# Lab Answer Key: Module 5: Implementing Azure App Service

# Lab: Implementing Azure web apps

## Exercise 1: Creating web apps

#### Task 1: Create a web app

1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa55w.rd**. Ensure that the Add-20533EEnvironment script successfully completed.
2. Open Microsoft Edge, browse to the Azure portal at [**http://portal.azure.com**](http://portal.azure.com), and then sign in using the Microsoft account that is the Service Administrator of your Azure subscription.
3. In the top-left corner of the portal, click **+ Create resource**, and then click **Web**.
4. On the New blade, click **Web App**.
5. On 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. On the Web App blade, in the **Resource Group**, verify that the **Create new** option is selected and then, in the **Resource group** text box, type **20533E0501-LabRG**.
7. Click **Windows**.
8. Click **App Service plan/Location**.
9. On the App Service plan blade, click **Create New**.
10. On the New App Service Plan blade, in the **App Service plan** text box, type **20533E0501LabPlan**.
11. In the **Location** drop down list, select an Azure region close to the lab location, in which you can create app service plans.
12. Click **Pricing tier**, next, click **Production**, click **S1 Standard**, and then click **Apply**.
13. On the New App Service Plan blade, click **OK**.
14. On the Web App blade, leave the **Application Insights** setting at its default value.
15. On the Web App blade, click **Create**. Wait until the web app is provisioned.

#### Task 2: Add a deployment slot

1. In the hub menu of the Azure portal, click **All services**, and then click **App Services**.
2. On the App Services blade, click the web app that you created in the first task.
3. On the web app blade, in the **DEPLOYMENT** section, click **Deployment slots**.
4. On the Deployment slots blade, click **Add Slot**.
5. On the Add a slot blade, in the **Name** text box, type **Staging**.
6. In the **Configuration Source** list, accept the default setting and then click **OK**.
7. On MIA-CL1, click **Start**, in the Start menu, expand the **Windows PowerShell** folder, right-click **Windows PowerShell**, click **More**, and then click **Run as administrator**. When prompted by User Account Control for confirmation, click **Yes**.
8. Sign in to the Azure subscription by typing the following command in the Windows PowerShell window, and then pressing Enter:

Add-AzureRmAccount

1. In the sign-in windows that appears, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
2. If you have multiple subscriptions associated with your Microsoft account, to identify the subscription in which you are going to create a virtual network, type the following command, and then press Enter:

Get-AzureRmSubscription

1. Note the value of the ***Id*** property for each subscription in the output of the previous command. To specify the subscription in which you are going to create a virtual network, type the following commands, and then press Enter (replace ***SubscriptionId*** with the actual **SubscriptionId** property of that subscription):

Set-AzureRmContext -SubscriptionId 'SubscriptionId'

1. Type the following and then press Enter:

Get-AzureRmWebApp -ResourceGroupName '20533E0501-LabRG'

1. Verify that the output references the web app that you created in the previous task.
2. Type the following and then press Enter. Replace ***Name of your web app*** with the name of the web app you chose in the previous task:

Get-AzureRmWebAppSlot -ResourceGroupName '20533E0501-LabRG' -Name 'Name of your web app'

1. Verify that the web app staging slot you created in this task is listed in the output
2. Keep the Windows PowerShell window open.

#### Task 3: Configure deployment credentials

1. In the Azure portal, in the web app that you created in the first task, on the Deployment Center blade, click **FTP** and then click **Dashboard**.
2. In the FTP dashboard, switch to the **User Credentials** tab, and in the **FTP/Deployment user name** text box, type a unique name.
3. In the **Password** text box, type **Pa55w.rd**.
4. In the **Confirm password** box, type **Pa55w.rd**, and then click **Save**.

**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 deployment credentials.

## Exercise 2: Deploying a web app

#### Task 1: Obtain a publishing profile

1. In the Azure portal, on the blade of the web app that you created in the exercise 1, click **Overview** and then click **Get publish profile**. You might need to first click **... More** if the **Get publish profile** option does not appear in the toolbar at the top of the blade.
2. When prompted whether to open or save the **.PublishSettings** file, click **Save**. Microsoft Edge will save the publishing profile in the **Downloads** folder on your lab computer.
3. On MIA-CL1, click **Start**, in the Start menu, click **Visual Studio 2017**
4. On the **File** menu, click **Open**, and then click **Project/Solution**.
5. Browse to the folder **F:\LabFiles\Lab05\Starter\AdatumWebsite**, click **AdatumWebsite.sln**, and then click **Open**. If prompted, in the **Security Warning for AdatumWebsite** dialog box, clear the checkbox **Ask me for every project in this solution** and click **OK**.
6. In the **Debug** menu, click **Start Debugging**. This will open a new tab in the Microsoft Edge window.
7. On the new Microsoft Edge tab, under **A. Datum Corporation**, click **Learn More**.
8. In the Microsoft Edge window, click **Contact**.
9. Close the new Microsoft Edge tab.

#### Task 2: Deploy a web app

1. In Visual Studio, click **Debug** and, in the **Debug** menu, click **Stop Debugging**.
2. In the Solution Explorer, right-click the **AdatumWebsite** project, and then click **Publish**.
3. On the Publish page, click **Create new profile**.
4. In the **Pick a publish target** window, click **Import Profile**.
5. In the **Import Publish Settings** dialog box, browse to the **Downloads** folder.
6. Select the **.PublishSettings** file that you downloaded in Task 1 of this exercise, click **Open**.
7. Press **Publish** button and it will automatically build and publish the web app from Visual Studio to the Azure Web app you created in the first exercise and open a new tab in the Microsoft Edge window displaying it.
8. Verify that A. Datum's web app is open in a new Microsoft Edge tab and verify the web app's URL.
9. Close the new Microsoft Edge tab.
10. Leave Visual Studio open.

**Result**: After completing this exercise, you should have deployed a web app hosted in Azure.

## Exercise 3: Managing web apps

#### Task 1: Deploy a web app for staging

1. In Microsoft Edge, in the Azure portal, navigate to the blade of the web app you created in Exercise 1, Task 1.
2. On the web app blade, in the **DEPLOYMENT** section, click **Deployment slots**.
3. On the Deployment slots blade, click the staging slot ***yourwebapp*-staging** that was created in Exercise 1, Task 2.
4. On the Staging blade, click **Get publish profile**. You might need to first click **... More** if the **Get publish profile** option does not appear in the toolbar at the top of the blade.
5. When prompted, click **Save**.
6. Switch to **Visual Studio**.
7. On the **File** menu, click **Open**, and then click **Project/Solution**.
8. Browse to the folder **F:\LabFiles\Lab05\Starter\NewAdatumWebsite**.
9. Click **AdatumWebsite.sln**, and then click **Open**. If prompted, in the **Security Warning for AdatumWebsite** dialog box, clear the checkbox **Ask me for every project in this solution** and click **OK**.
10. In Solution Explorer, right-click the **AdatumWebsite** project, and then click **Publish**.
11. On the Publish page, click **Create new profile**.
12. In the **Pick a publish target** window, click **Import Profile**.
13. In the **Import Publish Settings** dialog box, browse to the **Downloads** folder.
14. Select the **.PublishSettings** file that you downloaded in step 5 of this task, click **Open**.
15. Press **Publish** button and it will automatically build and publish the web app from Visual Studio to the Azure Web app you created in the first exercise and open a new tab in the Microsoft Edge window displaying it.
16. Verify that A. Datum's web app is open in a new Microsoft Edge tab and verify the web app's URL.
17. Close the new Microsoft Edge tab.
18. Leave Visual Studio open.

#### Task 2: Swap deployment slots

1. In the Azure portal, navigate to the web app blade and, in the **Essentials** section, click the **URL** link for your web app. This will open another tab in a new Microsoft Edge tab. Note the color scheme.
2. Close the tab that displays the A. Datum web app.
3. In the Azure portal, on the web app blade, click **Swap**
4. On the Swap blade, in the **Swap type** drop-down list, verify that **Swap** is selected.
5. In the **Source** drop-down list, ensure that **Staging** is selected.
6. In the **Destination** drop-down list, ensure that **production** is selected, and then click **OK**.
7. Wait until swap operation completes.
8. On the web app blade, in the **Essentials** section, click the URL link for your web app. This will open another tab in a new Microsoft Edge tab. Note the color scheme. Notice that the color scheme has changed.
9. Close the tab that displays the A. Datum's web app.

#### Task 3: Roll back a deployment

1. In the Azure portal, on the web app blade in the command bar at the top, click **Swap**.
2. On the Swap blade, in the **Swap type** drop-down list, verify that **Swap** is selected.
3. In the **Source** drop-down list, select **production**.
4. In the **Destination** drop-down list, select **Staging**, and then click **OK** button.
5. Wait until Swap operation completes.
6. On the web app blade, in **Essentials** section, click the URL link for your web app. This will open another tab in a new Microsoft Edge tab. Notice that the color scheme reverted to the original one.
7. Close the A. Datum tab in Microsoft Edge.

**Result**: After completing this exercise, you should have an updated web app in the staging slot and have tested the slot swap functionality.

## Exercise 4: Implementing Traffic Manager

#### Task 1: Deploy a web app to another region

1. Switch to the Windows PowerShell window.
2. At the Windows PowerShell prompt, type the following command, and then press Enter:

* Get-AzureRmWebApp -ResourceGroupName '20533E0501-LabRG'

1. Note the name of your original web app and its location.
2. Choose an Azure region that is different from the location of the original web app, preferably on another continent. This will become the ***SecondLocation***. To identify names of Azure regions, at the Windows PowerShell prompt, type the following command, and then press Enter:

* ((Get-AzureRmResourceProvider -ProviderNamespace Microsoft.Web).ResourceTypes | Where-Object ResourceTypeName -eq sites).Locations
* **Note:** You can also use Get-AzureRmLocation | Select-Object -Property Location, DisplayName or (Get-AzureRmLocation).Location

1. At the Windows PowerShell prompt, type the following command to create a new resource group, and then press Enter (replace ***SecondLocation*** with the name of the Azure region you chose):

* $rg2 = New-AzureRmResourceGroup -Name '20533E0502-LabRG' -Location 'SecondLocation'

1. At the Windows PowerShell prompt, type the following command to create a new App Service Plan, and then press Enter:

* $appSvcPlan2 = New-AzureRmAppServicePlan -Location $rg2.Location -Tier Standard -Name '20533E0502LabPlan' -ResourceGroupName $rg2.ResourceGroupName

1. At the Windows PowerShell prompt, type the following command (where the ***webappname2*** is a unique name that you will assign to a new web app you will create in the ***SecondLocation*** in the next step), and then press Enter:

* Test-AzureRmDnsAvailability -DomainNameLabel 'webappname2' -Location $rg2.Location
* **Note:** This command is case sensitive and accepts lower case only but the WebApp name is not; therefore, you can verify whether DNS is taken but then you can still use upper case to create your Web App in step 9.

1. Verify that the command returns **True**. If not, keep re-running the same command but with different values of the **DomainNameLabel** parameter.
2. At the Windows PowerShell prompt, type the following command to create a new web app, and then press Enter (the ***webappname2*** matches the name you identified in the previous step):

* New-AzureRmWebApp -ResourceGroupName $rg2.ResourceGroupName -Name 'webappname2' -Location $rg2.Location -AppServicePlan $appSvcPlan2.Name
* **Note:** If this cmdlet returns an error change the 'webappname2' value.

1. Switch to the Azure portal in the Microsoft Edge window.
2. On the left side of the Azure portal, click **All services**, and then click **App Services**.
3. On the App Services blade, click the entry representing the second web app you provisioned in this task.
4. On the web app blade, click **Get publish profile**. You might need to first click **... More** if the **Get publish profile** option does not appear in the toolbar at the top of the blade.
5. When prompted, click **Save**.
6. Switch to Visual Studio.
7. In the Visual Studio, on the **File** menu, click **Open**, and then click **Project/Solution**.
8. Browse to the folder **F:\LabFiles\Lab05\Starter\AdatumWebsite**.
9. Click **AdatumWebsite.sln**, and then click **Open**.
10. In Solution Explorer, right-click the **AdatumWebsite** project, and then click **Publish**.
11. On the Publish page, click **Create new profile**.
12. In the Pick a publish target window, ensure that the **Create New** option is selected, click **Import profile**, and click **OK**.
13. In the **Import Publish Settings** dialog box, browse to the **Downloads** folder.
14. Select the **.PublishSettings** file that you downloaded in step 13 of this task, and then click **Open**.
15. Press **Publish** button and it will automatically build and publish the web app from Visual Studio to the Azure Web app you created in this task and open a new tab in the Microsoft Edge window displaying it.
16. Verify that A. Datum's web app is open in a new Microsoft Edge tab and verify the web app's URL.
17. Close the new Microsoft Edge tab.
18. Close Visual Studio.

#### Task 2: Create a Traffic Manager profile

1. In Microsoft Edge, in the Azure portal, click **+ Create resource**.
2. In the **Search the marketplace** text box, type **Traffic Manager profile** and, in the list of results, click **Traffic Manager profile**.
3. On the **Traffic Manager profile** blade, click **Create**.
4. On 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.
5. In the **Routing Method** drop down list, select **Performance**.
6. In the **Subscription** drop down list, select the Azure subscription where you provisioned the web apps in this lab.
7. In the **Resource Group** section, ensure that **Create new** is selected and, in the text box below, type **20533E0503-LabRG**.
8. In the **Resource group location** drop-down list box, select the Azure region in which you provisioned the first web app, and then click **Create**. Wait until the Traffic Manager profile is created.

#### Task 3: Add endpoints, and configure Traffic Manager

1. In the Azure portal, in the hub menu, click **All services**, and then click **Traffic Manager Profiles**.
2. On the Traffic Manager profiles blade, click the profile you created in the previous task.
3. On the Traffic Manager profile blade, click **Endpoints**.
4. Click **Add**.
5. On the Add endpoint blade, ensure that **Azure endpoint** is selected in the **Type** drop down list.
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, click **App Service**.
8. Click **Choose an app service**.
9. On the Resource blade, click the web app that you created in Exercise 1.
10. Click **OK** to add the endpoint.
11. Repeat steps 4 through 10 to add the second endpoint for the web app that you created in Exercise 4. Use the name of the second web app as the name of the endpoint.
12. On the traffic manager profile blade, click **Configuration**.
13. On the Traffic Manager configuration blade, in the **DNS time to live (TTL)** text box, replace the default value by typing **30** and press **Enter**.
14. On the command bar at the top, click **Save**.

#### Task 4: Test Traffic Manager

1. In Microsoft Edge, in the Azure portal, on the traffic Manager profile blade, click **Overview**, wait until the **MONITOR STATUS** column displays the **Online** status for both web apps, and then, copy the URL to Microsoft Edge browser under the **DNS name** section.
2. Microsoft Edge displays the Adatum web app.
3. On MIA-CL1, click **Start**, in the Start menu, expand the **Windows System** folder, right-click **Command Prompt**, click **More**, and then click **Run as administrator**. When prompted by User Account Control for confirmation, click **Yes**.
4. At the Command Prompt, type the following command, replacing ***dnsname*** with the fully qualified DNS name of the Traffic Manager profile, and then press Enter:

nslookup dnsname

1. Note the DNS records that the command returns. This should include the name of the first web app.
2. In the Microsoft Edge window, switch to the tab that displays the Azure portal.
3. On the Traffic Manager profile blade, click **Endpoints**
4. In the list of endpoints, click the name of the web app, which name was included in the DNS records you noted in step 5.
5. On the endpoint blade, click **Disabled**, and then click **Save**.
6. Switch to the command prompt, type the following command, replacing ***dnsname*** with the fully qualified DNS name of the Traffic Manager profile, and then press Enter:

nslookup dnsname

1. Note that the record that the command returns is different from the one returned in step 4.

**Note:** You might have to wait in order for the endpoint state change to take effect. Wait about 1 minute and re-run the **nslookup** command.

#### Task 5: Reset the Azure environment

1. On MIA-CL1, close all open windows without saving any files.
2. On the taskbar, right-click the **Windows PowerShell** icon, and then click **Run as Administrator**. In the User Account Control dialog box, click **Yes**.
3. Type the following command, and then press **Enter**:

Remove-20533EEnvironment

1. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
2. If you have multiple Azure subscriptions, select the one you want the script to target.
3. If prompted, specify the current lab number.
4. When prompted for confirmation, type **y**.
5. Start Microsoft Edge, browse to the Azure portal at [**http://portal.azure.com**](http://portal.azure.com), and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
6. In the Azure portal, on Dashboard, click **Edit**.
7. Right-click unoccupied area of the dashboard and, in the right-click menu, click **Reset to default state**. When prompted to confirm, click **Yes**.
8. Click **Done customizing**.
9. Close all open windows.

**Result**: After completing this exercise, you should have implemented two Azure web apps and a Traffic Manager profile configured to distribute requests between them.

©2016 Microsoft Corporation. All rights reserved.

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

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