MOC 20533C: IMPLEMENTING MICROSOFT AZURE SOLUTIONS

Table of Contents

Lab Answer Key: Module 1: Introduction to Microsoft Azure

Lab: Managing Microsoft Azure

Exercise 1: Using the Azure portals

Task 1: Use the Azure classic portal

1. Ensure that you are signed in to the 20533C-MIA-CL1 virtual machine as **Student** with the password **Pa$$w0rd**. You should have already run the preparation script in the "Preparing the environment" demonstration at the beginning of the module.

2. On the taskbar, click the **Internet Explorer** shortcut.

3. In Internet Explorer, in the address bar, type **https://manage.windowsazure.com**, and then press Enter.

4. On the Microsoft Azure sign in page, enter the email address with which your Azure account is associated, and then click **Continue**.

5. On the Sign in page, type the email address and password you set up for this course, and then click **Sign in**.

6. On the Azure classic portal page, in the navigation bar, scroll down, and then click **SETTINGS**.

7. On the **settings** page, click **ADMINISTRATORS**, and then at the bottom of the pane, click **ADD**.

8. In the **EMAIL ADDRESS** box, type a random email address ending with **@outlook.com**, select the check box to select the free trial subscription, and then click the check mark icon.

   Note: Observe that the email address you typed is now listed as the co-administrator. An email containing an invitation to act as co-administrator has been sent to this email address.

9. On the Azure classic portal page, in the navigation bar, click **ACTIVE DIRECTORY**.

10. In the **active directory** pane, click **Default Directory**. This is the default Azure Active Directory (Azure AD) instance for your subscription.

11. On the **Let's talk about Azure AD** page, deselect all checkboxes and then click the checkmark at the bottom of the page.

12. On the **Default Directory** page, click **USERS**. Note the two accounts that are listed: your account and the co-administrator account you created earlier.

13. On the **Default Directory** page, click **DOMAINS**. Note the name of the default domain for your subscription displayed in the Default Directory pane.

Task 2: Use the Azure portal

1. In Internet Explorer, on the Azure classic portal page, at the top of the screen, click **Check out the new portal**, and then click **Launch**. If you do not have the Azure classic portal open, go to **https://portal.azure.com.**

2. On the **Dashboard** page, at the top of the screen, click **Edit dashboard**.

3. On the **Dashboard** page, on the **All resources** tile, click the ellipses (**...**), and then click **4x6**.

4. On the **Dashboard** page, on the **Service health** tile, click the ellipses (**...**), and then click **2x4**.

5. On the **Dashboard** page, at the top of the screen, click **Done customizing**.

6. On the **Dashboard** page, on the Hub menu, click **Browse**, and then click the star beside **Storage accounts**.

Task 3: Use the account page of the Azure portal

1. In Internet Explorer, in the address bar, type **https://account.windowsazure.com**, and then press Enter.

2. On the **Account** page, click **SIGN IN**.

3. On the Sign in page, type the email address and password you set up for this course, and then click **Sign in**.

4. On the **Account portal** page, click **subscriptions**.

5. On the **subscriptions** page, click the subscription you are using for this course. View the billing summary for your subscription on the page.

6. On the **subscriptions** page, on the right side of the screen, click **Download usage details.**

7. On the **Summary for Azure Pass** page, click **Download Usage**, and then click **Version 1**.

8. In Internet Explorer, when prompted whether to open or save the .csv file, click **Open**.

9. When prompted, **How do you want to open this file?** Click **Notepad** and then click **OK**.

10. View the contents of the file in Notepad. Note that this is intended to simply review its content - typically to analyze it in more details, you would use Microsoft Excel or other program capable of parsing csv files.

11. Close Notepad.

12. On the **Account portal** page, click **Preview features**.

13. On the **Preview features** page, find a preview feature and click **try it now**.

14. In the Add Preview Feature window, click the check mark to approve the preview feature. After the window closes, note the status of the feature (**You are queued**).

Result: After completing this exercise, you will have used the Azure portals.

Exercise 2: Using the Azure Resource Manager features in the Azure portal

Task 1: Create and manage a resource group

1. In Internet Explorer, in the address bar, type **https://portal.azure.com**, and then press Enter.

2. In the Azure portal, click **Resource groups**.

3. On the Resource groups blade, click **Add**.
4. On the Resource groups blade, type the following values, and then click **Create**:
    - Resource group name: **TestRG1**
    - Subscription: leave as the default
    - Resource group location: *your preferred location*

Task 2: Create Azure resources

1. On the Azure portal page, click **New**, click **Data + Storage**, and then click **Storage account**.
2. On the Create storage account blade, in the **Name** box, type a unique name for the storage account you are creating. You can use the current date and your initials to create a unique value in the format **storage***MMDDYYYYab*.

    Note: For example, a student named Ed Meadows might use **storage04252016em**. All alphabetical characters must be lowercase.
3. In the **Resource Group** section, click the dropdown box, and then click **TestRG1**.
4. In the **Location** list, select the location you have been using for the course, and then click **Create**. Note the progress of the storage account creation on the dashboard. Wait until the creation is complete before moving to the next task.

Task 3: Configure tagging

1. In the Azure portal, on the Hub menu, click **Resource groups**.
2. On the Resource groups blade, click **TestRG1**, and then in the Settings blade for TestRG1, click **Tags.**
3. On the Tags blade, in the **Key** box, type **project**, and then in the **Value** box, type **Test.** Click **Save**.
4. On the TestRG1 blade, click **storage***DDMMYYYYab*, and then in the upper-right area of the storage *DDMMYYYYab* pane, click the **Tags** icon.
5. In the Tags pane, in the **Key** box, type **project**. In the **Value** box, type **Test**. Click **Save**.
6. In the Tags pane, click the ellipses (**...**) next to **project: Test**, and then click **Pin to dashboard**.
7. On the Azure portal page, in the upper left, click **Microsoft Azure** to go to the **Dashboard** page.
8. On the **Dashboard** page, click the **project:Test** tile. View the resources associated with this tag.

Task 4: Configure RBAC

1. On the Azure portal page, in the upper left, click **Microsoft Azure** to go to the **Dashboard** page.
2. On the **Hub** menu, click **Resource groups**, and then click **TestRG1**.

3. On the Settings blade for TestRG1, under Resource Management, click **Users**.
4. On the Users blade, click **Roles**, and then on the Roles pane, click **Storage Account Contributor**.
5. In the Storage Account Contributor blade, click **Add**, and then in the Add users blade, click the user you added earlier in the lab. Click **Select**.
6. Scroll to the Users blade, and then note that the user has been added to the user list as a Storage Account Contributor.

Result: After completing this exercise, you will have used the Azure Resource Manager features in the Azure portal.

Exercise 3: Using Azure PowerShell

Task 1: Connect Azure PowerShell to your Azure subscription

1. On MIA-CL1, on the taskbar, click **Start**, type **ISE**, and then click **Windows PowerShell ISE**.
2. In the Windows PowerShell Integrated Scripting Environment (ISE), at the command prompt, type the following command, and then press Enter:
3. `Login-AzureRMAccount`
4. In the sign-in windows that appears, sign in to your Azure account.
5. In the Windows PowerShell ISE window, at the command prompt, type the following cmdlet, and then press Enter:
6. `Get-AzureRmSubscription`
7. In the Windows PowerShell ISE window, at the command prompt, type the following cmdlet, and then press Enter:
8. `Get-AzureRmResourceProvider`
9. View the Azure resource providers, resource types, and the Azure regions where these resources are available.

Task 2: Manage Azure services and resource groups

1. In the Windows PowerShell ISE window, open the **D:\Labfiles\Lab01\Starter\Lab01Starter.ps1** file.
2. In the **#Variables** section, modify the **$locName** variable to match the Azure location that your instructor asked you to use.
3. In the **#Variables** section, modify the **$webappName** variable to a unique name by using the current date and your initials in the **TestWebApp***MMDDYYAB* format.
4. Under the line that starts: **#Create a web app**, type the following code.
5. `New-AzureRmWebApp -Name $webappName -ResourceGroupName $rgname -Location $locName`
6. Select all of code in the file, including the line you just typed, right-click it, and then click **Run selection**.

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

8.
   ```
   Get-AzureRmResource | Where {$_.ResourceGroupName -eq $rgName}
   ```

9. View the list of resources that belong to the TestRG1 resource group.

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

11.
    ```
    New-AzureRMResourceGroup -Name $newrgname -location $locname
    ```

12. In the Windows PowerShell ISE window, in the script pane, under the line that starts with ** #Move the web app**, type the following code, and then press Enter:

13.
    ```
    $resource = Get-AzureRmResource -ResourceName $webappname -ResourceGroupName $rgname
    ```

14.
    ```
    Move-AzureRmResource -DestinationResourceGroupName $newrgname -ResourceId $resource.ResourceId
    ```

15. Select the code in step 9, right-click it, and then click **Run Selection**.

16. In the Confirm window, click **Yes**.

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

18.
    ```
    Get-AzureRmResource | Where {$_.ResourceGroupName -eq $newrgName}
    ```

19. View the web app you created earlier, which is now in the TestWebRG resource group.

Task 3: Reset the environment

1. Close all open apps without saving any files.

2. On the taskbar, right-click **Windows PowerShell**, and then click **Run as administrator**. In the **User Account Control** dialog box, click **Yes**.

3. Type the following command, and then press Enter:

4.
   ```
   Reset-Azure
   ```

5. When prompted (twice), sign in by using the Microsoft account associated with your Azure subscription.

6. If you have multiple Azure subscriptions, select the one you want the script to target.

7. When prompted for confirmation, type **y**.

Note: This script removes 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 resets your Azure environment so that it is ready for the next lab. The script removes all storage accounts, virtual machines, virtual networks, cloud services, and resource groups containing these resources.

Result: After completing this exercise, you will have used Azure PowerShell to create and manage Azure resources.

Lab Answer Key: Module 2: Implementing and managing Azure networking

Lab A: Using a deployment template and Azure PowerShell to implement Azure virtual networks

Exercise 1: Creating an Azure virtual network by using a deployment template

Task 1: Access the template on GitHub

1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa$$w0rd**.
2. On the desktop, on the taskbar, click the **Internet Explorer** icon.
3. In the Microsoft Internet Explorer Address bar, type the following address, and then press Enter: http://aka.ms/Mt32e4.
4. Open a GitHub template that you can use to create a virtual network with two subnets.

Task 2: Load the template into new deployment on the Azure portal

1. In Internet Explorer, under Virtual Network with two Subnets, click **Deploy to Azure**.
2. When prompted, sign in using the Microsoft account associated with your Azure subscription.
3. In the Azure portal, in the Custom deployment blade, click the **Edit Template** link.
4. Review the structure of the JavaScript Object Notation (JSON) file. Examine the placeholders for values that can be edited during the deployment. This template contains the following parameters that you can edit: *location, vnetName*, *vnetAddressPrefix*, _ subnet1Name_, _ subnet1Prefix, subnet2Name_, _ subnet2Prefix._
5. Review the content under resources to identify type of the resource, its name, and properties.
6. Click **Discard** to close the Edit Template blade.

Note: If the template fails to load into the Azure portal, navigate to the following URL: http://aka.ms/Fpqovq . Then, select and copy all the text. Paste the copied text into the Edit Template blade, and then perform steps 4 and 5 to review the template.

Task 3: Run the deployment from the Azure portal

1. In the Custom deployment blade click **Edit Parameters**.
2. Type the following information for the Parameters, and then click **OK**.
    - VNETNAME: ** HQ**
    - VNETADDRESSPREFIX: **10.0.0.0/16**
    - SUBNET1NAME: **Subnet1**
    - SUBNET1PREFIX: **10.0.0.0/24**

    - SUBNET2NAME: ** Subnet2**
    - SUBNET2PREFIX: **10.0.1.0/24**

3. In the Custom Deployment blade, under the Resource Group section, from the drop-down list select **New.** In the New resource group name field type **AdatumLabRG** to create a new Resource group with that name.
4. In the Custom Deployment blade under Resource group location drop-down list select *<Location1>*.
5. In the Custom Deployment blade, click **Legal Terms** link. Review the Terms of use, and then click **Purchase**.
6. In the custom deployment blade, click **Create** to create the new virtual network.
7. Verify that provisioning of the new virtual network with name **HQ** completed successfully.

Result: After completing this exercise, you should have created virtual networks for A. Datum HQ.

Exercise 2: Creating a virtual network by using PowerShell

Task 1: Create a virtual network by using PowerShell

1. From the taskbar, start Windows PowerShell.
2. Sign in to your subscription by typing the following command, and then pressing Enter:
3. `Login-AzureRMAccount`
4. To select the subscription in which you are going to create a virtual network, type the following commands, and then press Enter after each (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):
5. `Get-AzureRmSubscription`
6. `Set-AzureRmContext -SubscriptionName 'Name of your subscription'`
7. To create a new resource group, type the following command, and then press Enter (replace ' *Location*' with the actual name of the primary Azure region provided by the instructor and make sure to enclose the name of the region in single quotes):
8. `New-AzureRMResourceGroup -Name AdatumTestRG -Location 'Location1'`
9. To create a new virtual network named **AdatumTestVnet** with the address space **10.0.0.0/16** and store a reference to it in the $vnet variable, type the following command, and then press Enter (replace ' *Location*' with the actual name of the primary Azure region provided by the instructor and make sure to enclose the name of the region in single quotes):
10. `$vnet = New-AzureRMVirtualNetwork -ResourceGroupName AdatumTestRG -Name AdatumTestVnet -AddressPrefix 10.0.0.0/16 -Location 'Location1'`
11. To add a subnet to the new virtual network, type the following command, and then press Enter:
12. `Add-AzureRmVirtualNetworkSubnetConfig -Name FrontEnd -VirtualNetwork $vnet -AddressPrefix 10.0.0.0/24`

13. To update the configuration in the virtual network, type the following command, and then press Enter:

```
Set-AzureRMVirtualNetwork -VirtualNetwork $vnet
```

Result: After completing this exercise, you should have created a test virtual networks for A. Datum by using Azure PowerShell.

Exercise 3: Configuring virtual networks

Task 1: Create an IaaS v1 virtual network gateway

1. Switch to Internet Explorer and in the new tab navigate to open the classic Azure portal by typing http://manage.windowsazure.com
2. If you are prompted to sign-in, use an account that is either a Service Admin or a co-admin of your Azure subscription.
3. From the navigation bar on the left hand side, select networks, and then click **ADATUM-BRANCH-VNET**.
4. On the **adatum-branch-vnet** page, click **DASHBOARD**,
5. From the command bar located at the bottom of the page, click **CREATE GATEWAY** and then select **Dynamic Routing**.
6. Click **Yes** to confirm creation of a VPN gateway.

Note: The creation of the VPN gateway could take 30 - 35 minutes to complete.

Task 2: Deploy an IaaS v2 virtual machine into an IaaS v2 virtual network

1. Right-click Windows PowerShell shortcut in the taskbar and click **Run ISE as Administrator**.
2. In the Windows PowerShell ISE, at the command prompt, type the following command, and then press Enter:
3. `CD D:\Labfiles\Lab02\Starter`
4. At the command prompt, type the following command, and then press Enter:
5. `.\CreateVirtualMachine.ps1`

Note: The command starts with dot backslash. 4. When prompted to sign in (twice), type in the user name and the password which is either the Service Administrator or a Co-Admin in your Azure subscription.

1. If you have multiple subscription, when prompted, type in the number corresponding to the subscription to which you deployed the virtual network in the first exercise of this lab and press Enter.

Note: The script takes about 10 minutes to complete. The script deploys an IaaS v2 virtual machine named ARMSrv2 onto the first subnet of the IaaS v2 HQ virtual network you provisioned earlier in this lab.

Result: After completing this exercise, you should have created a virtual network gateway on the existing IaaS v1 virtual network and deployed a virtual machine to the newly created IaaS v2 HQ virtual network.

Lab B: Configuring connectivity between IaaS v1 and IaaS v2

Exercise 1: Using a PowerShell script to connect IaaS v1 VNet and IaaS v2 VNet

Task 1: Configure Resource Manager virtual network

1. On MIA-CL1, launch Internet Explorer and browse to the classic Azure portal by typing http://manage.windowsazure.com
2. If you are prompted to sign-in, use an account that is either a Service Admin or a co-admin of your Azure subscription.
3. From the navigation bar on the left hand side, select networks, and then click **ADATUM-BRANCH-VNET**.
4. On the **adatum-branch-vnet** page, click **DASHBOARD**,
5. Ensure that the provisioning of the new virtual gateway that you started in the first lab of this module has been completed. You can determine this by checking whether **DELETE GATEWAY** button appears in the command bar at the bottom of the page. If it is there, the gateway provisioning has completed. If not, wait until icon appears in the command bar. You might need to occasionally refresh the page. Keep in mind that provisioning can take about 30-35 minutes
6. On MIA-CL1, from the Azure PowerShell window, set the current directory to **D:\Labfiles\Lab02\Starter** by typing
7. `CD D:\Labfiles\Lab02\Starter`
8. Open the **ConfigureARMGateway.ps1** file in Notepad and review its content.
9. From the Azure PowerShell window, run **ConfigureARMGateway.ps1** by typing the following command, and then pressing Enter.
10. `.\ConfigureARMGateway.ps1`
11. When prompted to sign-in (twice), use an account that is either a Service Admin or a co-admin of your Azure subscription.
12. Occasionally monitor the execution status.

Note: The script might take 20-25 minutes to complete. You do not have to wait for the script to finish. You can proceed with second task of this exercise and with Exercise 2 from this lab.

Task 2: Configure classic virtual network

1. On MIA-CL1, launch Internet Explorer and browse to the Azure Portal at https://portal.azure.com
2. If prompted, sign in to your Azure subscription when prompted with an account that is a Service admin or a co-admin of your subscription.

3. In the Azure portal, in the Hub menu, click Browse, scroll down towards the bottom of the list of services and click **Virtual networks.**

4. In the Virtual networks blade, click **HQ**.

5. In the HQ blade, in the **Connected devices** section, take the note of the value in the **IP ADDRESS** column for **gatewayARM**.

6. On MIA-CL1, launch File Explorer and browse to D:\Configfiles\Lab02 folder.

7. Right-click on the **NetworkConfig.xml** file and click **Open with** in the menu.

8. In the next cascading menu, click **Notepad** in the list of programs.

9. In Notepad, under the **LocalNetworkSite** section, modify the value of **<VPNGatewayAddress>** (which is at this point set to **1.1.1.1**) by replacing **1.1.1.1** with the value of the IP address that you recorded in step 5, save the changes to **NetworkConfig.xml**, and then close the file.

10. On MIA-CL1, launch Windows PowerShell as Administrator by using the taskbar icon.

11. At the Windows PowerShell prompt, sign into your Azure subscription by running:
12. `Add-AzureAccount`
13. If you have multiple subscriptions, to select the target subscription, type the following commands, and then press Enter after each (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):
14. `Get-AzureSubscription`
15. `Set-AzureSubsciption -SubscriptionName 'Name of your subscription'`
16. At the Windows PowerShell command prompt, type the following command, and then press Enter
17. `Set-AzureVNetConfig -ConfigurationPath D:\Configfiles\Lab02\NetworkConfig.xml`
18. To set the IPSec shared key for the classic VNet gateway, type the following command at the Windows PowerShell command prompt, and then press Enter.
19. `Set-AzureVNetGatewayKey -VnetName Adatum-Branch-Vnet -LocalNetworkSiteName  HQ -SharedKey 12345`
20. Wait for the command to complete and display the StatusCode OK.

21. Open Internet Explorer.

22. In Internet Explorer, browse to the classic Azure Management Portal at **https://manage.windowsazure.com**, and, if prompted, sign in using the Microsoft account that is either the Service Admin or a co-Admin of your subscription.

23. Click **NETWORKS** in the navigation bar on the left hand side.

24. On the **networks** page, click, **ADATUM-BRANCH-VNET**.

25. On the **adatum-branch-vnet** page, click **DASHBOARD**.

26. On the **DASHBOARD** page, verify that the **ADATUM-BRANCH-VNET** and **HQ** are connected. You might need to click **CONNECT** in the menu bar or refresh the Internet Explorer page.

27. Leave the Internet Explorer window open.

Result: After completing this exercise, you should have connected the A. Datum HQ and branch virtual networks, and deployed dynamic routing gateways for each virtual network.

Exercise 2: Configuring a point-to-site VPN

Task 1: Configure a VPN from a client to the headquarters virtual network

1. In Internet Explorer, on the **adatum-branch-vnet** page, click **CONFIGURE**.

2. Select the **Configure point-to-site connectivity** checkbox.

3. In the **STARTING IP** text box, type **172.16.0.0**, and then set the **CIDR (ADDRESS COUNT)** to **/24 (254)**.

4. At the bottom of the page, click **SAVE**.

5. In the warning message, click **Yes**.

6. Press the Windows logo key, and then type **Command**.

7. Right-click **Command Prompt**, and then click **Run as administrator**.

8. In the **User Account Control** dialog box, click **Yes**.

9. At the Command Prompt, type the following command, and then press Enter:

10. `CD C:\Program Files (x86)\Windows Kits\10\bin\x64`

11. Type the following command, and then press Enter:

12. `makecert -sk exchange -r -n "CN=AdatumRootCertificate" -pe -a sha1 -len 2048 -ss My "AdatumRootCertificate.cer"`

13. Switch back to Internet Explorer.

14. On the **adatum-branch-vnet** page, click **CERTIFICATES**.

15. Click **UPLOAD A ROOT CERTIFICATE**.

16. In the **Upload Certificate** dialog box, click **BROWSE FOR FILE**.

17. In the **Choose File to Upload** dialog box, navigate to **C:\Program Files (x86)\Windows Kits\10\bin\x64**, select ** AdatumRootCertificate**, and then click ** Open**.

18. In the **Upload Certificate** dialog box, click the **check mark** icon.

19. Switch back to the Command Prompt window.

20. At the command prompt, type the following command, and then press Enter:

21. `makecert.exe -n "CN=AdatumClientCertificate" -pe -sk exchange -m 96 -ss My -in "AdatumRootCertificate" -is my -a sha1`

22. Switch back to Internet Explorer.

23. Press the **ALT** key, click **Tools**, and then click **Internet Options**.

24. In **Internet Options**, click the **Content** tab, and then click **Certificates**.
25. Verify that both the **AdatumClientCertificate** and ** AdatumRootCertificate** display in the Personal store.
26. Close Certificates.
27. Close Internet Options.

Task 2: Connect to the HQ virtual network

1. Switch back to Internet Explorer tab with the classic portal.
2. On the **adatum-hq-vnet** page, click **DASHBOARD**.
3. In the quick glance section, click **Download the 64-bit Client VPN Package**.
4. When prompted whether to save or run the file, click **Run**.
5. In the **Windows protected your PC** dialog box, click **More Info**, and then click **Run anyway**.
6. In the **User Account Control** dialog box, click **Yes**.
7. In the **ADATUM-BRANCH-VNET** dialog box, click **Yes**.
8. In the **Search the web and Windows** textbox, type ** ncpa.cpl** and press Enter.
9. In the Network Connections window, right-click **ADATUM-BRANCH-VNET**, and then click **Connect/Disconnect**. This will launch the Settings app with the VPN tab of **NETWORK & INTERNET** page.
10. Click **ADATUM-BRANCH-VNET**, and then click **Connect**. This will open the **ADATUM-BRANCH-VNET** dialog box.
11. Click **Connect** again.
12. When prompted to accept that the Connection Manager needs elevated privileges to run CMROUTE.DLL, click **Continue**.
13. After the connection is established, switch back to the Command Prompt window
14. At the command prompt, type the following command, and then press Enter:
15. `ipconfig /all`
16. In the results, verify that there is a PPP adapter **ADATUM-BRANCH-VNET** section, and that you have an assigned IP address from the IP address range you defined for the point to site connectivity (172.16.0.0/24).
17. In the Search the web and Windows textbox in the taskbar of MIA-CL1, type the following and press Enter:
18. `Mstsc /admin /f /v:192.168.0.4`
19. If prompted whether to connect, click **Yes**.
20. In the **Windows Security** dialog box, enter the following credentials, and then click **OK**:
    - User name: **Student**

- Password: **Pa$$w0rd123**

21. If prompted again whether to connect, click **Yes**.
22. This will establish a Remote Desktop session to the private IP address of ClassicSrv1 Azure virtual machine. Verify that you can successfully log on to ClassicSrv1.

    Note: Note that you could potentially also test connectivity to a file share on ClassicSrv1 Azure virtual machine or ping it by its IP address, however, that would require modifying Windows Firewall settings on ClassicSrv1 in order to allow File and Printer Sharing traffic.

23. Close the Remote Desktop session.
24. Switch back to the VPN tab of the **NETWORK & INTERNET** page of the Settings app.
25. Click **Disconnect**, next to the **ADATUM-BRANCH-VNET** entry.

Result: After completing this exercise, you should have configured and tested a point-to-site VPN connection.

Exercise 3: Validating virtual network connectivity

Task 1: Connect to the A. Datum VMs

1. On MIA-CL1, Open Internet Explorer and navigate to the Azure classic portal at https://manage.windowsazure.com.
2. In the Azure classic portal, click **VIRTUAL MACHINES** in the navigation bar on the left side of the window.
3. Make sure that **ClassicSrv1** is selected, and then click **CONNECT** in the menu bar at the bottom of the window.
4. When prompted whether to open or save the .rdp file, click **Open**.
5. If a Remote Desktop Connection warning message displays, select **Don't ask me again for connections to this computer**, and then click **Connect**.
6. In the **Windows Security** dialog box, type the following credentials, and then click **OK**:
    - User name: **Student**
    - Password: **Pa$$w0rd123**
7. If another Remote Desktop Message displays, select **Don't ask me again for connections to this computer**, and then click **Yes**.
8. Minimize the ClassicSrv1 RDP session.
9. From MIA-CL1, launch Internet Explorer and navigate to the Azure portal at https://portal.azure.com.
10. In the Azure portal, click **Browse** in the hub menu on the left hand side of the window and click **Virtual machines**.

11. On the **Virtual machines** blade, click **ARMSrv2**.
12. On the **ARMSrv2** blade, click **Connect** in the toolbar.
13. When prompted whether to open or save the ** .rdp** file, click **Open**.
14. If a Remote Desktop Connection warning message displays, select **Don't ask me again for connections to this computer**, and then click **Connect**.
15. In the **Windows Security** dialog box, type the following credentials, and then click **OK**:
    - User name: **Student**
    - Password: **Pa$$w0rd123**
16. If another Remote Desktop Message displays, select **Don't ask me again for connections to this computer**, and then click **Yes**.

Task 2: Test TCP/IP connectivity between the sites

1. From the ARMSrv2 RDP session, if prompted whether to enable network discovery, click **No**.
2. In Server Manager, click **Local Server**.
3. If the status of Windows Firewall is **On** for the Public profile, click **Public:On**.
4. In Windows Firewall, click **Turn Windows Firewall on or off**.
5. Under Public network settings, click **Turn off Windows Firewall (not recommended)**, and then click **OK**.
6. Close the **Windows Firewall** window.
7. Minimize the ARMSrv2 RDP session.
8. Maximize the ClassicSrv1 RDP session.
9. If prompted whether to enable network discovery, click **No**.
10. In Server Manager, click **Local Server**.
11. If the status of Windows Firewall is **On** for the Public profile, click **Public:On**.
12. In Windows Firewall, click **Turn Windows Firewall on or off**.
13. Under Public network settings, click **Turn off Windows Firewall (not recommended)**, and then click **OK**.
14. Close the **Windows Firewall** window.
15. In the ClassicSrv1 RDP session, on the taskbar, click the **Windows PowerShell** icon.
16. At the Windows PowerShell command prompt, type the following command, and then press Enter:
17. `Ping 10.0.0.4`
18. Verify that ARMSrv1 responds to Internet Control Message Protocol (ICMP) messages.

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:
5. `Reset-Azure`
6. When prompted, sign in by using the Microsoft account associated with your Azure subscription.
7. If you have multiple Azure subscriptions, select the one you want to target with the script.
8. 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 ready it for the next lab. The script removes all storage, virtual machines (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 find objects remaining after the reset script is complete, you can re-run **Reset-Azure** script, or use the full Azure Management Portal to manually delete all the objects in your Azure subscription, with the exception of the default directory.

Result: After completing this exercise, you should have verified that VMs can communicate between the virtual networks.

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 Virtual Machines 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

    - **User name:** Student

    - **Password:** Pa$$w0rd

    - **Subscription:***Your subscription*

7. **In the Resource group section, click the drop-down list and clickResDevRG**.

8. Accept the default Location value and click **OK**.

9. On the Choose a size blade, click **A1 Standard**, and then click **Select**.

10. On the Settings blade, ensure that **HQ-VNET** is selected as the **Virtual network**.

11. Click **Subnet**.

12. On the Choose subnet blade, click **Database**.

13. On the Settings blade, click **OK**

14. 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:\Labfiles\Lab03\Starter\**
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:
2. `Get-AzureRmResource | where {$_.ResourceType -like "*VirtualMachines"}`
3. 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**
6. 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, on the taskbar, click the **Visual Studio** icon.
2. In Visual Studio, click **File**, click **Open**, click **Project/Solution**, and then browse to **D:\Labfiles\Lab03\Starter\ResDev\ResDevLinuxDeploy**.
3. In the Open Project window, click **ResDevLinuxDeploy.sln**, and then click **Open**.
4. In Visual Studio, in the Solution Explorer pane, expand **Templates**, and then click **azuredeploy.json**.
5. View the parameters and variables section of the template.
6. In the Solution Explorer pane, right-click **ResDevLinuxDeploy**, click **Deploy**, and then click **New Deployment**.
7. 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.
8. In the Deploy to Resource Group window, click the **Resource Group** drop-down box, and then click **ResDevRG**.
9. In the Deploy to Resource Group window, click **Edit Parameters**.
10. 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
11. In the Edit Parameters window, click the **Save passwords** check box, and then click **Save**.
12. 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.

13. 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.
14. In the Solution Explorer pane, right-click **ResDevLinuxDeploy**, click **Deploy**, and then click **ResDevRG**.
15. In the Deploy to Resource Group window, click **Edit Parameters**.
16. In the Edit Parameters window, in the **vmNameValue** box, type **ResDevApp2**, and then click **Save**.
17. 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.
18. In the Visual Studio, click **File** and then on the drop-down menu, click **Close Solution**.
19. 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:
3. `Login-AzureRMAccount`
4. When prompted, sign in to your Azure subscription with an account that is either the Service Administrator or Co-administrator of your Azure subscription.
5. If you have multiple subscriptions associated with your account, at the Windows PowerShell ISE prompt, type the following cmdlet, and then press Enter:
6. `Get-AzureRmSubscription`
7. 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):
8. `Set-AzureRmContext -SubscriptionName 'Name of your subscription'`
9. Type the following cmdlet, and then press Enter:
10. `Find-AzureRMResource -ResourceGroupNameContains ResDevRG | Format-Table -Property ResourceName, ResourceType`

11. 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.

12. 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:\Labfiles\Lab03\Starter\ResDev** 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:\Labfiles\Lab03\Starter\ResDev** 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**

10. 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:
5. `Reset-Azure`
6. When prompted, sign in by using the Microsoft account associated with your Azure subscription.
7. If you have multiple Azure subscriptions, select the one you want to target with the script.
8. 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.

Lab Answer Key: Module 4: Managing virtual machines

Lab: Managing Azure virtual machines

Exercise 1: Configuring availability

Task 1: Create virtual machines in an availability set

1. On MIA-CL1, open Internet Explorer and navigate to the Azure portal at https://portal.azure.com.

2. When prompted, sign in with an account that is either a Service Administrator or Co-Admin in the subscription you are using for this lab.

3. On the **Hub** menu, click **+New**. In the **New** blade, in the **Search the marketplace** text box, type **Availability Set**, and press the Enter keyboard key.

4. In the **Everything** blade, select **Availability Set**.

5. In the **Availability Set** blade, click **Create**.

6. In the **Create availability set** blade, specify the following settings:

   - Name: **ResDevWebAS**

   - Fault domains: **3**

     Note: You can decrease the value to 2, but not increase it.

   - Update domains: **5**> **Note:** The number of update domains can vary between 5 and 20.
   - Subscription: ***Y our Azure subscription you intend to use for this demo***.

   - Resource group name: **ResDevWebRG**

   - Location: ***The Azure region closest to the location of your lab computer***.

7. Click **Create**. Wait for deployment to complete. This should take only a few seconds.

8. On the **Startboard** of the Azure portal, click **+New** on the **Hub** menu.

9. In the **New** blade, click **Virtual Machines**.

10. In the **Virtual Machines** blade, click **Windows Server 2012 R2 Datacenter**.

11. In the **Windows Server 2012 R2 Datacenter** blade, ensure that **Resource Manager** appears in the **Select a deployment model** drop-down list, and then click **Create**. The **Create virtual machine** blade appears and expands its **Basics** blade.

12. In the **Basics** blade, specify the following:

    - Name: **ResDevWebVM1**

    - VM disk type: **HDD**

- User name: **Student**
- Password: **Pa$$w0rd1234**
- Subscription: ***Your Azure subscription you intend to use for this demo***.
- Resource group: **ResDevWebRG**
- Location: ***The same location you chose for the availability set***.

13. Click **OK**. The **Choose a size** blade automatically appears.
14. In the **Choose a size** blade, select **A1 Standard**, and then click **Select**. The **Settings** blade appears.
15. In the **Settings** blade, specify the following settings:
    - Disk type: **Standard**
    - Storage account: ***Accept the default***.
    - Virtual network: **ResDevWebAS**
    - Subnet: ***Accept the default***.
    - Public IP address: **ResDevWebVM1**
    - Network security group: **ResDevWebVM1**
    - Monitoring: **Disabled**
    - Availability set: **ResDevWebAS**
16. Click **OK**. The **Summary** blade automatically appears.
17. In the **Summary** blade, click **OK**.
18. On the **Startboard** of the Azure portal, click **+New** on the **Hub** menu.
19. In the **New** blade, click **Virtual Machines**.
20. In the **Virtual Machines** blade, click **Windows Server 2012 R2 Datacenter**.
21. In the **Windows Server 2012 R2 Datacenter** blade, ensure that **Resource Manager** appears in the **Select a deployment model** drop-down list, and then click **Create**. The **Create virtual machine** blade appears and expands its **Basics** blade.
22. In the **Basics** blade, specify the following:
    - Name: **ResDevWebVM2**
    - VM disk type: **HDD**
    - User name: **Student**
    - Password: **Pa$$w0rd1234**
    - Subscription: ***Your Azure subscription you intend to use for this demo.***

- Resource group: **ResDevWebRG**
- Location: ***The same location you chose for the availability set***.

23. Click **OK**. The **Choose a size** blade automatically appears.
24. In the **Choose a size** blade, select **A1 Standard**, and then click **Select**. The **Settings** blade automatically appears.
25. In the **Settings** blade, specify the following settings:
    - Disk type: **Standard**
    - Storage account: ***Accept the default***.
    - Virtual network: **ResDevWebAS**
    - Subnet: ***Accept the default***.
    - Public IP address: **ResDevWebVM2**
    - Network security group: **ResDevWebVM2**
    - Monitoring: **Disabled**
    - Availability set: **ResDevWebAS**
26. Click **OK**. The **Summary** blade automatically appears.
27. In the **Summary** blade, click **OK**.
28. On the **Startboard** of the Azure portal, click **Browse** on the **Hub** menu.
29. In the list of services, select **Availability sets**.
30. On the **Availability sets** blade, click **ResDevWebAS**.
31. On the **ResDevWebAS** blade, note that the availability set contains the two newly deployed virtual machines (at this point, both of them will likely display the **Creating** status). Point out that each VM has a unique fault domain and update domain.
32. Leave the instance of Internet Explorer with the Azure portal open.

Task 2: Configure the Azure Load Balancer

1. On MIA-CL1, in the Azure portal within the Internet Explorer window, on the Hub menu, click **New**, click **Networking**, and then, in the **Networking** blade, click **Load Balancer**.
2. In the **Create load balancer** blade, specify the following settings:
    - Name: **ResDevWebLB**
    - Scheme: **Public**
    - Public IP address: ***Create*** **_ a_ _ new dynamic address named_ ResDevWebLBIP**.
    - Subscription: **Your subscription**.
    - Resource group: **ResDevWebRG**

- Location: ***The same location you chose for the availability set***.
- Pin to dashboard: **Unchecked**

3. Click **Create**. Wait for the deployment to complete. This should take a few seconds.
4. On the **Startboard** of the Azure portal, click **Browse**, select **Load Balancers** from the list of services, and then in the **Load balancers** blade, click **ResDevWebLB**.
5. In the **Setting** blade of the ResDevWebLB Load balancer, click **Backend pools**, and then, in the **Backend address pools** blade, click **Add**.
6. In the **Add backend pool** blade, in the **Name** text box, type **ResDevWebLBPool**, and then click **Add a virtual machine**.
7. In the **Choose virtual machines** blade, click **Choose an availability set**, and then, in the **Choose an availability set** blade, click **ResDevWebAS**.
8. In the **Choose virtual machines** blade, click **Choose the virtual machines**, click the check boxes to the left of ResDevWebVM1 and ResDevWebVM2, and then click **Select**.
9. In the **Choose virtual machines** blade, click **OK**.
10. In the **Add backend pool** blade, click **OK**.
11. In the **Setting** blade for ResDevWebLB, click **Probes**, and then in the **Probes** blade, click **Add**.
12. In the **Add probe** blade, specify the following settings, and then click **OK**:
    - Name: **ResDevWebProbe80**
    - Protocol: **HTTP**
    - Port: **80**
    - Path: **/**
    - Interval: **5**
    - Unhealthy threshold: **2**
13. In the **Setting** blade for ResDevWebLB, click **Load balancing rules**, and then in the **Load balancing rules** blade, click **Add**.
14. In the **Add load balancing rule** blade, complete the following , and then click **OK**:
    - Name: **ResDevWebLBRule80**
    - Protocol: **TCP**
    - Port: **80**
    - Backend Pool: **ResDevWebPool**
    - Probe: **ResDevWebProbe**

- Backend port: **80**
- Session persistence: **None**
- Idle timeout: **4**
- Floating IP: **Disabled**

15. Refresh the Azure portal. In the **Setting** blade of ResDevWebLB, you should be able to identify its public IP address. 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. You will change these settings later in this lab.

Result: After completing this exercise, you should have created an availability set for Azure IaaS v2 virtual machines and configured them up as a load balanced pair.

Exercise 2: Implementing DSC

Task 1: Install and configure IIS by using DSC and Windows PowerShell

1. On MIA-CL1, start File Explorer and browse to the **D:\Labfiles\Lab04\Starter** folder.
2. In the D:\Labofiles\Lab04 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 2012 R2 Web-Server role.
4. Close the Windows PowerShell ISE window.
5. In the File Explorer, right click on the **D:\Labfiles\Lab04\Starter\DeployAzureDSC.ps1** file and select **Edit** from the right-click menu. This will open the file in the **Windows PowerShell ISE** window.
6. Review the content of the script. Note the variables that it uses, including the storage account and its key. The script first publishes the DSC configuration defined in the **Install.ps1** file to the same storage account hosting the VHD files of the two virtual machines (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 in the previous lab 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 at https://portal.azure.com.
9. Within the Azure portal, click **Virtual Machines** on the Hub menu.

10. In the Virtual machines blade, click the **ResDevWebVM1** entry.

11. In the **ResDevWebVM1** blade, click **Connect**.

12. When prompted whether to open or save the **ResDevWebVM1.rdp** file, click **Open**.

13. If prompted to confirm that you want to connect, click **Connect**.

14. When prompted to enter credentials to connect, type **Student** as the user name and **Pa$$w0rd** as the password.

15. If prompted again to confirm that you want to connect, click **Yes**.

16. After 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.

17. Repeat steps 10 through 16 for the other virtual machine ResDevWebVM2.

18. After completing the tasks, switch back to your lab computer MIA-CL1. Leave both Remote Desktop sessions open.

Task 2: Test the DSC configuration and virtual machine availability

1. In the Azure portal within the Internet Explorer window on MIA-CL1, on the Hub menu, click **Resource groups**.

2. In the **Resource groups** blade, click **ResDevWebAS**.

3. In the **ResDevWebAS** blade, in the **Resources** tile, click the **ResDevWebVM1** entry representing the network security group (with the icon in the form of a shield). This will open the corresponding blade along with its **Settings** blade.

4. In the **Settings** blade of the **ResDevWebVM1** network security group, click **Inbound security rules**.

5. In the **Inbound security rules** blade, click **Add**.

6. In the **Add inbound security rue** blade, specify the following settings:

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

7. Click **OK**.

8. In the breadcrumb trail at the top of the portal interface, click **ResDevWebAS** to return to the **ResDevWebAS** blade.
9. In the **ResDevWebAS** blade, in the **Resources** tile, click the **ResDevWebVM2** entry representing the network security group (with the icon in the form of a shield). This will open the corresponding blade along with its **Settings** blade.
10. In the **Settings** blade of **ResDevWebVM2** network security group, click **Inbound security rules**.
11. In the **Inbound security rules** blade, click **Add**.
12. In the **Add inbound security rules** blade, specify the following settings:
    - Name: **allow-http**
    - Priority: **1100**
    - Source: **Any**
    - Protocol: **TCP**
    - Source port range: *****
    - Destination: **Any**
    - Destination port range: **80**
    - Action: **Allow**
13. Click **OK**.
14. In the breadcrumb trail at the top of the portal interface, click **ResDevWebAS** to return to the **ResDevWebAS** blade.
15. In the **ResDevWebAS** blade, in the **Resources** tile, click **ResDevWebLB**, representing the load balancer.
16. In the **ResDevWebLB** blade, note the value of its **IP address** entry.
17. Open a new InPrivate Browsing Internet Explorer session and type the IP address that you noted in the previous step in the navigation bar, and then press the **Enter** key.
18. Verify that you can access the default IIS webpage.
19. Close the **InPrivate Browsing** window.
20. Switch to the Remote Desktop session on **ResDevWebVM1**. On the **Tools** menu in the **Server Manager** window, select **Services**.
21. In the **Services** window, scroll down to the **World Wide Web Publishing Service** entry, right click on it, and select **Stop** on the context sensitive menu.
22. Switch to the Remote Desktop session on **ResDevWebVM2**. On the **Tools** menu in the **Server Manager** window, select **Services**.

23. In the **Services** window, scroll down to the **World Wide Web Publishing Service** entry, right click on it, and select **Stop** on the context sensitive menu.

24. Switch back to MIA-CL1. From MIA-CL1, open a new InPrivate Browsing Internet Explorer session. In the new Internet Explorer window, click the **cogwheel** icon in the upper-right corner, click **Safety** on the drop-down menu, and then click **Delete browsing history**.

25. In the **Delete Browsing History** dialog box, click **Delete**.

26. In the InPrivate Browsing Internet Explorer window, type the IP address of the load balancer in the navigation bar, and then press the **Enter** key.

27. Verify that the **This page can't be displayed** message appears.

28. Switch back to the **Services** window the Remote Desktop session on ResDevWebVM1.

29. In the **Services** window, right-click the **World Wide Web Publishing Service** entry, and then select **Start** from the right-click menu.

30. 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 IIS webpage.

Note: Optionally you can repeat this sequence, but this time stopping the **World Wide Web Publishing Service** on ResDevWebVM1 and starting it on ResDevWebVM2. 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 implemented DSC.

Exercise 3: Implementing Storage Space-based volumes

Task 1: Attach VHDs to an Azure VM

1. On MIA-CL1, in the Azure portal within the Internet Explorer window, on the Hub menu, click **Virtual Machines**.

2. In the **Virtual machines** blade, click **ResDevWebVM1**. This automatically opens the **ResDevWebVM1** blade and its **Settings** blade.

3. In the **Settings** blade of ResDevWebVM1, click **Disks**.

4. In the **Disks** blade, click **Attach new**.

5. In the **Attach new disk** blade, specify the following settings, and then click **OK**:

    - Name: ***A ccept the default***
    - Type: **Standard**
    - Size: **1023**
    - Location: ***N ote that this cannot be changed since the location of the VM determines the location of its disks***.
    - Host caching: **None**

6. Repeat the steps 4 and 5 to attach one more new data disks. Note that with current VM size (Standard A1), there is a limit of 2 data disks per VM.

Task 2: Configure a Storage Spaces simple volume

1. On MIA-CL1, switch to the Remote Desktop session to ResDevWebVM1.
2. In the Remote Desktop session, in the **Server Manager** window, click **File and Storage Services**.
3. In the **Servers** navigation pane on the left side, click **Storage Pools**.
4. In the **STORAGE POOLS** pane, click the **TASKS** menu, and then click **New Storage Pool** on the drop-down menu. This will open the New Storage Pool Wizard.
5. On the **Before you begin** page, click **Next**.
6. On the **Specify a storage pool name and subsystem** page, type **StoragePool1** in the **Name** text box, and then click **Next**.
7. On **Select physical disks for the storage pool**, select the check boxes next to **PhysicalDisk2** and **PhysicalDisk3** (which represent disks you attached in the Azure portal), and then click **Next**.
8. On the **Confirm selections** page, click **Create**.
9. On the **View results** page, select the **Create a virtual disk when this wizard closes** check box, and then click **Close**. This will launch the New Virtual Disk Wizard.
10. On the **Before you begin** page, click **Next**.
11. On the **Select the storage pool** page, ensure that **StoragePool1** is selected, and then click **Next**.
12. On the **Specify the virtual disk name** page, type **VirtualDisk1** in the **Name** text box, and then click **Next**.
13. On the **Select the storage layout** page, ensure that **Simple** is selected, and then click **Next**.
14. On the **Specify the provisioning type** page, ensure that **Fixed** is selected, and then click **Next**.
15. On the **Specify the size of the virtual disk**, select **Maximum size**, and then click **Next**.
16. On the **Confirm selections** page, click **Create**.
17. On the **View results** page, ensure that the **Create a volume when this wizard closes** check box is selected, and then click **Close**. This will open the New Volume Wizard.
18. On the **Before you begin** page, click **Next**.
19. On the **Select the server and disk** page, ensure that **VirtualDisk1** is selected, and then click **Next**.
20. On the **Specify the size of the volume** page, accept the default (2.00 TB), and then click **Next**.

21. On the **Assign to a drive letter or folder** page, accept the default drive letter (F:), and then click **Next**.

22. On the **Select file system settings** page, accept the default settings (NTFS with default allocation unit size), and then click **Next**.

23. On the **Confirm selections** page, click **Create**.

24. On the **Completion** page, click **Close**.

25. From the desktop of ResDevWebVM1, open File Explorer, and then verify that there is a new drive F with 2 TB of available disk space.

26. Close the Remote Desktop session to ResDevWebVM1.

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 box, click **Yes**.

3. Type the following command, and press Enter:

4. `Reset-Azure`
5. When prompted (twice), sign in using the Microsoft account associated with your Azure subscription.

6. If you have multiple Azure subscriptions, select the one you want to target by the script.

7. When prompted for confirmation, type **y**.

Note: This script will remove Azure services in your subscription. We, therefore, 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 to 10 minutes to reset your Microsoft Azure environment, before it is ready for the next lab. The script removes all storage, VMs, virtual networks, cloud services, and resource groups.

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

Lab Answer Key: Module 5: Implementing Azure App Service

Lab: Implementing web apps

Exercise 1: Creating web apps

Task 1: Create a web app

1. Ensure that the MSL-TMG1 and 20533C-MIA-CL1 virtual machines are running, and then sign in to 20533C-MIA-CL1 as **Student** with the password **Pa$$w0rd**.

2. Open Internet Explorer, browse to **http://portal.azure.com**, and then sign in using a Microsoft account that is either the Service Admin or co-admin of your subscription.

3. In the top-left corner of the portal, click **New**, and then click **Web+Mobile**.

4. In the Web+Mobile blade, click **Web App**.

5. In the Web App blade, in the ** App** **name** text box, type a unique name. If the name is unique and valid, a green check mark appears.

6. In the Web App blade, in the **Resource Group**, verify that **New** is selected from drop down list, and then in the **New resource group name** text box type **AdatumLabWebRG**.

7. In the Web App blade, click the **App Service plan/Location** link.

8. In the App Service plan blade, click **Create New**.

9. In the **App Service plan** text box, type **WebAppStandardPlan**.

10. In the **Location** drop down list, select a location near you.

11. In **Pricing tier**, select **S1 Standard**, and then click **OK**.

12. In the Web App blade, click **Create**. The web-app creation process may take several minutes.

Task 2: Add a deployment slot

1. On the left of the Azure portal, click **Browse**, and then click **App Services**.

2. In the App Services blade, click the web app that you created in the first task.

3. In the Settings blade, scroll down to locate the **PUBLISHING** section, and then click **Deployment slots**.

4. In the Deployment slots blade, click **Add Slot**.

5. In the Add a slot blade, in the **Name** text box, type **Staging**.

6. In the **Configuration Source** list, select the web app you created in the first task, and then click **OK**. Azure adds the new deployment slot to the list.

7. Close the Deployment slots blade.

8. Open Windows PowerShell by clicking its shortcut in the taskbar.

9. Sign in to the Azure subscription by typing the following command in the Azure PowerShell window, and then pressing Enter:
10. `Login-AzureRMAccount`
11. Sign in to the Azure subscription with a Microsoft account that is either the Service Admin or co-admin of your subscription.
12. If you have multiple subscriptions, to select the target one type the following commands at the each followed by pressing Enter:
13. `Get-AzureRmSubsctiption`
14. `Set-AzureRmContext -SubscriptionName "Name of your subscription"`
15. Type the following PowerShell command, and then press Enter:
16. `Get-AzureRMWebApp`
17. Verify that the list of web apps includes the web app that you created in Task 1.
18. Type the following Azure PowerShell command, and then press Enter:
19. `Get-AzureRMWebAppSlot -ResourceGroupName AdatumLabWebRG -name "Name of your WebApp"`
20. Verify that the web app is listed the staging slot you created in this task.
21. Keep the Azure PowerShell window open.

Task 3: Configure deployment credentials

1. In the Azure portal, in the web app that you created in Task 1, in the Settings blade, scroll down to the Publishing section, and then click **Deployment credentials**.
2. In the Set Deployment credentials blade, in the **FTP/Deployment user name** text box, type **ftpadmin *XXXX (replace***_XXXX_ is a unique number).
3. In the **Password** text box, type **Pa$$w0rd**.
4. In the **Confirm password** box, type **Pa$$w0rd**, and then click **Save**.
5. Close the Set deployment credentials blade.

Result: After completing this exercise, you should have created a new web app in the Azure portal, and configured the new web app with deployment slots and credentials.

Exercise 2: Deploying a web app

Task 1: Obtain a publishing profile

1. In the Azure portal, in the blade that shows the web app that you created in the exercise 1, in the command bar located on the top of the blade, click **Get publish profile**.
2. In the dialog box, click **Save**. Internet Explorer saves the publishing profile in the **Downloads** folder.
3. On the taskbar, click **Visual Studio 2015**.
4. On the **File** menu, click **Open**, and then click **Project/Solution**.
5. Browse to the folder **D:\LabFiles\Lab05\Starter\AdatumWebsite**, click **AdatumWebsite.sln**, and then click **Open**.

6. On the **Debug** menu, click **Start Debugging**.
7. Under **A. Datum Corporation**, click **Learn More**.
8. Click **Contact**.
9. Close Internet Explorer.

Task 2: Deploy a web app

1. In Visual Studio, on the **Debug** menu, click **Stop Debugging**.
2. In the Solution Explorer, right-click the **AdatumWebsite** project, and then click **Publish**.
3. In the Publish Web Wizard, on the **Profile** page, click **Import**.
4. In the **Import Publish Settings** dialog box, click **Browse**.
5. Browse to the **Downloads** folder.
6. Select the **.PublishSettings** file that you downloaded in Task 1 of this exercise, and then click **Open**.
7. In the **Import Publish Settings** dialog box, click **OK**.
8. On the **Connection** page, click **Validate Connection**. Visual Studio connects to the Azure web app. If the connection is valid, a green tick mark appears.
9. Click **Next**.
10. On the **Settings** page, in the **Configuration** drop-down list, select **Release**.
11. Click **Next**.
12. On the **Preview** page, click **Start Preview**.
13. Examine the list of changes to apply to the web app.
14. Click **Publish**.

    Note: The publish operation may take approximately two to three minutes. When the operation is complete, Microsoft Edge opens and displays the new web app hosted in Azure.

15. Verify that A. Datum's web app is open in Microsoft Edge, and then verify the web app's current address.
16. Close the **Home Page** tab.
17. Close Visual Studio.

Result: After completing this exercise, you should have a deployed a web app hosted in Azure that you can open with any common web browser.

Exercise 3: Managing web apps

Task 1: Deploy a web app for staging

1. In Internet Explorer, in the Azure portal, navigate to the web app you created in Exercise 1, Task 1.

2. In the Settings blade, scroll down to the **PUBLISHING** section, and then click **Deployment Slots**.

3. In the Deployment slots blade, click the staging slot ***yourwebapp*-staging** that was created in Exercise 1, Task 2.

4. In the *yourwebapp*(Staging) blade from the command bar located on the top section of the blade, click **Get publish profile**.

5. In the dialog box, click **Save**.

6. On the taskbar, click **Visual Studio 2015**.

7. On the **File** menu, click **Open**, and then click **Project/Solution**.

8. Browse to the folder **D:\LabFiles\Lab05\Starter\NewAdatumWebsite**.

9. Click **AdatumWebsite.sln**, and then click **Open**.

10. In Solution Explorer, right-click the **AdatumWebsite** project, and then click **Publish**.

11. In the Publish Web Wizard, on the **Profile** page, click **Import**.

12. In the **Import Publish Settings** dialog box, click **Browse**.

13. In the **Downloads** folder, select the ***YourWebapp*(Staging).PublishSettings** file, and then click **Open**.

14. In the **Import Publish Settings** dialog box, click **OK**.

15. On the **Connection** page, click **Validate Connection**.

16. If the connection details are correct, a green tick mark appears.

17. Click **Next**.

18. In the **Configuration** drop-down list, ensure that **Release** is selected, and then click **Next**.

19. On the **Preview** page, click **Start Preview**.

20. Examine the files that are to be published, and then click **Publish**.

21. When the publish operation is complete, Microsoft Edge opens and displays the new web app in the staging slot.

22. Close Microsoft Edge and Visual Studio.

Task 2: Swap deployment slots

1. In the Azure portal, click **Browse**, and then click **App Services**.

2. In In the App Services blade, click the web app you created in Exercise 1, Task 1.

3. In the *yourwebapp* blade, under the **Essentials** section, click the **URL** link for your web app. Notice the color scheme has not changed.

4. Close the tab that displays the A. Datum web app.

5. In the Azure portal, in the Settings blade, scroll down to the **PUBLISHING** section, and then click **Deployment slots**.

6. In the Deployment slots blade, on the command bar, click **Swap**.

7. In the Swap blade, in the **Swap type** drop-down list, verify that **Swap** is selected.

8. In the **Source** drop-down list, ensure that **Staging** is selected.

9. In the **Destination** drop-down list, ensure that **production** is selected, and then click **OK**.

10. Wait until swap operation completes.

11. Close all the open blades except *yourwebapp* blade.

12. In the *yourwebapp* blade in Essentials section, click the URL link for your web app. Notice that the color scheme is new.

13. Close the tab that displays the A. Datum's web app.

Task 3: Roll back a deployment

1. In the Azure portal, in the *yourwebapp* blade in the command bar at the top, click **Swap**.

2. In the Swap blade, in the **Swap type** drop-down list, verify that **Swap** is selected.

3. In the **Source** drop-down list, select **Staging**.

4. In the **Destination** drop-down list, select **production**, and then click **OK** button.

5. Wait until Swap operation completes.

6. In the *yourwebapp* blade, in **Essentials** section, click the URL link for your web app.

7. Notice that the color scheme is reverted to the old scheme.

8. Close the A. Datum tab in Internet Explorer.

Result: After completing this exercise, you should have an updated web app staged and published in Azure.

Exercise 4: Implementing Traffic Manager

Task 1: Deploy a web app to another region

1. Switch to Microsoft Azure PowerShell.

2. At the command prompt, type the following command, and then press Enter:

3. `Get-AzureRMWebApp`

Note the name of your original web app and location.

1. Choose an Azure region that is different from the location of the original web app. This will become the "*SecondLocation*"

2. At the command prompt, type the following command to create a new resource group, and then press Enter:

```
3. New-AzureRMResourceGroup -Name AdatumLabWebRG2 -Location "SecondLocation"
```

4. At the command prompt, type the following command to create new App Service Plan, and then press Enter:

```
5. New-AzureRMAppServicePlan -Location "SecondLocation" -Tier Standard -Name StandardPlan
   -ResourceGroupName AdatumLabWebRG2
```

6. At the command prompt, type the following command to create a new web app, and then press Enter:

```
7. New-AzureRMWebApp -ResourceGroupName AdatumLabWebRG2 -Name "WebAppName2" -Location
   "SecondLocation" -AppServicePlan StandardPLan
```

   WebAppName2 is the name of your first web app with the number 2 appended, and *SecondLocation*****is the location you chose in step 4.

8. Switch to the Azure portal in the Internet Explorer window.
9. On the left side of the Azure portal, click **Browse**, and then click **App Services**.
10. In the App Services blade, click **_WebAppName_2**.
11. In the _WebAppName_2 blade, on the command bar located at the top of the blade, click **Get publish profile**.
12. When prompted, click **Save**.
13. On the taskbar, click **Visual Studio 2015**.
14. In the Visual Studio, on the **File** menu, click **Open**, and then click **Project/Solution**.
15. Browse to the folder **D:\LabFiles\Lab05\Starter\AdatumWebsite**.
16. Click **AdatumWebsite.sln**, and then click **Open**.
17. In Solution Explorer, right-click the **AdatumWebsite** project, and then click **Publish**.
18. In the Publish Web Wizard, on the left click **Profile**, and then click **Import**.
19. In the **Import Publish Settings** dialog box, click **Browse**.
20. In the **Downloads** folder, select the **_WebAppName_2.PublishSettings** file, and then click **Open**.
21. In the **Import Publish Settings** dialog box, click **OK**.
22. On the **Connection** page, click **Validate Connection**.
23. If the connection details are correct, a green tick mark appears.
24. Click **Next**.
25. In the **Configuration** drop-down list, ensure that **Release** is selected, and then click **Next**.
26. On the **Preview** page, click **Start Preview**.
27. Examine the files that will be published, and then click **Publish**.

28. When the publish operation completes, Internet Explorer opens and displays the new web app.

29. Close the **Home Page** tab.

30. Close Visual Studio.

Task 2: Create a Traffic Manager profile

1. In Internet Explorer, in the Azure portal, click **New**, and then click **Networking**.

2. In the Networking blade, click **Traffic Manager profile**.

3. In the Create Traffic Manager profile blade, in the **Name** text box, type a unique name. This will be appended with the suffix **trafficmanager.net**. If the name is unique and valid, a green checkmark appears.

4. In the **Routing Method** drop-down list, select **Performance**.

5. In the **Resource Group** drop down list, ensure that **New** is selected .

6. In the **New resource group name** text box, type **AdatumLabTMRG**.

7. In the **Resource group location** drop-down list box, select the Azure region that is closest to your location, and then click **Create**. Wait until the Traffic Manager profile is created.

Task 3: Add endpoints, and configure Traffic Manager

1. In the hub menu on the left side, click **Browse**, and then select **Traffic Manager Profiles**.

2. In the Traffic Manager profiles blade, locate and click your previously created profile.

3. In the Settings blade, click **Endpoints**.

4. In the Endpoints blade, click **Add**.

5. In the Add endpoint blade, in the **Type** drop-down list, select **Azure endpoint**.

6. In the **Name** text box, type the name of your web app, which you created in Exercise 1.

7. In the **Target resource type** drop-down list of websites, select **App Service**.

8. Click **Choose an app service**.

9. In the Resource blade, select the web app that you created in Exercise 1.

10. Click **OK** to add the endpoint.

11. Repeat steps 4 through 10 to add a second endpoint for the web app that you created in Exercise 4.

12. Close the Endpoints blade.

13. In the Settings blade, click the **Configuration** link.

14. In the Configuration blade, in the **DNS time to live (TTL)** text box, remove the original setting, and then type **30**.

15. On the command bar at the top, click **Save**.

16. Close the Configuration blade, and then close the Settings blade.

Task 4: Test Traffic Manager

1. In Internet Explorer, in the Azure portal, in the *Yourname* Traffic Manager blade, click the link under the **DNS name** section.
2. Internet Explorer displays the Adatum web app.
3. On the Start menu, type **cmd**, and then press Enter.
4. Type the following command, and then press Enter:
5. `nslookup dnsname`

where *dnsname* is the DNS name of the traffic manager profile that you accessed in step 1.

1. Note the aliases that are returned.
2. In Internet Explorer, switch to the tab that displays the Azure portal.
3. In the *Yourname**Traffic Manager blade, click the *All settings** link.
4. In the Settings blade, click the Endpoints link.
5. In the Endpoints blade, in the list of endpoints, select the web app that you created in Exercise 1.
6. In the *YourWebApp* blade, click **Edit.** Under **Status**, click **Disabled**, and then click **Save**.
7. Switch to the command prompt, type the following command, and then press Enter:
8. `nslookup dnsname`

Please note *dnsname*****is the DNS name that you used in step 4.

1. Note that the aliases that return are different from those returned in step 4.

Note: If the aliases are not changed, at the command prompt, reissue the **nslookup** commands until there is a change.

Task 5: Reset the Azure 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 box, click **Yes**.
3. Type the following command, and then press Enter:
4. `Reset-Azure`
5. When prompted (twice), sign in by using the Microsoft account associated with your Azure subscription.
6. If you have multiple Azure subscriptions, select the one you want the script to target.
7. When prompted for confirmation, type **y**.

Note: This script may 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 approximately two or three minutes to reset your Azure environment, so that you are ready for the next lab. The script removes all storage, virtual machines, virtual networks, cloud services, and resource groups. **Important:** The script may not have exclusive access to a storage account so that it can delete it. If this occurs, you will see an error. If you find objects remaining after the reset script is complete, you can rerun the **Reset-Azure** script, or use the Azure portal to delete all objects in your Azure subscription manually, with the exception of the default directory. Do not delete it.

Result: After completing this exercise, you should have a web app set up in two Azure regions and Traffic Manager configured to distribute requests between them.

Lab Answer Key: Module 6: Planning and implementing storage, backup, and recovery services

Lab: Planning and implementing Azure Storage

Exercise 1: Creating and configuring Azure Storage

Task 1: Create a storage account

1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa$$w0rd** and that the setup script that you ran in the previous demonstration to prepare the environment has completed.
2. Start Internet Explorer, and then browse to **https://portal.azure.com**. When prompted, sign in by using the Microsoft account that is the Service Administrator or Co-Administrator of your Microsoft Azure subscription.
3. On the **Hub** menu, click **New**, and then click **Data + Storage**.
4. In the Data + Storage blade, click **Storage account**.
5. In the Create storage account blade, apply the following settings, and then click **Create**:
    - Name: Enter a valid, unique name consisting of between 3 and 24 lower case characters or digits.
    - Deployment model: Classic
    - Performance: Standard
    - Replication: Geo-redundant storage (GRS)
    - Subscription: *Your Azure subscription*
    - Resource group: make sure that **+New** appears in the drop down list and then type **Asset-Management** in the **New resource group name** textbox
    - Location: Select the Azure region nearest to you
    - Pin to dashboard: Clear the check box
6. At the top of the portal window menu, click the **Notifications** icon, and then wait for the notification that the storage account has been created.
7. On the **Hub** menu, click **Browse**, and then click **Storage accounts (classic)**.
8. In the Storage accounts (classic) blade, click the storage account that you just created.
9. In the blade for your storage account, click the **Blobs** tile.
10. In the Blob service blade, click **Container** in the toolbar.
11. In the New container blade, apply the following settings, and then click **Create**:

- Name: **asset-images**
- Access type: **Private**

12. Close the Blob service blade.
13. On the taskbar, right-click **Windows PowerShell**, and then click **Run ISE as Administrator**. Click **Yes** when prompted.
14. In the Windows PowerShell Interactive Scripting Environment (ISE), click **File**, and then click **Open**.
15. In the **Open** dialog box, browse to **D:\Labfiles\Lab06\Starter\**, click **ExampleCommands.ps1**, and then click **Open**.
16. If the Script pane is not visible, on the **View** menu, click **Show Script Pane**.
17. In Windows PowerShell ISE, in the Script pane, type the name of the storage account that you created in the previous task.
18. Leave the Internet Explorer window open. You will use it later in this lab.

Task 2: Install AzCopy

1. In Internet Explorer, open a new tab, and then browse to **http://aka.ms/AzCopy**.
2. In the Download and install AzCopy section, click the link to **Download the latest version of AzCopy**.
3. When prompted to run or save the file, click **Run**. Then click **Yes** if prompted to allow the program to make changes to the computer, and then complete the wizard to install AzCopy by using the default installation options.
4. Right-click **Start**, click **System**, and then in the System window, click **Advanced system settings**.
5. In the **System Properties** dialog box, on the **Advanced** tab, click **Environment Variables**.
6. In the **Environment Variables** dialog box, in the **System variables** list, select **Path**, and then click **Edit**.
7. In the **Edit environment variable** dialog box, click **New**.
8. In the text box, type **C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy**, and then click **OK**.
9. In the **Environment Variables** dialog box, click **OK**.
10. In the **System Properties** dialog box, click **OK**, and then close the System window.
11. Right-click **Start**, click **Command Prompt (Admin)**, and then click **Yes** when prompted.
12. At the command prompt, type the following command, and then press Enter:
13. `AzCopy /?`
14. View the syntax information that displays. Leave the Command Prompt window open for the next task.

Task 3: Use AzCopy to upload blobs

1. In Internet Explorer, on the **Microsoft Azure** tab, in the blade for your storage account, click the **Keys** icon.

2. On the Manage keys blade, click the **Copy** icon next to the primary access key. If prompted to allow access to the Clipboard, click **Allow access**.

3. In the Command Prompt window, enter the following commands to change the current directory context:

4. `D:`
5. `CD D:\Labfiles\Lab06\Starter`
6. Switch to Windows PowerShell ISE.

7. In Windows PowerShell ISE, in the Script pane, locate the following code:

8.
   ```
   AzCopy /Dest:https://<your storage account>.blob.core.windows.net/asset-images
   /destkey:<your primary access key> /Source:asset-images
   ```
9. Replace < *your storage account*> with your storage account name.

10. Replace with your primary access key.

11. In Windows PowerShell ISE, in the Script pane, select the code that you just edited. Click **Edit**, and then click **Copy**.

12. Switch to the Command Prompt window.

13. In the Command Prompt window, click the control box at the top left of the window, point to **Edit**, click **Paste**, and then press Enter to run the command.

14. Wait for the command to complete, and then view the file transfer information that displays.

15. Close the Command prompt window.

Result: At the end of this exercise, you should have created a new Azure storage account with a container named "asset-images."

Exercise 2: Using Azure File storage

Task 1: Create a file share and upload files

1. Switch to Windows PowerShell ISE.

2. Click **File**, and then click **Open**.

3. In the **Open** dialog box, browse to **D:\Labfiles\Lab06\Starter\**, click **FileShare.ps1**, and then click **Open**.

4. In Windows PowerShell ISE, in the command prompt pane, enter the **Get-AzureAccount** command, and then verify that your Microsoft account displays.

   Note: If your account does not display, enter the **Add-AzureAccount** command, and then sign in by using your Microsoft account.

5. In the script pane, in the *$storageAccountName* variable declaration at the beginning, replace the <*your_storage_account_name*> value with the name of the Azure storage account that you created in the previous task.

6. Review the script, noting that it:

   - Declares variables named *$shareName* and *$folderName* for the file share and the folder to create.
   - Uses the **Get-AzureStorageKey** cmdlet to retrieve the access key for your storage account.
   - Uses the **New-AzureStorageContext** cmdlet to create a storage context that connects to your storage account by using the access key.
   - Uses the **New-AzureStorageShare** cmdlet to create a share.
   - Uses the **New-AzureStorageDirectory** cmdlet to create a folder in the share.
   - Finds the folder where the script is stored, and then declares a variable named *$sourceFolder* that references the invoices subfolder.
   - Iterates through the files in the source folder, and then uses the **Set-AzureStorageFileContent** cmdlet to write each file to the folder in the file share.

7. Save the script, and then on the toolbar, click **Run Script**.

8. Observe the script as it runs, and then view the output. When you finish, close Windows PowerShell ISE without saving any changes.

Task 2: Access a file share from a VM

1. In Internet Explorer, on the Azure portal, click **Browse**, click **Virtual machines (classic)**, and then in the Virtual machines (classic) blade, click **AdatumSvr1**.

2. In the AdatumSvr1 blade, click **Connect**, and then when prompted to open or save the AdatumSvr1.rdp file, click **Open**.

3. When prompted to connect, click **Connect**, enter the following credentials, and then click **OK**:

   - User name: **AdatumSvr1\Student**
   - Password: **Pa$$w0rd123**

4. If prompted to connect again, click **Yes**, and then wait for the remote desktop session to open and initialize. If you are promoted to find PCs, devices, and content on this network, click **No**.

5. When Server Manager starts, on the **Local Server** page, click the status for **IE Enhanced Security Configuration**, select **Off** for Administrators, and then click **OK**.

6. Close Server Manager.

7. In the AdatumSvr1 remote desktop window, switch to the **Start** page, and then click **Internet Explorer**. If prompted to set up Internet Explorer, select **Use recommended security, privacy, and compatibility settings**, and then click **OK**.

8. Browse to **https://portal.azure.com**, and then sign in by using the Microsoft account that is the Service Administrator or Co-Administrator of your Azure subscription.

9. On the **Hub** menu, click **Browse**, and then click **Storage accounts (classic)**.

10. In the Storage accounts (classic) blade, click the storage account that you created in the previous exercise, and then in the blade for your storage account, click the **Keys** icon.

11. On the Manage Keys blade, click the **Copy** icon next to the primary access key. If prompted to allow access to the Clipboard, click **Allow access**.

12. Right-click **Start**, and then click **Command Prompt (Admin)**.

13. In the Command Prompt window, enter the following command to map a network drive to the assets file share in your Azure storage account. Replace both instances of *storage_account* with the name of your storage account, and then press Enter:

14. `net use z: \\storage_account.file.core.windows.net\assets /u:storage_account`
15. When prompted, paste the *access_key* from the Clipboard (to paste into a Command Prompt window, click the control box at the top left of the window, point to **Edit**, and then click **Paste**).

16. At the command prompt, enter the following command to view the contents of the invoices folder in drive Z, which is now mapped to the assets file share that you created in a previous task:

17. `dir z:\invoices`
18. Verify that three invoice files are listed.

19. Close the Command Prompt window and Internet Explorer, and then sign out of the remote desktop session to AdatumSvr1.

Result: At the end of this exercise, you should have created a file share named "assets" that contains a folder named "invoices." This folder will contain three invoice documents and will be accessible on the AdatumSvr1 virtual machine (VM).

Exercise 3: Protecting data with Azure Backup

Task 1: Create a backup vault

1. On the MIA-CL1, open Internet Explorer, and then browse to **https://manage.windowsazure.com**.

2. If prompted, sign in by using the Microsoft account that is the Service Administrator or Co-Administrator of your Azure subscription.

3. In the Azure classic portal, click **NEW**, click **DATA SERVICES**, click **RECOVERY SERVICES**, click **BACKUP VAULT**, and then click **QUICK CREATE**.

4. Enter a valid, unique name, select your closest region, and then click **CREATE VAULT**.

Task 2: Obtain vault credentials

1. On the Azure classic portal, click **RECOVERY SERVICES** on the **Hub** menu, and then click your new backup vault.
2. On the backup vault **Quick Start** page, click **Download vault credentials**.
3. Click **Save** to download the vault credentials to the Downloads folder.
4. After the credentials download, you will be prompted to open the folder. Click the prompt window.

Task 3: Install and configure the Azure Backup agent

1. On the Azure classic portal, on the **Quick Start** page of your backup vault, under Download Azure Backup Agent, click the **Agent forWindows Server or System Center Data Protection Manager or Windows Client** link.
2. When prompted to run or save the file, click **Run**. When prompted to allow the program to make changes, click **Yes**, and then complete the wizard to install the agent. Use the default installation options, and if prompted, choose the option to use Microsoft Update to check for updates.
3. When installation is complete, click **Close**.
4. Minimize all active windows, and then on the desktop, double-click **Microsoft Azure Backup**. When prompted to allow the program to make changes, click **Yes**.
5. In the Microsoft Azure Backup window, in the Actions pane, click **Register Server**.
6. In the Register Server Wizard, on the **Proxy Configuration** page, click **Next**.
7. On the **Vault Identification** page, click **Browse**, navigate to the **Downloads** folder, select the credentials that you created earlier, and then click **Open**.
8. On the **Vault Identification** page, click **Next**.
9. On the **Encryption Setting** page, click **Generate Passphrase**, click **Browse**, browse to the **D:\Labfiles\Lab06\Starter** folder, and then click **OK**.
10. Click **Register**, and then when registration is complete, click **Close**.
11. Leave Azure Backup open for the next task.

Task 4: Create a backup schedule

1. In Azure Backup, in the Actions pane, click **Schedule Backup**.
2. In the Schedule Backup Wizard, on the **Getting started** page, click **Next**.
3. On the **Select Items to Backup** page, click **Add Items**.
4. In the **Select Items** dialog box, expand **D**, expand **Labfiles**, expand **Lab06**, expand **Starter**, select the following folders, and then click **OK**:
    - asset-images
    - invoices

5. On the **Select Items to Backup** page, click **Next**.

6. On the **Specify Backup Schedule** page, in the first drop-down list box below the **At following times (Maximum allowed is three times a day)** box, select **4:30 AM**, and then click **Next**.

7. On the **Select Retention Policy** page, accept the defaults, and then click **Next**.

8. On the **Choose Initial Backup type** page, accept the defaults, and then click **Next**.

9. On the **Confirmation** page, click **Finish**. When the backup schedule is created, click **Close**.

Task 5: Run a backup

1. In Azure Backup, in the Actions pane, click **Back Up Now**.

2. In the Back Up Now Wizard, on the **Confirmation** page, click **Back Up**.

3. When the backup is complete, click **Close**, and then close Azure Backup.

4. In Internet Explorer, on the Azure portal, on the page for your backup vault, click **REGISTERED ITEMS**.

5. In the **TYPE** drop-down list box, select **Windows server**, click the **check mark** on the right side, and then verify that the MIA-CL1 server lists as registered.

6. Click **PROTECTED ITEMS**.

7. In the **TYPE** drop-down list box, select **Files and Folders**, click the **check mark** on the right side, and then verify that drive D of MIA-CL1 lists as protected.

Task 6: 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 box, click **Yes**.

3. Type the following command, and then press Enter:

4. `Reset-Azure`

5. When prompted (twice), sign in by using the Microsoft account that is associated with your Azure subscription.

6. If you have multiple Azure subscriptions, select the one that you want to target with the script.

7. When prompted for confirmation, press Y and press Enter.

Note: This script will remove Azure services in your subscription. We therefore recommended 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 Azure environment, ready for the next lab. The script removes all storage, virtual machines, virtual networks, cloud services, and resource groups. **Note: Important**: The script might not be able to access a storage account to delete it (if this occurs, you will see an error). If you find objects remaining after the reset script is complete, you can rerun the **Reset-Azure** script, or you can use the Azure portal and the Azure classic portal to delete all the objects in your Azure subscription manuallyâ€”with the exception of the default directory.

Result: At the end of this exercise, you should have created an Azure Backup vault in your subscription, created Azure Backup vault credentials, and installed the Azure Backup agent on the MIA-CL1 lab computer. You should have backed up the contents of the asset-images and invoices folders to the backup vault.

Lab Answer Key: Module 7: Planning and implementing Azure SQL Database

Lab: Planning and implementing Azure SQL Database

Exercise 1: Creating, securing, and monitoring an Azure SQL Database

Task 1: Create an Azure SQL Database

1. Sign in to the MIA-CL1 lab virtual machine as **Student** with the password **Pa$$w0rd**.

2. Start Internet Explorer, browse to **https://portal.azure.com**, and sign in by using the Microsoft account that is the Subscription Administrator or Co-Administrator of your Azure subscription.

3. In the Hub menu on the left, click **New**, click **Data + Storage**, and then click **SQL Database**.

4. On the **SQL database** blade, in the **Database name** box, type **operations**.

5. Click **Server**, and then on the **Server** blade, click **Create a new server**.

6. On the **New server** blade, enter the following settings and then click **Select**:

    - Server name: any valid unique name
    - Server admin login: **Student**
    - Password: **Pa$$w0rd**
    - Confirm password: **Pa$$w0rd**
    - Location: the closest Azure region (to your location)
    - Create V12 server (Latest update): **Yes**
    - Allow azure services to access server: Enabled

7. Ensure that **Blank database** is selected as the Select source.

8. Click **Pricing tier**.

9. On the **Choose your pricing tier** blade, select **S1 Standard** and click Select.

10. On the **SQL database** blade, in the **Resource group drop down list, select +New and then in the New resource group name** text box, type **OpsRG**.

11. On the **SQL database** blade, ensure that **Pin to dashboard** is selected and click **Create**. Then wait for the SQL Database to be created.

12. After the database is created, the portal will automatically display its **Settings** blade.

Task 2: Configure server firewall rules

1. Navigate to the **operations** blade in the Azure portal in the Internet Explorer window.

2. On the **operations** blade, click the hyperlink containing the server name.
3. Navigate to the **Settings** blade of the server and click **Show firewall settings**.
4. On the **Firewall settings** blade, note the value of the **Client IP address** entry.
5. On the **Firewall settings** blade, specify the following:
    - RULE NAME: **AllowLabVM**
    - START IP: **_XXX.XXX._0.0**
    - END IP: ***XXX.XXX*.255.255**

Note: where ***XXX.XXX*** represents the first two octets of the value of the **Client IP address** entry.

1. Click **Save**

Task 3: Use SQL Server Management Studio

1. Start SQL Server 2014 Management Studio, and in the **Connect to Server** dialog box, specify the following settings (replacing *server_name* with the unique name you specified when creating your SQL Database server), and click **Connect**:
    - Server type: **Database Engine**
    - Server name: **_server_name._database.windows.net**
    - Authentication: **SQL Server Authentication**
    - Login: **Student**
    - Password: **Pa$$w0rd**
2. In SQL Server Management Studio, in the Object Explorer, under the server name expand **Databases**, and verify that the **operations** database is listed.
3. In SQL Server Management Studio, navigate to the D:\Labfiles\Lab07\Starter folder, open the **Operations.sql** file and view the Transact-SQL code it contains.
4. On the toolbar, in the **Available Databases** list, select **operations**. Click **Execute**
5. Click **New Query** and enter the following Transact-SQL code in the new query pane:
6. `SELECT * FROM dbo.serverlist;`
7. On the toolbar, in the **Available Databases** list, ensure that **operations** is selected. Then click **Execute**.
8. View the query results and verify that a list of three servers and their IP addresses is returned.
9. Keep SQL Server Management Studio and Internet Explorer open.

Task 4: View database metrics

1. In Internet Explorer, in the Azure portal, navigate to the **operations** SQL Database blade.
2. On the **operations** blade, note the charts displayed in the **Monitoring** section, which show resource utilization in terms of DTU percentage.

3. Click **Edit**, in the **Resource utilization** chart, deselect **DTU percentage** and click **Total database size**, and then click **OK**.
4. Click the chart. This will display the **Metric** blade.
5. On the **Metric** blade, click **Add alert**. Then, on the **Add an alert rule** blade, specify the following settings and click **OK**:
   - Resource: leave the default setting in place
   - Name: **operations storage alert**
   - Description: **storage alert for operations database**
   - Metric: **total database size**
   - Condition: **greater than**
   - Threshold: **1024**
   - Period: **over the last 5 minutes**
   - Email owners, contributors, and readers: selected
   - Additional administrator email(s): any email address
   - Webhook: leave blank
6. Keep Internet Explorer open for the next exercise

Result: After completing this exercise, you should have created an Azure SQL Database named operations on a new server with a name of your choosing. You should also have used SQL Server Management Studio to create a table named dbo.serverlist and created an alert to help you monitor database storage.

Exercise 2: Migrating a Microsoft SQL Server Database to Azure SQL Database

Task 1: Deploy a database to Azure

1. In SQL Server Management Studio, in Object Explorer, in the **Connect** drop-down list, click Database Engine.
2. In the **Connect to Server** dialog box, specify the following settings, and click **Connect**:
   - Server type: **Database Engine**
   - Server name: **MIA-CL1**
   - Authentication: **Windows Authentication**
3. In SQL Server Management Studio, in Object Explorer, under the **MIA-CL1** server, expand **Databases** and verify that the **sales** database is listed.
4. Right-click the **sales** database, point to **Tasks**, and then click **Deploy Database to Windows Azure SQL Database**.

5. In the **Deploy Database "sales"** wizard, on the **Introduction** page, click **Next**.

6. On the **Deployment Settings** page, click **Connect**. Then in the **Connect to Server** dialog box, specify the following settings (replacing ***server_name*** with the unique name of your SQL Database server) and click **Connect**:

   - Server type: **Database Engine**
   - Server name: ***server_name*.database.windows.net**
   - Authentication: **SQL Server Authentication**
   - Login: **Student**
   - Password: **Pa$$w0rd**

7. On the **Deployment Settings** page, ensure that the new database name is **sales** and note the temporary file name used for the .bacpac file that will be exported and imported, ensure that the **Service Objective is set** to **S2,** and then click **Next**..

8. On the **Summary** page, click **Finish**.

9. On the **Results** page, verify that the operation completed successfully, and click **Close**.

10. In SQL Server Management Studio, in Object Explorer, if necessary, right-click the **Databases** folder under your Azure SQL Database server and click **Refresh** to verify that the sales database has been copied to this server.

Task 2: Configure SQL Database security

1. In SQL Server Management Studio, in Object Explorer, under your Azure SQL Database server, expand **Security**, expand **Logins**, and verify that only the **Student** login is listed.

2. Right-click **Logins** and click **New Login**. Then, replace the auto-generated Transact-SQL script that is generated as shown here and then click **Execute**:

```
3.  CREATE LOGIN SalesApp
4.  WITH PASSWORD = 'Pa$$w0rd'
5.  GO
```

6. In Object Explorer, right-click the **Logins** folder and click **Refresh** to verify that the **SalesApp** login has been created.

7. In Object Explorer, in the **Databases** folder for your Azure SQL Database server, expand the **sales** database, expand **Security**, and expand **Users**.

8. Right-click **Users** and click **New User**. Then, modify the Transact-SQL script that is generated as shown below and then click **Execute**:

```
9.  CREATE USER SalesApp
10. FOR LOGIN SalesApp
11. WITH DEFAULT_SCHEMA = dbo
12. GO
13. EXEC sp_addrolemember 'db_owner', 'SalesApp'
14. GO
```

15. In Object Explorer, right-click the **Users** folder and click **Refresh** to verify that the **SalesApp** user has been created.

16. Keep SQL Server Management Studio open for the next exercise.

Task 3: Configure an application connection string

1. Start Visual Studio, navigate to the **D:\Labfiles\Lab07\Starter** folder and open the **SalesApp.sln** solution.

2. In Solution Explorer, double-click **Web.config**.

3. In Web.config, note that the **SalesConnectionString** element contains a **connectionString** attribute that connects to the **sales** database on the **localhost** server using integrated security (Windows authentication).

4. In Internet Explorer, on the tab containing the preview Azure portal, on the Hub menu, click **Browse** and then click **SQL databases**.

5. On the **SQL databases** blade, click the **sales** database.

6. On the **sales** blade, under Essentials, click **Show database connection strings**.

7. On the **Database connection strings** blade, click the **Click to copy** icon for the **ADO.NET** connection string. If prompted, click **Allow access**.

8. Minimize Internet Explorer, you will return to it in the next exercise.

9. In Visual Studio, in Web.config, select the existing value for the **connectionString** attribute and then paste the connection string you copied to replace it.

10. In the pasted connection string, set the value of the **User ID** parameter to **SalesApp@*server_name*** (where ***server_name*** is the unique name of your Azure SQL Database server). Next, set the value of the Password parameter to **Pa$$w0rd** (by replacing the **{your_password_here}** placeholder. The new connectionString value should look similar to this (on a single line):

```
11. Server=tcp:server_name.database.windows.net,
12. 1433;Database=sales; User ID=SalesApp@server_name;
13. Password=Pa$$w0rd;Encrypt=True;
14. TrustServerCertificate=False;
15. Connection Timeout=30;
```

16. Save Web.config. Then on the **Debug** menu, click **Start Debugging**.

17. When Internet Explorer opens, verify that the sales application shows invoice history data for the selected customer. The data is retrieved from the sales database you migrated to Microsoft Azure SQL Database.

18. Close the Internet Explorer window that contains the **Customer Invoice History** page, ensure that Visual Studio debugger is stopped, and then close Visual Studio, saving changes if prompted.

Result: After completing this exercise, you should have deployed the sales SQL Server database on the local SQL Server instance to your Azure SQL Database server, and configured the SalesApp web application to use a connection string for the new Azure SQL Database.

Exercise 3: Restoring a database

Task 1: Delete a database

1. In Internet Explorer, in the Azure Portal, in the Hub menu, click **Browse**, and then click **SQL Database**.

2. On the **SQL Databases** blade, click the **operations** database.

3. On the **operations** blade, click **Restore**.

4. On the **Restore** blade, verify whether a restore point is available. If not, wait until that is the case.

5. On the **operations** blade, click **...More** and then click **Delete**.

6. When prompted to confirm, click **Yes**.

7. In SQL Server Management Studio, in Object Explorer, under your Azure SQL Database server, right-click the **Databases** folder and click **Refresh** to verify that the **operations** database is no longer on the server.

Task 2: Restore a deleted database

1. In the Hub menu of the Azure Portal, click **Browse**, select **SQL Servers**, and click the name of the SQL server where the **operations** database was created.

2. On the **SQL server** blade, scroll down to the **Operations** section and click **Deleted databases**.

3. On the **Deleted databases** blade, click **operations**.

4. On the **Restore** blade, set the database name to **operations**. Notice that you are restoring the most recent restore point to the same server.

5. Click **OK**.

6. Wait for the restore operation to complete by monitoring **Notifications** area in the portal or on the **Audit Logs** blade (this can take several minutes).

   Note: If the initial restore attempt fails, try again.

7. In SQL Server Management Studio, in Object Explorer, under your Azure SQL Database server, right-click the **Databases** folder and click **Refresh** to verify that the operations database has been restored.

8. In SQL Server Management Studio, click **New Query** and enter the following Transact-SQL code in the new query pane:

9. `SELECT * FROM dbo.serverlist`
10. On the toolbar, in the **AvailableDatabases** list, ensure that **operations** is selected and then click **Execute**.

11. View the query results and verify that a list of three servers and their IP addresses is returned.

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 box, click **Yes**.

3. Type the following command, and then press Enter:

```
4. Reset-Azure
```

5. When prompted (twice), sign in using the Microsoft account associated with your Azure subscription.

6. If you have multiple Azure subscriptions, select the one you want to target by the script.

7. When prompted for confirmation, type **y**.

Note: This script will remove Azure services in your subscription. We, therefore, 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, ready for the next lab. The script removes all storage, VMs, virtual networks, cloud services, and resource groups

Important: The script might not be able to get exclusive access to a storage account to delete it (if this occurs, you will see an error). If you find objects remaining after the reset script is complete, you can re-run the **Reset-Azure** script, or use the Azure portal and Azure classic portal to manually delete all the objects in your Azure subscriptionâ€”with the exception of the default directory.

Task 4: To prepare for the next module

Leave the virtual machines running for the next module.

Result: After completing this exercise, you should have deleted and restored the operations database.

Lab Answer Key: Module 8: Implementing PaaS cloud services

Lab: Implementing PaaS cloud services

Exercise 1: Deploying a PaaS cloud service

Task 1: Create a linked resource for a PaaS cloud service

1. Ensure that the MIA-CL1 lab virtual machine is running, and then sign in as **Student** with the password **Pa$$w0rd**.

2. Start Windows PowerShell as an Administrator.

3. Type the following command, and then press Enter.

4. `Add-AzureAccount`
5. Sign in with the user credentials associated with your Azure account.

6. Type the following command, and then press Enter.

7. `Get-AzureLocation | Select-Object Name`
8. From the list of Azure regions, identify the one closest to your location, and then note the region's name.

9. Type the following command, and then press Enter.

10. `New-AzureSqlDatabaseServer -AdministratorLogin 'yourname' -AdministratorLoginPassword 'Pa$$w0rd' -Location 'YourRegion'`

Replace *yourname* with your first name and *Your Region* with the Azure region you noted in step 6.

1. Type the following command, and then press Enter.

2. `Get-AzureSqlDatabaseServer`
3. Note the server name of the Azure SQL Database server you created in step 7.

4. Launch Windows Internet Explorer, navigate to https://portal.azure.com, and then sign in with the service administrator account of your Azure subscription.

5. In the left navigation bar, click **Browse**, and then in the blade that is being displayed, click **SQL databases**.

6. In the SQL Database blade, click **Add**.

7. In the **Database name** box, type **CloudServiceProdDB**.

8. Ensure that the name of the server you created in step 7 appears in the **Server** entry.

9. In the SQL Database blade, click **Create**.

10. Switch to Windows PowerShell, type the following command, and then press Enter.

```
New-AzureStorageAccount -StorageAccountName 'cloudappprodxxx' -Location 'Your Region'
```

Replace *xxx* with a unique sequence of characters (digits or lowercase letters), and replace *Your region* with the Azure region you noted in step 6. If the cmdlet fails because the storage account name you chose is already in use, try a different one.

To test if the storage account name is already in use, type the following command, and then pressing Enter.

```
Test-AzureName -Storage 'cloudappprodxxx'
```

Replace *xxx* with a unique sequence of characters (digits or lowercase letters). An output of **False** indicates that the name has not been assigned yet and is available for you to use.

Task 2: Configure the service definition file

1. On the Taskbar, click **Visual Studio 2015**.
2. Click **File**, click **Open**, and then click **File**.
3. Browse to D:\LabFiles\Lab08\Starter\ Production\Package.
4. Click **ServiceConfiguration.Cloud.cscfg**, and then click **Open**.
5. Locate the **<Role>** element with the name set to **AdatumAdsWebRole**.
6. Within that **<Role**> element, locate the **<Instances>** element.
7. In the **<Instances>** element, set the **count** attribute to **2**.
8. Locate the **<Role>** element with the name set to **AdatumAdsWorkerRole**.
9. Within that **<Role**> element, locate the **<Instances>** element.
10. In the **<Instances>** element, set the **count** attribute to **2**.
11. Launch Internet Explorer, navigate to https://manage.windowsazure.com, and then if prompted, sign in with the service administrator account of your Azure subscription.
12. In the Azure classic portal, in the navigation bar on the left, click **STORAGE**.
13. On the **storage** page, in the list of storage accounts, click **cloudappprod***xxx*.
14. In the command bar at the bottom, click **MANAGE ACCESS KEYS**.
15. To the right of the **PRIMARY ACCESS KEY** box, click **Copy**, and then click **Allow access**.
16. In the Manage Access Keys window, click **OK**.
17. Click the large left arrow at the upper-left side of the window.
18. Switch to Microsoft Visual Studio.
19. In the ServiceConfigurationCloud.cscfg file, locate the **<Role>** element with the name **AdatumAdsWebRole**.
20. Within that **<Role>** element, locate the **<Setting>** element with the name set to **StorageConnectionString**.
21. Delete the string in the **value** attribute, leaving the leading and trailing quotation marks, and then type the following text in its place (on a single line).
22. `DefaultEndPointsProtocol=https;AccountName=cloudappprodxxx;AccountKey=`
23. In the preceding text, **cloudappprod***xxx* is the name of the storage account you created in the previous task.

24. Place the cursor at the end of the text you just typed, and then press Ctrl+V to paste the storage account primary key.

25. Locate the **<Role>** element with the name **AdatumAdsWorkerRole**.

26. Within that **<Role>** element, locate the **<Setting>** element with the name **StorageConnectionString**.

27. Delete the string in the **value** attribute, leaving the leading and trailing quotation marks, and then type the following text in its place (on a single line).

28. `DefaultEndPointsProtocol=https;`
29. `AccountName=cloudappprodxxx;AccountKey=`
30. In the preceding text, **cloudappprod**_xxx_****is the name of the storage account you created in the previous task.

31. Place the cursor at the end of the text you just typed, and then press Ctrl+V to paste the storage account primary key.

32. Locate the **<Role>** element with the name **AdatumAdsWebRole**.

33. Within that **<Role>** element, locate the **<Setting>** element with the name set to **Microsoft.WindowsAzure. Plugins.Diagnostics.ConnectionString**.

34. Delete the string in the **value** attribute, leaving the leading and trailing quotation marks, and then type the following text in its place (on a single line).

35. `DefaultEndPointsProtocol=https;`
36. `AccountName=cloudappprodxxx;AccountKey=`
37. In the preceding text, **cloudappprod**_xxx_ is the name of the storage account you created in the previous task.

38. Place the cursor at the end of the text you just typed, and then press Ctrl+V to paste the storage account primary key.

39. Locate the **<Role>** element with the name **AdatumAdsWorkerRole**.

40. Within that **<Role>** element, locate the **<Setting>** element with the name set to **Microsoft.WindowsAzure. Plugins.Diagnostics.ConnectionString**.

41. Delete the string in the **value** attribute, leaving the leading and trailing quotation marks, and then type the following text in its place (on a single line).

42. `DefaultEndPointsProtocol=https;`
43. `AccountName=cloudappprodXXX;AccountKey=`
44. In the preceding text, **cloudappprod**_xxx_ is the name of the storage account you created in the previous task

45. Place the cursor at the end of the text you just typed, and then press Ctrl+V to paste the storage account primary key.

46. Switch to the Internet Explorer window displaying the Azure portal.

47. In the CloudServiceProdDB blade, click **Show database connection strings.**

48. In the Database connection strings blade, click **Copy** next to the ADO.NET box, and then when prompted, click **Allow access.**

49. Switch back to Visual Studio.

50. Locate the **<Role>** element with the name **AdatumAdsWorkerRole**.

51. Within that **<Role>** element, locate the **<Setting>** element with the name set to **AdatumAdsDbConnectionString**.

52. Delete the string in the **value** attribute, leaving the leading and trailing quotation marks.

53. Press Ctrl+V to paste the connection string you copied to the Clipboard.

54. In the connection string you just pasted, locate the text **{your_username_here}**.

55. Delete the located text, and then replace it with the administrator login username specified in Task 1 **your_name**.

56. In the connection string you just pasted, locate the text **{your_password_here}**.

57. Delete the located text, and then replace it with **Pa$$w0rd**.

58. Click **File**, and then click **Save ServiceConfiguration.Cloud.cscfg**.

Task 3: Deploy the PaaS cloud service

1. In the Internet Explorer window displaying the Azure classic portal, in the left navigation bar, click **CLOUD SERVICES**.

2. On the toolbar at the bottom, click **NEW**, and then click **CUSTOM CREATE**.

3. In the **URL** box, type your name followed by today's date in the MMDDYY format. If a green check mark does not appear, try another name.

4. In the **REGION OR AFFINITY GROUP** drop-down list, select the same Azure region you used in task 1.

5. Select the **Deploy a cloud service package** check box, and then click **Next**.

6. In the **DEPLOYMENT NAME** box, type **AdatumAdsProd**.

7. Next to the **PACKAGE** box, click **FROM LOCAL**.

8. Browse to D:\LabFiles\Lab08\Starter\Production\Package.

9. Click **AdatumAds.cspkg**, and then click **Open**.

10. Next to the **CONFIGURATION** box, click **FROM LOCAL**.

11. Click **ServiceConfiguration.Cloud.cscfg**, and then click **Open**.

12. Click **Complete**.

Note: The deployment process for the platform as a service (PaaS) cloud service can take several minutes to complete. Watch the **cloud services** page. Wait for the **Service Status** column to display **Created** and the **Production** column to display **Running** before you continue to the next task.

Result: You created a storage account and a SQL database, edited the service configuration file, and deployed the cloud service to the production slot.

Exercise 2: Configuring deployment slots and RDP

Task 1: Perform a staged deployment of a PaaS cloud service

1. In the Azure classic portal in the Internet Explorer window, on the **cloud services** page, in the list of cloud services, click the name of the PaaS cloud service you created in the first exercise.
2. Under **Deployment settings**, click **New staging deployment**.
3. In the Upload a package window, in the **DEPLOYMENT LABEL** box, type **AdatumAdsStage**.
4. To the right of the **PACKAGE** box, click **FROM LOCAL**.
5. Browse to D:\LabFiles\Lab08\Starter\Staging\Package.
6. Click **AdatumAds.cspkg**, and then click **Open**.
7. To the right of the **CONFIGURATION** box, click **FROM LOCAL**.
8. Browse to D:\LabFiles\Lab08\Starter\Production\Package.
9. Click **ServiceConfiguration.Cloud.cscfg**, and then click **Open**.
10. Click **OK**.
11. Click the large arrow pointing to the left to return to the **cloud services** page.

Note: The deployment process for the PaaS cloud service can take several minutes to complete. Watch the **cloud services** page. Wait for the **Staging** column to display **Running** before you continue to the next task.

Task 2: Configure RDP access

1. In the Azure classic portal in the Internet Explorer window, on the **cloud services** page, in the list of cloud services, click the name of the PaaS cloud service you created in the first exercise.
2. Click **CONFIGURE**, and then ensure that the **PRODUCTION** deployment is displayed. If not, click the **PRODUCTION** tab.
3. On the toolbar at the bottom, click **REMOTE**.
4. With **(All)** in the **ROLE** drop-down list, select the **Enable Remote Desktop** check box.
5. In the **USER NAME** box, type **RDPAdmin**.
6. In the **NEW PASSWORD** box, type **Pa$$w0rd**.
7. In the **CONFIRM PASSWORD** box, type **Pa$$w0rd**.
8. In the **EXPIRES ON** box, select a date one month from today's date.
9. Click **Complete**.
10. Wait until the configuration operation is complete.

Task 3: Test connectivity

1. In the Azure classic portal in the Internet Explorer window, on the **CONFIGURE** tab of the PaaS cloud service you created in the first exercise, click **DASHBOARD**, and then ensure that the **PRODUCTION** deployment is displayed. If not, click the **PRODUCTION** tab.
2. Under **quick glance**, click **SITE URL**. The cloud service home page opens in a new Internet Explorer tab.
3. Leave the new Internet Explorer tab open. You will use it in the next exercise.
4. On the cloud service dashboard, click **STAGING**.
5. Under **quick glance**, click **SITE URL**. The cloud service staging home page opens in a new Internet Explorer tab.
6. Close the new Internet Explorer tab.
7. At the top of the portal, click **INSTANCES**, and then click **PRODUCTION**.
8. In the list of instances, click **AdatumAdsWebRole_IN_0**.
9. On the toolbar at the bottom, click **CONNECT**, and then click **Open**.
10. In the **Remote Desktop Connection** dialog box, click **Connect**.
11. In the **Password** box, type **Pa$$w0rd**, and then click **OK**.
12. In the **Remote Desktop Connection** dialog box, click **Yes**. The Remote Desktop Protocol (RDP) client displays the desktop for the first instance of the web role.
13. Close the remote desktop connection.
14. In the Remote Desktop Connection window, click **OK**.

Result: At the end of this exercise, you will be able to:Perform a staging deployment of a PaaS cloud service.Enable RDP access to a PaaS cloud service.Connect to production and staging instances via HTTP and via RDP.

Exercise 3: Monitoring cloud services

Task 1: Add metrics to the PaaS cloud service monitoring

1. In the Azure classic portal in the Internet Explorer window, on the **INSTANCES** tab of the PaaS cloud service you created in the first exercise, click **MONITOR**. and then click **PRODUCTION**.
2. On the toolbar at the bottom, click **ADD METRICS**.
3. Expand the **NETWORK IN** section.
4. Select Role Name **AdatumAdsWebRole** with Scope **Aggregate**.
5. Click **Yes**.
6. In the list of metrics, select the **Network In** metric for the **AdatumAdsWebRole** role.

7. To the left of the metric, click the circle to add the metric to the graph.

Task 2: Create an alert

1. In the list of metrics, select the **Network In** metric for the aggregate of the AdatumAdsWebRole role.
2. In the command bar at the bottom, click **ADD RULE**.
3. In the **NAME** box, type **Network In Alert**, and then click **Next**.
4. In the **THRESHOLD VALUE** box, type **1**.
5. Under **ACTIONS**, select **Specify the email address for another administrator**.
6. In the **ADDRESS** box, type the email address of the service administrator account of your Azure subscription.
7. Click **Complete**.
8. After the rule has been created, click **DISMISS COMPLETED**.
9. Switch to the Internet Explorer tab showing the **PRODUCTION** deployment of the PaaS cloud service. Refresh the page several times.

Note: It might take a few minutes before the alert is triggered.

Task 3: Monitor an active cloud service

1. In the Azure classic portal in the Internet Explorer window, on the **MONITOR** tab of the PaaS cloud service you created in the first exercise with the **PRODUCTION** deployment selected, in the list of metrics, select the **Network In** metric for the AdatumAdsWebRole role.
2. On the right on the metric, click **1 rules configured**.
3. In the list of rules, click **Network In Alert**.
4. Inspect the data for the alert.
5. Open a new browser tab in Internet Explorer.
6. On the address bar, type **www.outlook.com**, and then press Enter.
7. If you are prompted to sign in, use the user name and password of the service administrator account of your Azure subscription.
8. In the list of emails, click **Microsoft Azure Alerts**.
9. Inspect the details of the alert.
10. Close Internet Explorer.

Task 4: 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 box, click **Yes**.

3. Type the following command, and then press Enter.
4. `Reset-Azure`
5. When prompted, sign in by using the Microsoft account associated with your Azure subscription.
6. If you have multiple Azure subscriptions, select the one you want the script to target.
7. When prompted for confirmation, type **y**.

Note: This script removes Azure services from your subscription. It is therefore recommended that you use an Azure trial pass that was provisioned specifically for this course and not your own Azure account. The script takes 5-10 minutes to reset your Microsoft Azure environment so that it is ready for the next lab. The script removes all storage, virtual machines, virtual networks (VNETs), cloud services, and resource groups. **Important**: The script might not be able to get exclusive access to a storage account to delete it. (If this occurs, you will see an error.) If you find objects remaining after the reset script is complete, you can rerun the **Reset-Azure** script or use the Azure portal and the Azure classic portal to manually delete all the objects in your Azure subscriptionâ€”with the exception of the default directory.

Result: At the end of this exercise, you will have configured monitoring for a PaaS cloud service with new metrics and an alert.

Lab Answer Key: Module 9: Implementing Azure Active Directory

Lab: Implementing Azure AD

Exercise 1: Administering Active AD

Task 1: Create directories

1. Ensure that the MSL-TMG1 and 20533C-MIA-CL1 virtual machines are both running, and then sign in to 20533C-MIA-CL1 as **Student** with the password **Pa$$w0rd**.

2. Start Internet Explorer, browse to **http://manage.windowsazure.com**, and then sign in by using the Microsoft account that is associated with your Azure subscription.

3. In the navigation panel on the left, click **ACTIVE DIRECTORY**.

4. Click **New**, click **DIRECTORY**, and then click **CUSTOM CREATE**.

5. In the **Add directory** dialog box, enter the following settings, and then click **Complete** (check mark):

    - **DIRECTORY**: Create new directory

    - **NAME**: Adatum

    - **DOMAIN NAME**: *Use your initials + the directory name + random numbers* (for example, abcadatum123456). If you get the **The domain is not unique** message, change the numbers until you get a green check mark.

    - **COUNTRY OR REGION**: United States

Task 2: Activate Azure AD Premium trial

1. In the navigation panel on the left, click **ACTIVE DIRECTORY**.

2. Click the **Adatum** directory.

3. Click **Licenses**.

4. Click **Try Azure Active Directory premium now**.

5. In the Activate Azure AD Premium trial pop-up window, click the check mark to confirm the selection.

6. Click the **Click here to refresh** link, and then verify that Azure AD Premium is activated.

Task 3: Manage users by using the Azure portal

1. On the **active directory** page, click **Adatum**.

2. On the **Adatum** page, click **USERS**.

3. Click the **ADD USER** button at the bottom of the page.

4. In the **Tell us about this user** dialog box, enter the following settings, and then click **Next**:

- **TYPE OF USER**: New user in your organization
- **USER NAME**: rdesforges

5. In the **user profile** dialog box, enter the following settings, and then click **Next**:
   - **FIRST NAME**: Remi
   - **LAST NAME**: Desforges
   - **DISPLAY NAME**: Remi Desforges
   - **ROLE**: User
   - **Enable Multi-Factor Authentication**: Do not select
6. Click **Create**.
7. On the **Get temporary password** page, note the new password.
8. Click **Complete** (check mark).
9. Click **ADD USER**.
10. In the **Tell us about this user** dialog box, enter the following settings, and then click **Next**:
    - **TYPE OF USER**: New user in your organization
    - **USER NAME**: kgruber
11. In the **user profile** dialog box, enter the following settings, and then click **Next**:
    - **FIRST NAME**: Karen
    - **LAST NAME**: Gruber
    - **DISPLAY NAME**: Karen Gruber
    - **ROLE**: Global Admin
    - **ALTERNATE EMAIL ADDRESS**: Type the email address of the Microsoft account that is the Service Administrator or a Co-Administrator of your Azure subscription.
    - **Enable Multi-Factor Authentication**: Do not select
12. Click **Create**.
13. On the **Get temporary password** page, note the new password.
14. Click **Complete** (check mark).
15. At the top-right corner of the page, click your Azure subscription name, and then click **Sign Out**.
16. On the **You have been signed out** page, click **SIGN IN**.
17. On the **Microsoft Azure** page, click **Use another account**, and then sign in to Azure by using the following credentials (where *XXXadatumXXX* is your unique Adatum number):

- **Username**: kgruber@XXXadatumXXX.onmicrosoft.com
- **Password**: Enter the temporary password you noted above.

18. On the **Update your password** page, in the **Current password** box, type the temporary password. In the **New password** and **Confirm password** boxes, type **Pa$$w0rd123**, and click **Update password and sign in**.

    Note: Although kgruber is a Global Administrator, the attempt to sign in to the portal fails and the following message appears: We were unable to find any Azure subscriptions where you are a service administrator or co-administrator. This is because this account is not the Service Administrator or a Co-Administrator of the Azure subscription. This is by design.

19. Close Internet Explorer.

Task 4: Manage groups by using the Azure portal

1. Start Internet Explorer, browse to **https://manage.windowsazure.com**, and sign in by using the Microsoft account that is associated with your Azure subscription.
2. In the navigation panel on the left, click **ACTIVE DIRECTORY**.
3. Click **Adatum**.
4. Click **Configure**.
5. Scroll down and in the group management, enable **Delegated Group Management Enabled** by selecting **Yes.**
6. Click **Save**.
7. Click **GROUPS**.
8. Click **ADD A GROUP**.
9. In the **Add Group** dialog box, enter the following settings, and then click **Complete**:
    - **NAME**: Sales
    - **DESCRIPTION**: Sales team
10. Click **Sales**.
11. Click **ADD MEMBERS**.
12. In the **Add members** dialog box, click **Remi Desforges**, and click **Complete**.
13. Click the **Back** button.
14. Click **ADD GROUP**.
15. In the **Add Group** dialog box, enter the following settings, and then click **Complete**:
    - **NAME**: Marketing
    - **DESCRIPTION**: Marketing employees

16. Click **Marketing**.

17. CLICK **ADD Members**.

18. In the **Add members** dialog box, click **Remi Desforges**, and click **Complete** (check mark).

19. Click the **Back** button.

20. Click **ADD GROUP**.

21. In the **Add Group** dialog box, enter the following settings, and then click **Complete**:

    - **NAME**: Sales and Marketing
    - **DESCRIPTION**: Sales and Marketing employees

22. Click **Sales and Marketing**.

23. Click **ADD MEMBERS**.

24. In the **Add members** dialog box, click the **SHOW** drop-down box, select **Groups**, and click the **Confirm** button to the right of the **SHOW** drop-down box.

25. Click **Marketing**.

26. Click **Sales**.

27. Click **Complete** (check mark).

28. Click the **Back** button.

Task 5: Manage users and groups by using Azure PowerShell

1. On the taskbar, right-click **Windows PowerShell**, and then click **Run ISE as Administrator**.

2. If a **User Account Control** dialog box appears, click **Yes**.

3. In the PowerShell ISE, click **File**, and then click **Open**.

4. In the **Open** dialog box, browse to **D:\Labfiles\Lab09\Starter\**.

5. Click **ExampleCommands.ps1**, and then click **Open**.

6. If the script pane is not visible, on the **View** menu, click **Show Script Pane**.

7. In the PowerShell ISE, in the command prompt pane, enter the following command and press Enter:

8.
```
Connect-MsolService
```

9. In the **Enter Credentials** dialog box, sign in as **kgruber@XXXadatumXXX.onmicrosoft.com** (where *XXXadatumXXX* is your unique Adatum domain name) with a password of **Pa$$w0rd123**, and then click **OK**.

10. In the PowerShell ISE, in the script pane, locate the following code:

11.
```
New-MsolUser -UserPrincipalName mledford@<#Copy your Azure Directory name
here#>.onmicrosoft.com  -DisplayName "Mario Ledford" -FirstName "Mario" -LastName
"Ledford" -Password 'Pa$$w0rd123' -ForceChangePassword $false -UsageLocation "US"
```

12. Replace **<#Copy your Azure Directory name here#>** with your Azure AD directory name.

13. In the PowerShell ISE, in the script pane, select the code that you just edited.

14. On the toolbar, click the **Run Selection** button and wait for the script to complete.

15. In the PowerShell ISE, in the command prompt pane, enter the following command, and then press Enter:

16. `Get-MsolUser`

17. In the PowerShell ISE, in the script pane, locate the following code and select it:

18. `New-MsolGroup -DisplayName "Azure team" -Description "Adatum Azure team users"`

19. On the toolbar, click the **Run Selection** button and wait for the script to complete.

20. In the PowerShell ISE, in the command prompt pane, enter the following command, and press Enter:

21. `Get-MsolGroup`

22. In the PowerShell ISE, in the script pane, locate the following code and select it:

23. `$group = Get-MsolGroup | Where-Object {$_.DisplayName -eq "Azure team"}`

24. On the toolbar, click the **Run Selection** button, and wait for the script to complete.

25. In the PowerShell ISE, in the script pane, locate the following code and select it:

26. `$user = Get-MsolUser | Where-Object {$_.DisplayName -eq "Mario Ledford"}`

27. On the toolbar, click the **Run Selection** button, and wait for the script to complete.

28. In the PowerShell ISE, in the script pane, locate the following code and select it:

29. `Add-MsolGroupMember -GroupObjectId $group.ObjectId -GroupMemberType "User" -GroupMemberObjectId $user.ObjectId`

30. On the toolbar, click the **Run Selection** button, and wait for the script to complete.

31. In the PowerShell ISE, in the script pane, locate the following code and select it:

32. `Get-MsolGroupMember -GroupObjectId $group.ObjectId`

33. On the toolbar, click the **Run Selection** button, and wait for the script to complete.

34. Switch to Internet Explorer.

35. Click **USERS**, and verify that **Mario Ledford** appears in the list of users.

36. Click **GROUPS**, and verify that **Azure team** appears in the list of groups.

Result: After completing this exercise, you should have created some pilot users and groups in Azure AD by using the Azure portal and Microsoft Azure Active Directory Module for Windows PowerShell. You will also enable the Azure AD Premium functionality.

Exercise 2: Configuring SSO

Task 1: Add directory applications and configure SSO

1. On the **Adatum** directory page, click **APPLICATIONS**.

2. Click **ADD**.

3. In the **What do you want to do?** dialog box, click **Add an application from the gallery**.

4. In the **Add an application for my organization to use** dialog box, in the search box, type **Microsoft**, and then press Enter.
5. Click **Microsoft Account (Windows Live)**, in the **Display** Name text box, type **Microsoft Account**, and then click the check mark.
6. Verify that **Configure single sign-on** is enabled by default.
7. Click **Assign accounts**.
8. From the **Show** drop-down menu, select **AllUsers**, and then click the check mark. In the user list, click **Mario Ledford**.
9. At the bottom of the screen, click **ASSIGN**.
10. In the **Assign Users** dialog box, select **I want to enter Microsoft Account credentials on behalf of the user**.
11. In the **Email Address** box, type the email address of the Microsoft account associated with your Azure subscription. In the **Password** box, type the corresponding password, and then click the check mark.
12. Above Microsoft account, click the **Back** arrow.
13. At the bottom of the screen, click **ADD**.
14. In the **What do you want to do?** dialog box, click **Add an application from the gallery**.
15. In the **Add an application for my organization to use** dialog box, in the search box, type **Skype**, and then press Enter.
16. Click **Skype**, in the **Display Name** text box, type **Skype**, and then click the check mark.
17. Verify that **Configure single sign-on** is enabled by default.
18. Click **Assign accounts**.
19. From the **Show** drop-down menu, select **All Users**, and then click the check mark.
20. In the user list, click **Mario Ledford**.
21. At the bottom of the screen, click **ASSIGN**.
22. In the **Assign Users** dialog box, clear **I want to enter Skype credentials on behalf of the user**, and then click the check mark.
23. On the top right side of the page, click your Azure account name, and then click **Sign out**.

Task 2: Test SSO

1. Close and restart Internet Explorer. In the address box, type **https://account.activedirectory.windowsazure.com/applications**, and then press Enter.
2. On the **Microsoft Azure** page, click **Use another account**.
3. On the **Microsoft Azure** page, enter the following credentials (where *XXXadatumXXX* is your unique Adatum domain name), and then click **Continue**:

- Username: **mledford@XXXadatumXXX.onmicrosoft.com**
- Password: **Pa$$w0rd123**

4. On the applications page, click the ellipsis (…) next to **Microsoft Account**. Note the options to update the credentials and report a problem about the Microsoft account.
5. On the applications page, click **Microsoft Account**.
6. In the **Microsoft Account** dialog box, click **Install Now**.
7. On the Internet Explorer bar, click **Run**.
8. In the **Access Panel Extension** dialog box, on the **Welcome to the Access Panel Extension Setup Wizard** page, click **Next**.
9. On the **Install Access Panel Extension** page, click **Install**.
10. In the **User Account Control** dialog box, click **Yes**.
11. In the **Access Panel Extension** dialog box, on the **Completed the Access Panel Extension Setup Wizard** page, click **Finish**.
12. On the Internet Explorer bar, click **Enable**.
13. Close Internet Explorer.
14. On the taskbar, click **Internet Explorer**.
15. In Internet Explorer, in the address box, type **https://account.activedirectory.windowsazure.com/applications**, and then press Enter.
16. On the **Microsoft Azure** page, enter the following credentials (where *XXXadatumXXX* is your unique Adatum domain name), and click **Continue**.
    - Username: **mledford@XXXadatumXXX.onmicrosoft.com**
    - Password: **Pa$$w0rd123**
17. On the **applications** page, click **Microsoft Account**, and then in the Microsoft account, enter your subscription credentials.

    Note: If you are prompted to sign in again, use the credentials for your subscription account.
18. Verify that you signed in to your Microsoft account based on the credentials that have been entered on behalf of the user.
19. Switch to the **Access Panel Applications** tab.
20. On the **applications** page, click **Skype**; note that you are now prompted for credentials, because you did not enter any credentials on behalf of the user when configuring SSO.
21. Close the **Skype** dialog box.
22. Close Internet Explorer.

Result: After completing this exercise, you should have installed and configured a test application and validated the SSO experience.

Exercise 3: Configuring Multi-Factor Authentication

Task 1: Configure Multi-Factor Authentication

1. On the taskbar, click **Internet Explorer**.
2. In Internet Explorer, in the address box, type **https://manage.windowsazure.com**, and then press Enter.
3. On the **Microsoft Azure** page, click your Azure subscription name; if your Azure subscription is not shown, click **Use another account**.
4. On the **Sign in** page, enter the credentials for the Microsoft account associated with your Azure subscription, and then click **Sign in**.
5. In the navigation pane, scroll down, and click **ACTIVE DIRECTORY**.
6. Click the right arrow next to the **Adatum** directory.
7. Click **CONFIGURE**.
8. Under **multi-factor authentication**, click **Manage service settings**.
9. If you get a **Sign in** page, enter the Microsoft account associated with your Azure subscription, and then click **Sign in**.
10. On the **multi-factor authentication** page, click **users**.
11. In the **users** list, select the check box for **Karen Gruber**, and in the **quick steps** section, click **Enable**.
12. On the **About enabling multi-factor auth** page, click **enable multi-factor auth**.
13. On the **Updates successful** page, click **close**.
14. In Internet Explorer, close the **Multi-factor Authentication** tab.
15. Close Internet Explorer.

Task 2: Test Multi-Factor Authentication

1. On the taskbar, click **Internet Explorer**.
2. In Internet Explorer, in the address box, type **https://account.activedirectory.windowsazure.com/applications**, and then press Enter.
3. On the **Microsoft Azure** page, click **Use another account**.
4. On the **Sign in** page, enter the following credentials (where *XXXadatumXXX* is your unique Adatum domain name), and then click **Sign in**:
   - Username: **kgruber@XXXadatumXXX.onmicrosoft.com**
   - Password: **Pa$$w0rd123**

Note: Note the following message: Your admin has required that you set up this account for additional security verification.

1. Click **Set it up now**.
2. On the **additional security verification** page, click in the first box, and note the contact method options.
3. Optional step: If you have access to a mobile phone in the classroom, and have a signal or data connection, you can complete the additional security verification steps on the **additional security verification** page.

Result: After completing this exercise, you should have configured Multi-Factor Authentication for administrators.

Exercise 4: Configuring SSO from a Windows 10-based computer that is joined to Azure AD

Task 1: Join a Windows 10-based computer to Azure AD

1. On the taskbar, click **Internet Explorer**.
2. In Internet Explorer, in the address box, type **https://manage.windowsazure.com**, and then press Enter. If required, sign in by using the Microsoft account that is associated with your Azure subscription.
3. On MIA-CL1, click the Start button, and then click **Settings**.
4. In **Settings**, click **Accounts**, and then click **Work Access**.
5. On the **Connect to work or school** page, click **Join or leave an organization**. This will redirect you to the About section of the **SYSTEM Settings**.
6. Click **Join Azure AD**.
7. On the **What happens next** page, click **Next**.
8. On the **Let's get you signed in** page, type the following credentials, and then click **Sign in**:
    - Username: **kgruber@XXXadatumXXX.onmicrosoft.com**
    - Password: **Pa$$w0rd123**
9. On the **Help us protect your account** page, click **Set it up now**.
10. On the **Verify your identity** page, from drop-down menu, select **Phone call**.
11. In the **Select your country or region** drop-down list, select the country or region where your phone is registered. In the **Phone number** text box, type your phone number. Click **Next**.
12. Press the # key on your phone to complete verification.
13. On the **Keep your existing apps working** page, click **Next**.
14. On the **Make sure this is your organization** page, click **Join**.

15. On the **All finished** page, click **Finish**.
16. Switch back to Internet Explorer.
17. In the navigation panel on the left, click **ACTIVE DIRECTORY**.
18. Select the **Adatum** directory.
19. Click **Users**, and then select the **Karen Gruber** account.
20. Click the **Devices** tab.
21. On the **You are about to view private user data** page, select **it is acceptable for admins in my organization to view this data**, and then click **OK** (confirm selection).
22. From the **View** drop-down menu, select **Devices**.
23. Verify that **MIA-CL1** is listed.
24. Sign out of **MIA-CL1**.

Task 2: Authenticate to Azure from a Windows 10 Azure-joined computer

1. Sign in to MIA-CL1 by using the following credentials:
   - Username: **kgruber@XXXadatumXXX.onmicrosoft.com**
   - Password: **Pa$$w0rd123**
2. On the **Set up a PIN** page, click **Setup PIN**.
3. On the **Set up a PIN** page, type and retype a four-digit PIN, and then click **OK**. Note that you cannot use a common number pattern (such as four identical digits).
4. Start Internet Explorer, and then go to **https://portal.office.com**.
5. Verify that you are automatically signed in as Karen Gruber by using SSO.
6. Close all open applications and sign out from 20533C-Mia-CL1.

Task 3: Reset the environment

1. Sign in to 20533C-Mia-CL1 as **Student** with the password **Pa$$w0rd**.
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. In the PowerShell ISE, in the command prompt pane, enter the following command, and then press Enter:
5. `Reset-Azure`
6. When prompted (twice), sign in by using the Microsoft account associated with your Azure subscription.
7. If you have multiple Azure subscriptions, select the one you want the script to target.
8. When prompted for confirmation, type **y.**

Note: This script removes 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 resets your Azure environment so that it is ready for the next lab. The script removes all storage accounts, virtual machines, virtual networks, cloud services, and resource groups containing these resources.

Result: After completing this exercise, you should have joined the Mia-CL1 computer to Azure AD and tested the SSO access to the resources in the cloud.

Lab Answer Key: Module 10: Managing an Active Directory infrastructure in a hybrid environment

Lab: Implementing and managing Azure AD synchronization

Exercise 1: Configuring directory synchronization

Task 1: Sign in to the Azure VM hosting an Active Directory domain controller.

1. Sign in to MIA-CL1 as **Student** with the password **Pa$$w0rd**.
2. Open Internet Explorer and browse to the Azure portal at https://portal.azure.com.
3. When prompted, sign in by using the Microsoft account that is the Service Administrator or a co-admin of your Azure subscription.
4. In the hub menu of the Azure portal, click **Virtual machines (classic)**.
5. In the Virtual machines (classic) blade, click **AdatumDC1**.
6. In the AdatumDC1 blade, click **Connect**.
7. When prompted whether to open or save the .rdp flie, click **Save**.
8. In the **Remote Desktop Connection** dialog box, click **Connect**.
9. If a Remote Desktop Connection warning message displays, select **Don't ask me again for connections to this computer**, and then click **Connect**.
10. In the **Windows Security** dialog box, enter a user name of **ADATUM\Student** with the password **Pa$$w0rd123**.
11. If another Remote Desktop Message displays, select **Don't ask me again for connections to this computer**, and then click **Yes**.

Task 2: Create a new Azure AD tenant and a Global Admin account

1. After the sign-in is complete, click the Windows logo in the lower left corner, and then on the **Start** screen, click **Internet Explorer**.
2. If a **Set up Internet Explorer 11** dialog box opens, click **Use recommended security, privacy, and compatibility settings**, and then click **OK**.
3. In Internet Explorer, navigate to the Azure classic portal at **https://manage.windowsazure.com**.
4. When prompted, sign in to the Azure classic portal by using an account that is the Service Administrator or a co-admin of your Azure subscription.
5. In the Azure classic portal, click **ACTIVE DIRECTORY** in the navigation bar.
6. Click **New**, click **DIRECTORY**, and then click **CUSTOM CREATE**.

7. In the **Add directory** dialog box, provide the following settings, and then click **Complete** (check mark):
    - DIRECTORY: *Create new directory*
    - NAME: **AdatumSync**
    - DOMAIN NAME: *Use your initials + the directory name + random numbers* (for example, abcadatum123456). If you get the message **The domain is not unique**, change the numbers until you get a green check mark.
    - COUNTRY OR REGION: **United States**
8. On the **adatumsync** page, click the **USERS** tab.
9. Click **ADD USER** on the command bar.
10. In the **Tell us about this user** dialog box, enter the following settings, and then click **Next**:
    - TYPE OF USER: *New user in your organization*
    - USER NAME: **SyncAdmin**
11. In the **USER PROFILE** dialog box, provide the following settings, and then click **Next**:
    - FIRST NAME: **Sync**
    - LAST NAME: **Admin**
    - DISPLAY NAME: **Sync Admin**
    - ROLE: Global **Admin**
    - ALTERNATE EMAIL ADDRESS: *Type the email address of your Microsoft account*
    - Enable Multi-Factor Authentication: *Do not select*
12. Click **Create**.
13. On the **Get temporary password** page, note the full user name and the temporary password, and then copy them to Notepad.
14. Click **Complete** (check mark).
15. Click the cogwheel icon in the upper right corner of the Internet Explorer window, click **Safety** in the drop-down menu, and then click **inPrivate Browsing.**
16. In the InPrivate Browsing session, navigate to the Azure classic portal at **https://manage.windowsazure.com**.
17. When prompted, type the full name of the newly created SyncAdmin account, and then click **Continue**.
18. When prompted for the password, type the temporary password which you copied to Notepad, and then click **Sign in**.

19. On the **Update your password** page, in the **Current password** text box, type the temporary password, in the **New password** and **Confirm password** text boxes, type **Pa$$w0rd**, and then click **Update password and sign in**.

20. On the **No subscriptions found** page, click **SIGN OUT**.

21. Close the InPrivate Internet Explorer session.

Task 3: Install Azure AD Connect with custom settings

1. Open an Internet Explorer session and browse to https://www.microsoft.com/en-us/download/details.aspx?id=47594

2. On the **Microsoft Azure Active Directory Connect** page, click the cogwheel in the upper right corner of the Internet Explorer window and select **Internet options** from the drop down menu.

3. In the **Internet Options** dialog box, click the **Security** tab.

4. Click **Trusted sites**, and then click **Sites**.

5. In the **Add this website to the zone** text box, replace the current entry with **https://*.microsoft.com**, and then click **Add**.

6. Click **Close**, and then click **OK**.

7. Back on the **Microsoft Azure Active Directory Connect** page, click **Download**. Note - if this fails, close your IE browser, reopen, and browse again to https://www.microsoft.com/en-us/download/details.aspx?id=47594.

8. In the pop-up window, click **Save**.

9. After the download is complete, click **Open Folder**.

10. In the File Explorer window, double-click **AzureADConnect.msi** to start the installation.

11. On the **Welcome** page, select **I agree to the license terms and privacy notice**, and then click **Continue**.

12. On the **Express Settings** page, click **Customize**.

13. On the **Required Component** page, review the options, and then click **Install**.

14. On the **User sign-in** page, verify that **Password Synchronization** is selected, and then click **Next**.

15. On the **Connect to Azure AD** page, provide the credentials of the newly created SyncAdmin Azure AD Global Admin, and then click **Next**:

    - User name: **SyncAdmin@ *yourdomainname*.onmicrosoft.com**
    - Password: **Pa$$w0rd**

16. On the **Connect your directories** page, verify that the **adatum.com** forest is selected, under user name, type **ADATUM\Student** with the password **Pa$$w0rd123**, and then click **Add Directory**.

17. Verify that under Configured Directories, adatum.com is listed, and then click **Next**.
18. On the **Azure AD sign-in configuration** page, note the message **Users will not be able to sign-in Azure AD using their on-premises credentials**, select **Continue without any verified domains**, and then click **Next**.
19. On the **Domain and OUfiltering** page, select the **Sync selected domains and OUs** check box, expand the **adatum.com** entry, clear all checkboxes with exception of the one next to the **Accounts** organization unit, and then click **Next**.
20. On the **Uniquely identifying your users** page, verify that **Users are represented only once across all directories** is selected, and then click **Next**.
21. On the **Filtering** page, verify that **Synchronize all users and devices** is selected, and then click **Next**.
22. On the **Optional feature** page, verify that **Password synchronization** is selected, and then click **Next**.
23. On the **Ready to configure** page, verify that **Start the synchronization process when configuration completes** is selected, and then click **Install**.

    Note: Installation might take 5-10 minutes.

24. On the **Configuration complete** page, click **Exit** to close Azure AD Connect.> **Note:** You might need to wait a few minutes for the initial synchronization to complete.
25. Switch back to the Azure classic portal in the Internet Explorer window.
26. In the Azure classic portal, navigate to the **adatumsync** Active Directory page, click **USERS**, and then confirm that the list of users includes all the names from the Accounts organizational unit (OU).

Result: After completing this exercise, you should have installed and configured Azure AD Connect, and you should have it ready for test synchronization.

Exercise 2: Synchronizing directories

Task 1: Modify attributes of an Active Directory user and Initiate manual synchronization

1. On AdatumDC1, switch to the Server Manager window, and from the **Tools** menu, open the **Active Directory Administrative Center**.
2. Click **adatum (local)**, and then double-click **Accounts**.
3. After the Accounts OU content is displays, double-click the **Beverly Beach (bbeach)** account.
4. In the **Beverly Beach (bbeach)** window, make changes to the following fields, and then click **OK**:

   - **Job Title**: **VP**
   - **Department**: **Marketing**

5. On AdatumDC1, on the taskbar, right-click the **Windows PowerShell** shortcut, right-click **Windows PowerShell** on the menu and then click **Run as administrator**.

6. At the command prompt in the Windows PowerShell command-line interface, type the following command, and then press Enter: Get-ADSyncScheduler

   Note: Get-ADSyncScheduler displays the current configuration settings for synchronization with Azure AD.

7. At the Windows PowerShell command prompt, type the following command, and then press Enter: Start-ADSyncSyncCycle -PolicyType Delta

8. Wait until synchronization completes before proceeding to the next step.

9. Switch back to the Azure classic portal in the Internet Explorer window.

10. Click the USERS tab on the **adatumsync** page.

11. Click the **Beverly Beach** entry.

12. On the **Beverly beach** page, click **WORK INFO**.

13. Verify that the **JOB TITLE** and the **DEPARTMENT** entries match the ones you configured for the Active Directory account. If you do not see any changes, wait for a few minutes, and then refresh the page.

14. Close the AdatumDC1 remote desktop session, and then click **OK** when prompted.

Task 2: Reset the environment

1. On MIA-CL1, 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. Type the following command, and then press Enter:

5. `Reset-Azure`

6. When prompted (twice), sign in by using the Microsoft account associated with your Azure subscription.

7. If you have multiple Azure subscriptions, select the one you want the script to target.

8. When prompted for confirmation, type **y**.

Note: This script might remove Azure services in your subscription. We therefore 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 Azure environment and make it ready for the next module. The script removes all storage, virtual machines, virtual networks, cloud services, and resource groups. **Important:** The script might not be able to get exclusive access to a storage account to delete it; if this occurs, you will see an error. If you find objects remaining after the reset script is complete, you can rerun the **Reset-Azure** script, or you can use the Azure classic portal to delete all the objects in your Azure subscription manually, with the exception of the default directory.

Result: After completing this exercise, you should have changed attributes on a user account, and then forced synchronization.

Lab Answer Key: Module 11: Implementing Azure-based management and automation

Lab: Implementing Automation

Exercise 1: Configuring Automation accounts

Task 1: Create an Automation account

1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa$$w0rd**, and that the **Setup-Azure** script has completed.

2. Start Internet Explorer and browse to **https://portal.azure.com**. When prompted, sign in by using the Microsoft account that is the Service Administrator or Co-Administrator of your Azure subscription.

3. On the menu hub, click **New**, and then click **Management**.

4. In the **Management** blade, click **Automation**.

5. In the Add Automation Account blade, specify the following:

   - Name: **LabAutomationAccount**

   - Resource group: create a new resource group named **AutomationLabRG**

   - Subscription: your current subscription

   - Region: an Azure region that you chose when running the provisioning script

   - Account Options: leave at the default setting. This will create a tutorial runbook in the new account

   - Pin to dashboard: Check the box to enable

6. Click **Create**.

7. Wait for the Automation account to be provisioned. This should take less than a minute.

Task 2: Create an Azure AD user

1. In the Azure portal, click **Browse** in the hub menu on the left side of the page, and then click **Active Directory**. This should automatically open another Internet Explorer tab with the view of your **default directory**.

2. On the **default directory** page, click **USERS**.

3. Click **ADD USER** in the command bar at the bottom of the page.

4. On the **Tell us about this user** page of the ADD USER Wizard, specify the following:

   - TYPE OF USER: **New user in your organization**

   - USER NAME: **LabAutomationUser**

- @: leave the default value

5. On the **user profile** page of the ADD USER Wizard, in the **DISPLAY NAME** box, type **LabAutomation User**.
6. In the **ROLE** drop-down list, ensure that **User** is selected (do not enable Multi-Factor Authentication). Note that you are creating an organizational account, and you will make this account a co-administrator of your Azure subscription.
7. Click **Next**.
8. On the **Get temporary password** page of the ADD USER Wizard, note the full user name (including the part after the @ sign), and then copy it to Notepad.
9. Click **create** and note the temporary password shown in the **NEW PASSWORD** text box. Click the Copy icon to the right of the text box. If prompted, click **Allow access** and click the Copy icon again. Paste the copied password to Notepad.
10. Click **Complete**.
11. Click the large blue arrow pointing to the left.
12. In the navigation bar on the left side, scroll down to the bottom, and then click **SETTINGS**.
13. Click **ADMINISTRATORS**.
14. At the bottom of the page, click **ADD**.
15. In the **EMAIL ADDRESS** box, type the name of the new user that you created that you copied to Notepad in the format **LabAutomationUser@ <*domain*>**.
16. Under **SUBSCRIPTION**, select your current Azure subscription, and then click **OK**.
17. At the top right of the page, click your current account name, and then click **Sign out**.
18. On the **You have been signed out** page, click **SIGN IN**.
19. On the **Microsoft Azure** sign-in page, click **Use another account**.
20. On the **Sign in** page, enter the newly created user's credentials, and then click **Continue**.
21. When prompted for the password, type the user's password that you copied to Notepad, and then click **Sign in**.
22. On the **Update your password** page, in the **Current password** text box, type the temporary password.
23. In the ** New password** and **Confirm password** text boxes, type **Pa$$w0rd**, and then click **Update password and sign in**.
24. If the **Sign in** page appears, enter your new password, and then click **Sign in**.
25. Close the **WINDOWS AZURE TOUR** dialog box.
26. At the top right of the page, click the currently signed-in user account name, and then click **Sign out**.

Task 3: Create Automation assets

1. Switch back to the Azure portal. In the Hub menu, click Browse, and then click **Automation Accounts**.
2. In the Automation Accounts blade, click the Automation account you created in Exercise 1 Task 1.
3. In the LabAutomationAccount blade click the **ASSETS** tile.
4. On the **Assets** blade, notice that you have several Windows PowerShell modules included in your account by default.
5. Click **Credentials**.
6. In the **Credentials** blade, click **Add a credential**.
7. In the **New Credential** blade, specify the following:
   - Name: **PSCredential**
   - Description: **Lab Automation User (Co-Administrator)**
   - User name: the name of the newly created AutomationUser account that you copied to Notepad
   - Password: **Pa$$w0rd**
   - Confirm password: **Pa$$w0rd**
8. Click **Create**.
9. Close the **Credentials** blade.
10. Click the **Variables** tile.
11. In the **Variables** blade, click **Add a variable**.
12. In the **New Variable** blade, specify the following:
    - Name: **SubscriptionName**
    - Description: **Subscription Name**
    - Type: **String**
    - Value: *name of your subscription*
    - Encrypted: **No**
13. Click **Create**.
14. Repeat the steps 12 and 13 to create four string variables. For each variable, specify the following name and value (leave the description blank):
    - Name: **AdminName** Value: **Student**
    - Name: **AdminPassword** Value: **Pa$$w0rd**

- Name: **Location** Value: *Location: the name of the Azure region that you used when running the provisioning script at the beginning of this module*
- Name: **Network** Value: **ADATUM-HQ-VNET**
- Name: **Subnet** Value: **Subnet-1**

15. Back on the **Assets** blade, click the **Schedules** tile.
16. In the **Schedules** tile, click **Add a schedule**.
17. In the **New Schedule** blade, specify the following:
    - Name: **EndOfDay**
    - Description: **End of Day**
    - Starts: tomorrow's date at **18:00:00**
    - Recurrence: **Daily**
    - Recur every **1 Day**
    - Set expiration: Leave the default **No**
    - Expires: **Never**
18. Click **Create**.
19. Close the Schedules blade.
20. Close the Assets blade.

Result: After completing this exercise, you should have configured a new Microsoft Azure Automation account, and created a new Microsoft Azure Active Directory (Azure AD) organizational account to use as an Automation Credential asset.

Exercise 2: Creating runbooks

Task 1: Import a runbook

1. In the Azure portal, in the blade displaying your Automation account, click the **Runbooks** tile.
2. In the **Runbooks** blade, click **Add a runbook**.
3. In the **Add Runbook** blade, click **Import an existing runbook**.
4. In the **Import** blade, specify the following:
    - Runbook file: **D:\Labfiles\Lab11\Solution**, select **New-StorageAndVMs.ps1**
    - Runbook type: **PowerShell Workflow**
    - Name: **New-StorageAndVMs**
    - Description: leave blank

5. Click **Create**.
6. In the **Runbooks** blade, click **New-StorageAndVMs**.
7. In the New-StorageAndVMs blade, click **Edit**.
8. In the Edit PowerShell Workflow Runbook blade, review the content of the PowerShell workflow.

Task 2: Publish and execute a runbook

1. In the Edit PowerShell Workflow Runbook blade, click **Publish**.
2. When prompted to confirm, click **Yes**. You will be automatically redirected to the **New-StorageAndVMs** blade.
3. Click **Start**.
4. When prompted to confirm, click **Yes**. You will be automatically redirected to a blade displaying the current job, with a name consisting of the combination of the runbook name and timestamp of its invocation.
5. Click **Output** tile.
6. Monitor the runbook execution. Wait until the job completes.

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 box, click **Yes**.
3. Type the following command, and then press Enter:
4. `Reset-Azure`
5. When prompted (twice), sign in by using the Microsoft account associated with your Azure subscription.
6. If you have multiple Azure subscriptions, select the one you want to target with the script.
7. When prompted for confirmation, type **y**.

Note: This script will remove Azure services in your subscription. We therefore 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 to be ready for the next lab. The script removes all storage, virtual machines (VMs), virtual networks, cloud services, and resource groups. **Important**: The script might not be able to get exclusive access to a storage account to delete it (if this occurs, you will see an error). If you find remaining objects after the reset script is complete, you can rerun the **Reset-Azure** script, or use the Azure portal and Azure classic portal to delete all the objects in your Azure subscription manuallyâ€”with the exception of the default directory.

Result: After completing this exercise, you should have imported, published, and executed a PowerShell workflow-based runbook that deploys two virtual machines in parallel.