**The lecture 7**

**Creating and deploying Web Apps**

Now that you understand App Services and App Service plans, I’ll show you what a Web App is, discuss some of its features, and then talk about the various options you have for creating one. Then, I’ll show you how to use a couple of those options to create and deploy a Web App.

**What is a Web App?**

A Web App is a web application that is hosted in an App Service. The App Service is the managed service in Azure that enables you to deploy a web application and make it available to your customers on the Internet in a very short amount of time. As noted above, you don’t directly support the VMs on which your web app runs; they are managed for you. In fact, you don’t have access to those underlying VMs.

Supported languages include .NET, Java, PHP, Node.js, and Python. In addition to creating your own web app, there are several web applications available to use as a starting point, such as WordPress, Umbraco, Joomla!, and Drupal.

You can use continuous deployment with Team Foundation Server (TFS), GitHub, TeamCity, Jenkins, or BitBucket so that every time you commit a change, a new version of the web app is deployed. Scaling is done by scaling the App Service plan to which the web app belongs. You can scale the number of instances in and out on demand. You can configure autoscaling so Azure will scale it in or out for you depending on specific performance measures such as CPU percentage. You can also publish your website to multiple locations and use the Azure Traffic Manager to handle the routing of the traffic to the location nearest to your customer.

For diagnostics, you can gather performance statistics, application logging, web server logging, IIS logs, and IIS Failed Request logs. If you’re using Microsoft Visual Studio, you can even remotely debug your application while it is running in the cloud.

In short, there are many features available when using Web Apps to make it easy for you to deploy, manage, and troubleshoot a web application.

**Options for creating Web Apps**

There are multiple options for creating a Web App and deploying the content to an app service. Let’s look at a few of these, including the following.

* **Azure Marketplace** This contains all of the resources you can deploy in Azure. I’ll show you how you can use this to create Web Apps from preexisting templates such as WordPress.
* **Visual Studio Code** This is a free, open source, cross-platform code editor with debugging capabilities.
* **Visual Studio** This is Microsoft’s full-featured development IDE.

**Marketplace**

There are many pre-created websites and templates in the Azure Marketplace that you can use. To see all of the options available, log into the Azure portal and click New > Web + Mobile > See All. This shows the Marketplace blade filtered for Web and Mobile apps, as displayed in Figure 2-7.



Figure 2-7 Options in the Azure Marketplace for Web and Mobile apps.

If you scroll down on the page, you can see the categories. At the end of any row, clicking More will show additional options in that category. Here are just a few of the choices available:

**Web Apps** Web App, Web App + SQL, Web App + MySQL, WordPress, and Umbraco CMS

**Blogs + CMSs** Joomla!, Drupal, DNN, Orchard CMS, Umbraco CMS, and MonoX

**Starter Web Apps** ASP.NET, HTML5, Node.js, PHP, Apache Tomcat, and some examples like the Bakery web app and the Java Coffee Shop web app

**Visual Studio Code**

Visual Studio Code (VS Code) is a free, open source code editor with support for development operations such as debugging, task running, and version control. It runs on Windows, OS X, and Linux.

VS Code makes debugging easier, providing IntelliSense code completion and easy code refactoring. It integrates with Git and also package managers, repositories, and various build tools.

VS Code has built-in support for Node.js, JavaScript, and TypeScript. Using extensions, you can use VS Code to debug languages such as C#, C++, Python, Ruby, and PowerShell. There is also tooling for web technologies such as HTML, CSS, JSON, and Markdown.

Using the Azure portal, you can set your web app to get the source code from OneDrive, Dropbox, or a local code repository such as GitHub or Visual Studio Team Service. If you enable continuous deployment for your WebApp, updates will be published automatically when changes are made to your source repository. You can download Visual Studio Code for Windows, Linux, or Mac here: <https://code.visualstudio.com/#alt-downloads>*.*

**Visual Studio**

Visual Studio is a full development environment, giving you the ability to create many different kinds of applications including, but not limited to, ASP.NET MVC applications, .NET client applications, Windows Communication Foundation (WCF) services, Web APIs, and Cloud Services, using languages such as C#, C++, VB, F#, and XAML.

With Visual Studio, you can create a new web application and publish it to an app service in Azure. I’ll show you how to do this in an upcoming demo.

**Demo: Create a web app by using the Azure Marketplace**

Let’s take a look at how to create a web app from one of the templates available in the Azure Marketplace.

1. Log into the Azure portal. As seen in Figure 2-8, click New on the left side of the page, then click See All.



Figure 2-8 Go to the Marketplace Search blade.

1. This brings up the search screen for the Marketplace. All resources that can be deployed to Azure are listed in the Marketplace, including virtual machines, virtual networks, storage accounts, web apps, and so on. As shown in Figure 2-9, type in **WordPress** and press Enter to perform the search.



1. You see a list of matches, as displayed in Figure 2-10.



1. Select the row with WordPress from publisher WordPress. This shows you the blade for WordPress; click Create at the bottom to create a WordPress site. You now see a blade where you can start configuring your WordPress site, as displayed in Figure 2-11.



1. Now, start filling in the fields on this blade:

**App Name** This is used to create the URL to access your web app.

**Subscription** If the account you are using is associated with multiple subscriptions, select the subscription you want to use.

* **Resource Group** This is a way of grouping multiple resources that are related to one another, such as a web app and a database. Select the resource group you used for the App Service plan you created earlier.
* **App Service Plan** Select the App Service plan you created earlier in this chapter.
* Click Database to see the database settings, as shown in Figure 2-12. WordPress uses MySQL by default. Set your Database Name and Type (Shared or Dedicated). For Location, select the same region in which your app is going to run. Click Pricing Tier and select the least expensive, which at this time is Mercury. Click OK to save the database settings.



Figure 2-12 Specify database settings.

* Back on the WordPress Settings blade for your new website, click Legal Terms. If you agree with the Legal Terms, click OK at the bottom of that screen, which will set Legal Terms to Accepted.
* You can use Web App Settings (Optional) to set the WordPress-specific settings shown in Figure 2-13; this is optional.



* Back on the WordPress blade, select the check box to pin the web app to your Dashboard, then click Create. Azure will create the WordPress site for you.
1. After Azure has finished publishing the web app, click the tile on your Dashboard to open its properties, as displayed in Figure 2-14. To open the site, click the URL. You are prompted for the rest of the details needed to create your WordPress site, such as language, site title, username, password, and email address. After all the fields are filled in, click the Install WordPress button. After the WordPress installation is finished, you’re ready to go.





**Demo: Create an ASP.NET website in Visual Studio and deploy it as a web app**

To perform this tutorial, you must have Visual Studio 2013 or Visual Studio 2015 installed and the most recent version of the Azure Tools and SDK.

Create a new web application with Visual Studio by following these steps:

**1. Open Visual Studio. Select File > New > Project.**

**2.** Select ASP.NET Web Application; the dialog box for creating a project appears, as shown in Figure 2-16. On the right side of the dialog box, clear the Add Application Insights To Project check box. This will prevent the creation of a separate Application Insights instance for this web application.



**3. Specify the Name of the application and the Location for the solution, then click OK.**

**4.** When prompted to select the type of ASP.NET application to create, select MVC from the list of ASP.NET Templates, as shown in Figure 2-17. Clear the Host In The Cloud check box. You will set that up separately. Click OK to continue.



**5.** Visual Studio will create a basic ASP.NET MVC application that runs “as is.” You can modify it later to make it your own.

**6.** Now, publish this web application to an App Service in Azure and assign it to the App Service plan created earlier in this chapter. You will create the App Service when you publish the web app the first time. Right-click the website and select Publish (Figure 2-18).



1. The Publish Web dialog box will be displayed. Select the Microsoft Azure App Service (Figure 2-19).



1. You will be prompted for your subscription name. You may be prompted again to enter the credentials for your Azure subscription. If the correct account is not displayed, click it to show a drop-down list and add an account if necessary. When the correct account is selected, select the Subscription and be sure the View is set to Resource Group. Open the Resource Group, and you will see the resources that have been set up already. In Figure 2-20, you can see the web apps that I have already created. To publish this application to a new web app, click New.



1. The Create App Service dialog box (Figure 2-21) appears next. Remember that an App Service is simply the host for a Web App, Mobile App, Logic App, API App, or Function App. You’ll create a new App Service to host your MVC web application here.



* Set the Web App Name. This will be used for the URL for the web app, so select it wisely.
* Select the Subscription.
* Select the Resource Group. If you use the one you created at the beginning of this chapter, then when you’re done, you can delete that Resource Group and all of your resources will be removed.
* Last, select the App Service plan that you created earlier in this chapter. This application will be hosted on the same VMs as the other web app(s) you have placed in that plan.

Click Create to create the App Service.

If you look in the Azure portal now, you will see your App Service has been created.

1. Now let’s use Web Deploy to publish our web app to our app service. After creating the app service, the Publish Web dialog box will be displayed (Figure 2-22). You can use the default values.

