, at the command prompt, 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

### Scenario

You must deploy the Web tier virtual machines by using an Azure Resource Manager template and the Azure portal. The Web tier should consist of two virtual machines that are running Windows Server 2012 R2, ResDevWeb1 and ResDevWeb2. You should deploy them to the ResDevRG resource group, and then host them on the web subnet of the HQ-VNet virtual network. You have a template and a Windows PowerShell script that you should edit to complete the deployment. After you deploy the virtual machines, confirm the deployment by viewing the newly deployed resources in the Azure portal.

The main tasks for this exercise are as follows:

1. Use Azure PowerShell to deploy the Windows virtual machines
2. Use the Azure portal to validate deployment of the Windows virtual machines
3. Reset the environment

#### 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, open **D:03 ResDevWindowsDeploy.ps1**
2. 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.
3. Switch to Visual Studio and open the file **D:03\***ResDevWindowsDeployTemplate.json**. >** 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.
4. Close Visual Studio.
5. Switch back to the Windows PowerShell ISE window and run the **ResDevWindowsDeploy.ps1** script. When prompted, provide the following values:

* **vmName:** **ResDevWeb1**
* **adminUsername:** **Student**
* **adminPassword:** **Pa$$w0rd1234**
* **virtualNetworkName:** **HQ-VNET**
* **subnetName:** **Web**

1. When the script completes, repeat step 5, 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, browse to the Azure portal.
2. Sign in using the Microsoft account that is either the Service Administrator or Co-administrator of your subscription.
3. Navigate to the blade of the **ResDevRG** resource group.
4. On the ResDevRG resource group blade, view the full list of its resources. > **Note:** Note the virtual machines, and the NIC and public IP resources for each virtual machine.
5. View the details for the ResDevWeb1 virtual machine. On the ResDevWeb1 blade, note that ResDevWeb1 has been assigned to the **HQ-VNet/Web** virtual network/subnet, and the operating system is **Windows**.
6. 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**. In the User Account Control dialog, click **Yes**.
3. 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.

**Question** Can Visual Studio and Windows PowerShell use the same Azure Resource Manager template to deploy a virtual machine?

**Question** How would you configure an Azure Resource Manager template to deploy multiple virtual machines with different configurations?

©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.