# Module 4: Managing virtual machines

# Lab: Managing Azure virtual machines

### Scenario

Now that you have validated basic deployment options of Azure VMs, you need to start testing more advanced configuration scenarios. Your plan is to step through a sample configuration of a two-tier A. Datum ResDev application. As part of your tests, you will set up the front-end tier consisting of a pair of load balanced Azure VMs hosting IIS configured by using the VM DSC extension. You will also set up a multi-disk volume by using Storage Spaces in a Windows Azure VM in the back-end tier.

### Objectives

After completing this lab, you will be able to:

* Set up load balancing for a pair of Azure VMs and configure their workload by using the VM DSC extension.
* Implement Storage Space–based simple volumes in Azure VMs.

### Lab Setup

Estimated Time: 60 minutes

Virtual Machine: **20533E-MIA-CL1**

User name: **Student**

Password: **Pa55w.rd**

## Exercise 1: Configuring front-end web tier

### Scenario

You need to test the ability of Azure VMs in the same availability set to operate in a load balanced configuration by leveraging Azure load balancer. You also need to test the implementation of the desired state configuration in Azure by using VM Agent DSC extension to install the default IIS website on two Azure VMs that will host the web tier of the A. Datum ResDev application. Once the installation is complete, you must test the availability of this setup by verifying that load balanced access to the default website is not affected by shutting down one of the Azure VMs.

The main tasks for this exercise are as follows:

1. Review the existing deployment
2. Implement an Azure Load Balancer
3. Install and configure IIS by using DSC and Windows PowerShell
4. Test the DSC configuration and virtual machine availability

#### Task 1: Review the existing deployment

1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa55w.rd**.
2. Start Microsoft Edge, browse to the Azure portal and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
3. In the Azure portal, navigate to the resource group **20533E0401-LabRG**.
4. On the 20533E0401-LabRG blade, review the list of resources. Note that includes an availability set named 20533E0401-avset.
5. Navigate to the **20533E0401-avset** blade and note that the availability set has 2 fault domains, 5 update domains, and it contains two virtual machines. Also note that each VM has a unique fault domain and update domain.
6. Leave the Microsoft Edge window with the Azure portal open.

#### Task 2: Implement an Azure Load Balancer

1. On MIA-CL1, from the Azure portal, create an Azure load balancer with the following settings:

* Name: **20533E0401-ilb**
* Type: **Public**
* SKU: **Basic**
* Public IP address: create a new IP address named **20533E0401-ilbfe** with dynamically assigned IP address
* Subscription: the name of your Azure subscription
* Resource group: **20533E0401-LabRG**
* Location: the same Azure region you chose when running the provisioning script at the beginning of this module

1. Configure the newly created load balancer with the backend pool named **20533E0401-ilb-bepool** and associate it to the availability set **20533E0401-avset** with **ipconfig1** of **20533E0401-vm0** and **ipconfig1** of **20533E0401-vm1**.
2. Configure the load balancer with the health probe that has the following settings:

* Name: **20533E0401-ilb-probetcp80**
* Protocol: **HTTP**
* Port: **80**
* Path: **/**
* Interval: **5**
* Unhealthy threshold: **2**

1. Configure the load balancer with the following load balancing rule:

* Name: **20533E0401-ilb-ruletcp80**
* IP Version: **IPv4**
* Frontend IP address: **LoadBalancerFrontEnd**
* Protocol: **TCP**
* Port: **80**
* Backend port: **80**
* Backend Pool: **20533E0401-ilb-bepool (2 virtual machines)**
* Probe: **20533E0401-ilbprobetcp80 (HTTP:80)**
* Session persistence: **None**
* Idle timeout: **4**
* Floating IP (direct server return): **Disabled**

1. Add to the load balancer with the following inbound NAT rule:

* Name: **20533E0401-ilb-natrulerdpvm0**
* Frontend IP address: **LoadBalancerFrontEnd**
* Service: **Custom**
* Protocol: **TCP**
* Port: **33890**
* Associated to: **20533E0401-avset (availability set)**
* Target virtual machine: **20533E0401-vm0**
* Network IP configuration: **ipconfig1**
* Port mapping: **Custom**
* Floating IP (direct server return): **Disabled**
* Target port: **3389**

1. Add to the load balancer with the following inbound NAT rule:

* Name: **20533E0401-ilb-natrulerdpvm1**
* Frontend IP address: **LoadBalancerFrontEnd**
* Service: **Custom**
* Protocol: **TCP**
* Port: **33891**
* Associated to: **20533E0401-avset (availability set)**
* Target virtual machine: **20533E0401-vm1**
* Network IP configuration: **ipconfig1**
* Port mapping: **Custom**
* Floating IP (direct server return): **Disabled**
* Target port: **3389**

**Note:** This configuration will allow you to connect to both Azure VMs via RDP even though they do not have directly assigned public IP address.

1. On the 20533E0401-ilb blade, review the **Overview** section and identify the public IP address assigned to the load balancer. Note that at this point, you will not be able to connect to the two virtual machines in the backend pool, because they are not running a web server and the connectivity is additionally restricted by default network security group settings and the operating system-level firewall. You will change these settings later in this lab.

#### Task 3: Install and configure IIS on Azure VMs by using DSC and Windows PowerShell

1. On MIA-CL1, start File Explorer and browse to the **F:\Labfiles\Lab04\Starter** folder.
2. In the **F:\Labfiles\Lab04\Starter** folder, right-click on the **IISInstall.ps1** file and select **Edit** from the right-click menu. This will open the file in the **Windows PowerShell ISE**.
3. Review the content of the file. Note that this is a DSC configuration that controls the installation of the Windows Server 2016 Web-Server role.
4. Close the Windows PowerShell ISE window.
5. In the File Explorer, right click on the **F:\Labfiles\Lab04\Starter\Deploy-20533E0401DSC.ps1** file and select **Edit** from the right-click menu. This will open the file in the **Windows PowerShell ISE** window with the current directory set to **F:\Labfiles\Lab04\Starter**.
6. Review the content of the script. Note the variables that it uses, including the storage account and its key. The script first retrieves the storage account from the resource group, and then publishes the DSC configuration defined in the **Install.ps1** into it, placing it in the default DSC container named **windows-powershell-dsc**, stores the resulting module URL in a variable, and then sets the Azure Agent VM DSC extension on two virtual machines deployed by the provisioning script by referencing that URL. The script generates a shared access signature token that provides read only access to the blob representing the DSC configuration archive.
7. Start the execution of the script. When prompted, sign in with the username and the password of an account that is either a Service Administrator or a Co-Admin of your Azure subscription. Wait until the script completes.
8. On MIA-CL1, open Internet Explorer and navigate to the Azure portal.
9. Initiate a Remote Desktop session to **20533E0401-vm0** from the Azure portal.
10. When prompted to enter credentials to connect, type **Student** as the user name and **Pa55w.rd1234** as the password.
11. Once you establish a Remote Desktop session to the VM, in the **Server Manager** window, verify that IIS appears in the left pane, indicating that the Web Server (IIS) server role is installed.
12. Repeat steps 9 through 11 for the other virtual machine, **20533E0401-vm1**.
13. After completing the tasks, switch back to your lab computer MIA-CL1. Leave both Remote Desktop sessions open.

#### Task 4: Test the DSC configuration and virtual machine availability

1. From the Azure portal within the Internet Explorer window on MIA-CL1, create a new inbound security rule for the **20533E0401-web-nsg** security group with the following settings:

* Source: **Any**
* Source port ranges: **Any**
* Destination: **Any**
* Destination port ranges: **80**
* Protocol: **TCP**
* Action: **Allow**
* Priority: **1100**
* Name: **allow-http**

1. From the Azure portal, identify the IP address of the **20533E0401-ilb** load balancer.
2. From MIA-CL1, open a new InPrivate Browsing Internet Explorer session and browse to this IP address.
3. Verify that you can access the default IIS webpage and close the InPrivate Browsing session.
4. From the Remote Desktop sessions to two Azure VMs, stop the **World Wide Web Publishing Service** service on both **20533E0401-vm0** and **20533E0401-vm1**
5. From MIA-CL1, open a new InPrivate Browsing Internet Explorer session.
6. In the new InPrivate Browsing window, delete browsing history.
7. Browse to the IP address of the **20533E0401-ilb** load balancer again and verify that you can no longer access the default IIS webpage.
8. From the Remote Desktop session window, start the **World Wide Web Publishing Service** service on **20533E0401-vm0**.
9. Once the service is running, switch back to MIA-CL1 and refresh the InPrivate Browsing Internet Explorer window. Verify that you can again access the default the default IIS webpage. Note that you might need to wait about a minute after you start the **World Wide Web Publishing Service** service.

**Note:** Optionally you can repeat this sequence, but this time stopping the **World Wide Web Publishing Service** on **20533E0401-vm0** and starting it on **20533E0401-vm1**. As long as the service is running on at least one of the two virtual machines, you should be able to access the webpage.

**Result**: After completing this exercise, you should have created and configured a load balancer in front of two Azure VMs in the same availability set and implemented DSC-based configuration on these VMs.

## Exercise 2: Implementing Storage Spaces–based volumes

### Scenario

To test provisioning of multi-disk volumes on Azure VMs, you want to create three new VM disks, attach them to the Azure VMs that will host the database tier of the A. Datum ResDev application, and then use Storage Spaces to create a new volume.

The main tasks for this exercise are as follows:

1. Attach VHDs to an Azure VM
2. Configure a Storage Spaces simple volume
3. Remove the lab environment.

#### Task 1: Attach VHDs to an Azure VM

1. On MIA-CL1, from the Azure portal in the Internet Explorer window, attach to the 20533E0401-vm2 virtual machine a managed data disks with the following settings:

* Name: **20533E0401-vm2-data01**
* Resource group: Select **20533E0401-LabRG** in the drop down list marked "Select Existing".
* Account type: **Standard\_SSD**
* Source type: **None (empty disk)**
* Size: **128**
* HOST CACHING: **None**

1. On MIA-CL1, from the Azure portal in the Internet Explorer window, attach to the 20533E0401-vm2 virtual machine a managed data disks with the following settings:

* Name: **20533E0401-vm2-data02**
* Resource group: Select **20533E0401-LabRG** in the drop down list marked "Select Existing".
* Account type: **Standard\_SSD**
* Source type: **None (empty disk)**
* Size: **128**
* HOST CACHING: **None**

#### Task 2: Configure a Storage Spaces simple volume

1. On MIA-CL1, switch to the Remote Desktop session to 20533E0401-vm2.
2. While connected to 20533E0401-vm2, from the Server Manager window, create a storage pool named **StoragePool1** consisting of two newly attached disks.
3. From the Server Manager window, create a new virtual disk named **VirtualDisk1** using **StoragePool1** with the **Simple** storage layout, the **Fixed** provisioning type, and the maximum size.
4. From the Server Manager window, create a new volume of maximum size, mount it as the F: drive and format it with NTFS and a default allocation unit.
5. From the desktop of 20533E0401-vm2, open File Explorer and verify that there is a new drive F:.
6. Close the Remote Desktop session to 20533E0401-vm2.

#### Task 3: Remove the lab environment

1. On MIA-CL1, close all open windows without saving any files.
2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
4. If you have multiple Azure subscriptions, select the one you want the script to target.
5. If prompted, specify the current lab number.
6. When prompted for confirmation, type **y**.
7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
8. In the Azure portal, reset the dashboard to the default state.
9. Close all open windows.

**Result**: After completing this exercise, you should have implemented Storage Spaces based volumes.

**Question** Why would you use Storage Spaces in an Azure VM considering that Azure already provides highly available storage built into a storage account?

©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.