# Module 5: Implementing Azure App Service

# Lab: Implementing web apps

### Scenario

The A. Datum Corporation's public-facing web app currently runs on an IIS web server at the company's chosen ISP. A. Datum wants to migrate this web app into Azure. You must test the Web Apps functionality by setting up a test A. Datum web app. An internal team provides you with a test web app to deploy. You must ensure that they can continue to stage changes to the test web app before deploying those changes to the public-facing site. A. Datum is a global company, so you also want to test Azure Traffic Manager, and show your organization's decision makers how it distributes traffic to instances close to users of the web app.

### Objectives

After completing this lab, you will be able to:

* Create a new web app.
* Deploy a web app.
* Manage web apps.
* Implement Traffic Manager to load-balance web apps.

### Lab Setup

Estimated Time: 60 minutes

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

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

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

Before you begin this lab, ensure that you perform the "Preparing the Azure environment" demonstration tasks at the beginning of this module's first lesson, and that the setup script is complete.

## Exercise 1: Creating web apps

### Scenario

You must set up a test web app in Azure. As the first step in the setup process, you want to create a new web app. Later in this lab, you will deploy this web app to the test web app.

The main tasks for this exercise are as follows:

1. Create a web app
2. Add a deployment slot
3. Configure deployment credentials

#### 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. In Internet Explorer, browse to **http://portal.azure.com**, and then sign into the portal by using a Microsoft account that is either the Service Admin or co-admin of your subscription.
3. To create a new web app, use the following information:

* App name: *any unique valid server name*
* Resource Group: **AdatumLabWebRG**
* Web Hosting Plan Name: **WebAppStandardPlan**
* Pricing tier: **S1 Standard**
* Location: *a location near you*

#### Task 2: Add a deployment slot

1. In the Azure portal, add a new deployment slot to the web app that you created in the first task, using the following information:

Name: **Staging**

Configuration Source: *choose the web app you created in Task 1*

1. Open Azure PowerShell window and sign in to the Azure subscription with a Microsoft account that is either the Service Admin or co-admin of your subscription.
2. If you have multiple subscriptions, select the target one by running the Azure PowerShell **Set-AzureRmContext** cmdlet.
3. Use the Azure PowerShell **Get-AzureRMWebApp** and **Get-AzureRMWebAppSlot** cmdlets to identify the web app and staging slot that you created.
4. Keep the Azure PowerShell window open.

#### Task 3: Configure deployment credentials

* Use the Settings blade to set the following deployment credentials for the web app that you created in the first task:
* FTP/Deployment User Name: **ftpadmin *XXXX*** (replace *XXXX* is a unique number)
* Password: **Pa$$w0rd**

**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

### Scenario

Now that you created a web app in Azure, and added a deployment slot for the web app, you can publish the internally developed web app that the A. Datum web-development team supplied. In this exercise, you will use a publishing profile in Visual Studio 2015 to connect to the new web app and deploy the web content.

The main tasks for this exercise are as follows:

1. Obtain a publishing profile
2. Deploy a web app

#### Task 1: Obtain a publishing profile

1. From the Azure portal, download the publish profile for the Web app you created in Exercise 1.
2. Open the following web-application project in Visual Studio 2015:

* **D:05.sln**

1. Start debugging the web application, examine the contents, and then close Internet Explorer.

**Note:** When you start the web application in Visual Studio, the web app runs in IIS Express on your local workstation.

#### Task 2: Deploy a web app

1. In Visual Studio, start the Publish Wizard for the **AdatumWebsite** project, and then import the **.PublishSettings** file that you downloaded in task 1 of this exercise.
2. Verify that the publish settings file includes correct connection information.
3. Ensure that the Release configuration is used for the published web app.
4. Preview the file changes, and then **Publish** the new website to Azure. > **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.
5. Verify that A. Datum's web app is open in Microsoft Edge, and then verify the web app's current address.
6. Close the Home Page tab.
7. 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

### Scenario

The web-deployment team created an updated style sheet for the A. Datum's test web app. You have to demonstrate to the decision makers how you can deploy these changes to a staging slot, and then test them, before you deploy to the production A. Datum web app. In this exercise, you will upload the new web app to the staging slot that you created in Exercise 1, and you then will move the new site into the production slot.

The main tasks for this exercise are as follows:

1. Deploy a web app for staging
2. Swap deployment slots
3. Roll back a deployment

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

1. In the Azure portal, download a publishing profile for the Staging slot for your web app.
2. Open the following project in Visual Studio:

* **D:05.sln**

1. Publish the new web app, and then import the staging publish settings file that you just downloaded.
2. Validate the connection, and then choose the **Release** configuration.
3. Publish the new web app to the Staging slot.
4. Close Internet Explorer and Visual Studio.

#### Task 2: Swap deployment slots

1. In Internet Explorer, access the web app that you created in Exercise 1.
2. Notice that the color scheme has not changed, because the Web app with the new color scheme is still in the staging slot. Close the A. Datum web app.
3. From the Settings blade of your web app, open the Deployments Slots blade, and swap the staging and production web-app slots.
4. When the swap completes, browse the web app, and notice that the color scheme is new.
5. Close the tab that displays the A. Datum's web app.

#### Task 3: Roll back a deployment

1. In the Azure portal, swap the staging and production slots again. > **Note:** By swapping the slots a second time, you simulate a deployment rollback.
2. When the swap is complete, browse the web app. Notice that the color scheme is reverted to the old scheme.
3. 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

### Scenario

Because A. Datum is a global brand, you must ensure that the A. Datum web apps respond rapidly to requests from multiple locations around the world. You must evaluate Traffic Manager to see if it can ensure web content is served from a location that is close to users. You have to set up Traffic Manager to serve content from two different Azure regions.

The main tasks for this exercise are as follows:

1. Deploy a web app to another region
2. Create a Traffic Manager profile
3. Add endpoints, and configure Traffic Manager
4. Test Traffic Manager
5. Reset the Azure environment

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

1. In Azure PowerShell, obtain a list of all the web apps in your Azure subscription by using the **Get-AzureRMWebApp** cmdlet. Note the name of your original web app and location.
2. Choose an Azure region that is different from the location of the original web app. This will become the *"SecondLocation".*
3. Use the **New-AzureRMResourceGroup** cmdlet to create a new resource group named **AdatumLabWebRG2**.
4. Use the **New-AzureRMAppServicePlan** cmdlet to create a new App Service plan named **StandardPlan** with Standard pricing tier in the resource group **AdatumLabWebRG2** and the *"SecondLocation"*.
5. Use the **New-AzureRMWebApp** cmdlet to create a new web app. Use the following information for the web app:

* Resource group: **AdatumLabWebRG2**
* Name: Use the name of your original web app with the number 2 appended *.*
* Service plan: **StandardPlan**
* Location: *"SecondLocation".*

1. In the Azure portal, download a publishing profile for the web app you just created ( \_WebappName\_2).
2. Open the following project in Visual Studio:

* **D:05.sln**

1. Start the Publish Web Wizard, and then import the publish settings file that you just downloaded. > **Note:** Be sure to add a new publish settings file on the **Profile** tab, so that you can publish its content to the new web app.
2. Validate the connection, and then choose the Release configuration.
3. Publish the web app, and then close Internet Explorer and Visual Studio.

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

* In the Azure portal, create a new Traffic Manager profile by using the following information:
* DNS Prefix: *Use unique domain name*
* Routing Method: **Performance**
* Resource Group: **AdatumLabTMRG**

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

1. Use the Settings blade of the Traffic Manager profile to add the web apps that you created in Exercise 1 and Exercise 4 as endpoints.
2. In the Settings blade, use **Configuration** link to configure the DNS TTL value to be 30 seconds.

#### Task 4: Test Traffic Manager

1. Use the DNS name listed in the newly created Traffic Manager profile in the Azure Portal to browse to the corresponding URL by using Internet Explorer.
2. Use the **nslookup** command to resolve the DNS name for your Traffic Manager profile. > **Note:** In the DNS aliases, Traffic Manager returns the web app you created in Exercise 1, which is the closest to your physical location.
3. In the Azure portal, disable the Traffic Manager endpoint that is the web app you created in Exercise 1.
4. Use the **nslookup** command to resolve the DNS NAME for your Traffic Manager profile. The results should differ from those in step 2.

**Note:** If the aliases are not changed, 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:

Reset-Azure

1. When prompted (twice), sign in by using the Microsoft account associated with your Azure subscription.
2. If you have multiple Azure subscriptions, select the one you want the script to target
3. 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 message. 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.

**Question** In Exercise 2, you deployed the A. Datum production web app to Azure. In Exercise 3, you deployed a new version of the site to a staging slot. How can you tell, within Internet Explorer, which is the production site and which is the staging site?

**Question** At the end of Exercise 4, you used an FQDN within the trafficmanager.net domain to access your web app. How can you use your own registered domain name to access this web app?

©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.