

Lab 2: Migrating an ASP.NET web application to Azure Web Apps

What you will learn

In this lab, you will publish your legacy ASP.NET application to an Azure Web App, out of Visual Studio 2017.

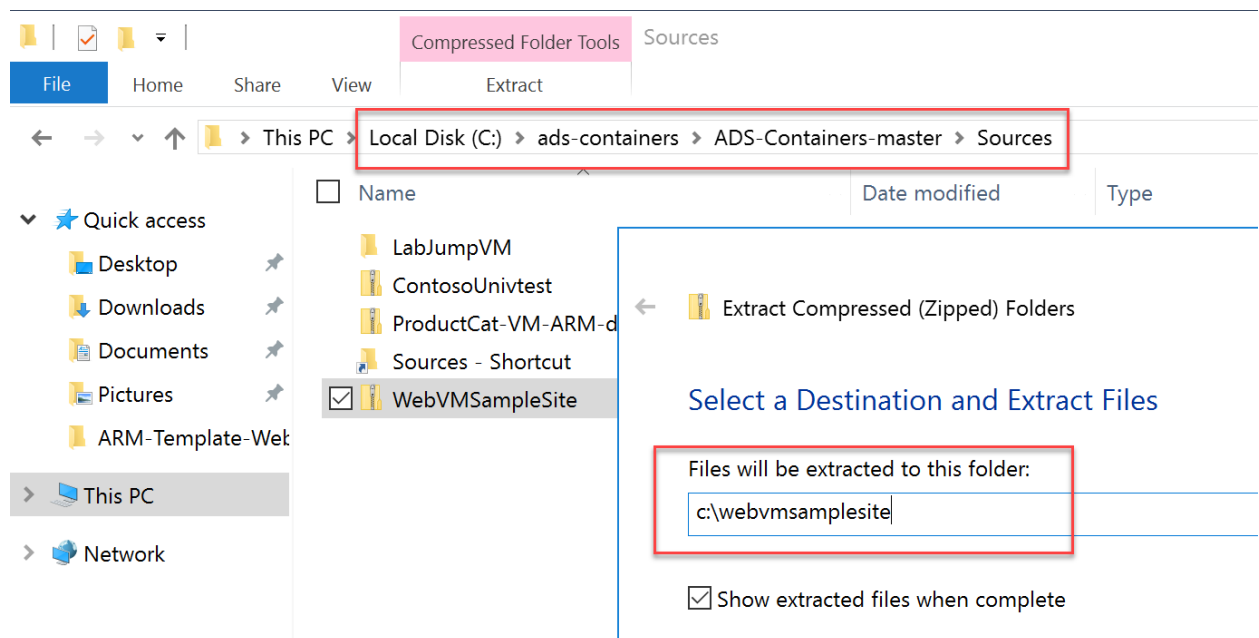
- Server admin login: labadmin
- Password: [L@BadminPa55w.rd](#)

Time Estimate

This lab shouldn't take longer than 15 minutes.

Task 1: Publish an ASP.NET project to Azure Web Apps from within Visual Studio 2017

1. **Log on** to the lab-JumpVM Virtual Machine (fyi, credentials labadmin / L@BadminPa55w.rd).
2. **From the lab-jumpVM**, browse to the folder that holds the GitHub downloaded source files. In here, find the **WebVMSampleSite.zip** file. This is the source code of the legacy ASP.NET web application. **Extract** this file to a folder on the local jumpVM, e.g. c:\webvmsamplesite



3. Once extracted, open the file **web.config** in the extracted folder.

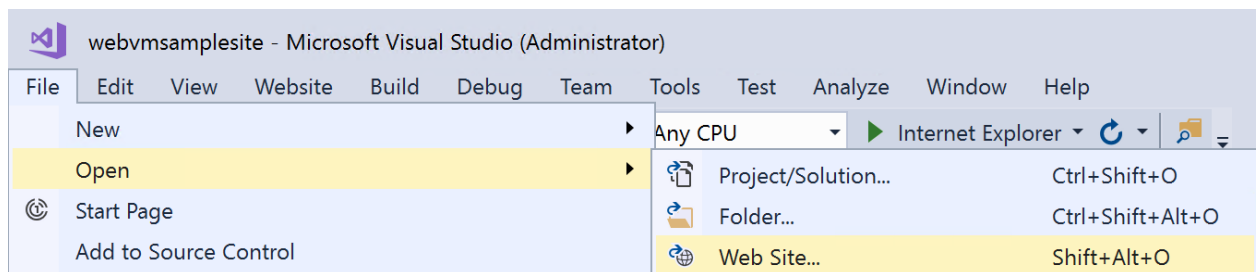
4. In here, edit the `<connectionStrings>` settings again, **Replacing the** following settings with the parameters from the Connection String information in the Azure Portal, in both lines starting with `<add name=`:

- `Data Source=10.0.1.4 =>` change the 10.0.1.4 with your SQL Server name e.g. `adssqlazure0923.database.windows.net` in our example

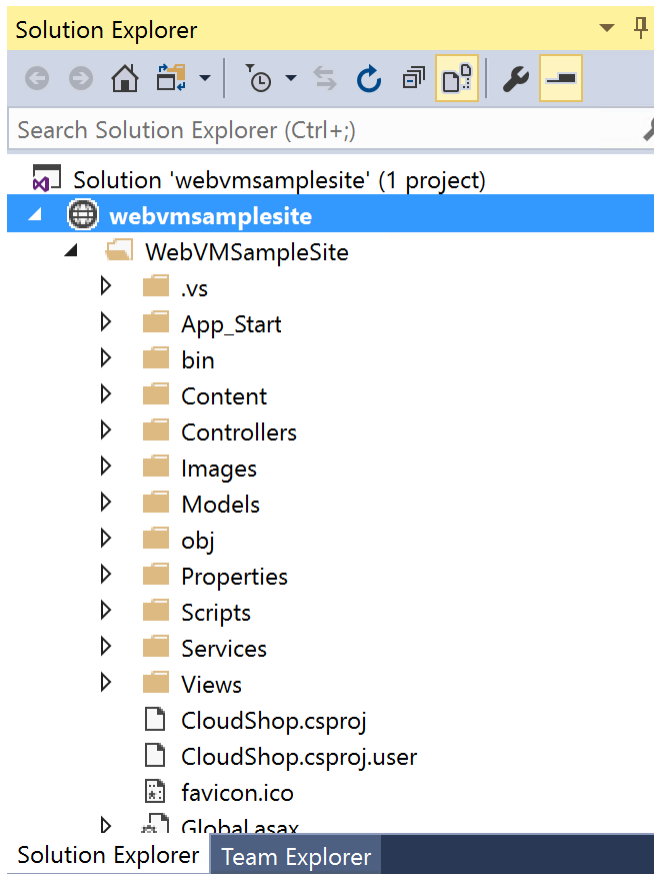
- `Uid=sa =>` change the sa account to labadmin

```
<connectionStrings>
<add name="DefaultConnection" connectionString="Data Source=adssqlazure0923.database.windows.net;
initial catalog=AdventureWorks;Uid=labadmin;Password=L@BadminPa55w.rd;MultipleActiveResultSets=True"
providerName="System.Data.SqlClient" />
<add name="AdventureWorksEntities" connectionString="metadata=res://*/Models.AdventureWorks.csdl
|res://*/Models.AdventureWorks.ssdl |res://*/Models.AdventureWorks.msl;provider=System.Data.SqlClient;
provider connection string="data source=adssqlazure0923.database.windows.net;initial
catalog=AdventureWorks;Uid=labadmin;Password=L@BadminPa55w.rd;multipleactiveresultsets=True;App=Entity
Framework";" providerName="System.Data.EntityClient" />
</connectionStrings>
```

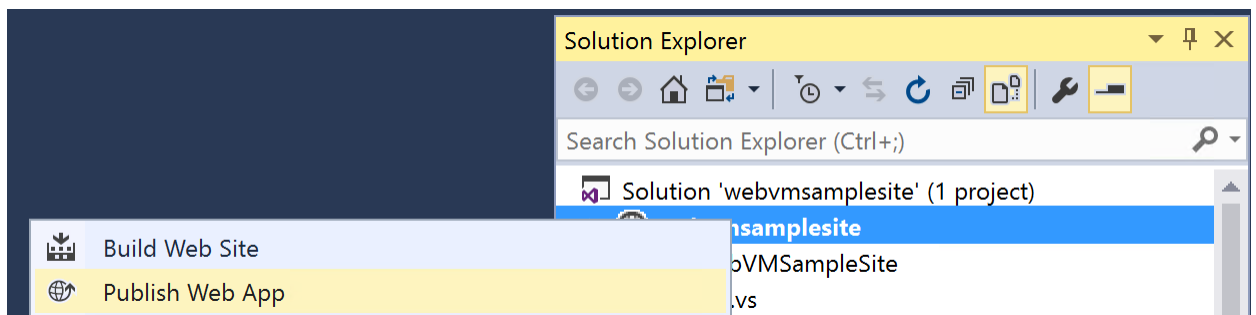
5. **Save** the changes to the web.config file.
6. Once this step is done, launch Visual Studio 2017.
7. From the **Visual Studio** menu, click **File / Open / Web Site...** and browse to the location where you extracted the webvmsamplesite



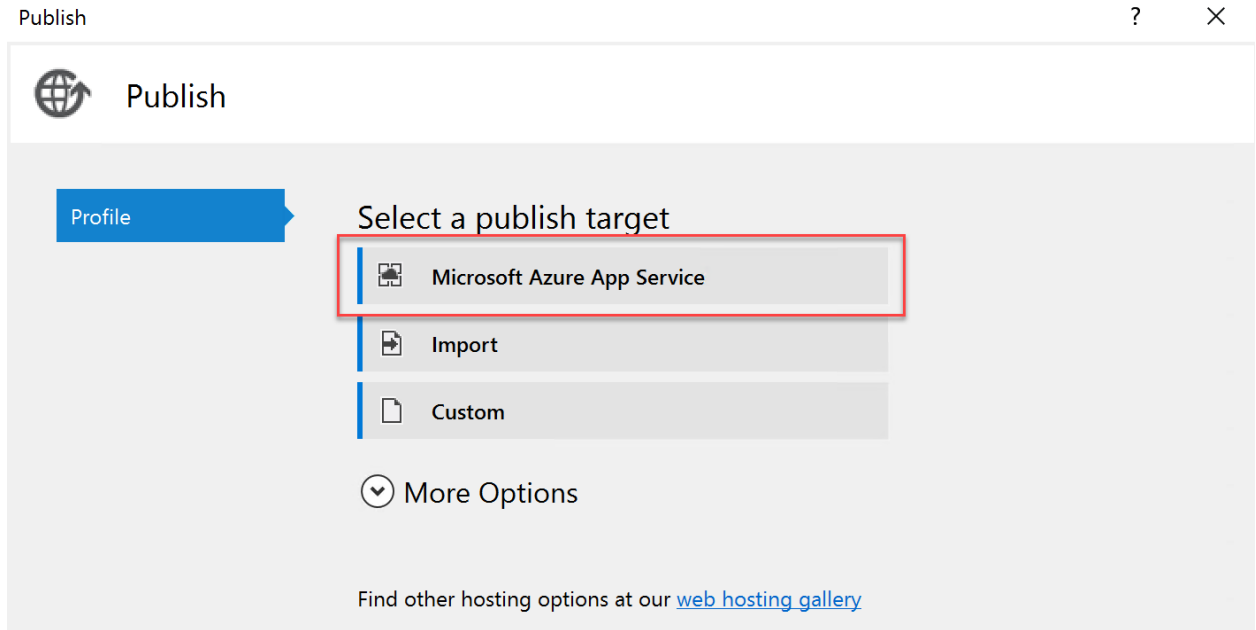
8. The Visual Studio **Solution Explorer** should show you the contents of the folder



9. Now, right-click on the web site again, and this time choose **Publish web app**



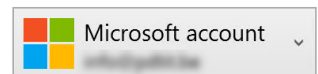
10. This launches the Publish wizard. In the **Select a publish target**, choose **Microsoft Azure App Service**.



11. In the **App Service** step, validate your Azure credentials and subscription are the correct ones, and click the **New** button.

App Service

Host your web and mobile applications, REST APIs, and more in Azure



Subscription

Microsoft Azure Sponsorship

View

Resource Group

Search

New...

12. In the **Create App Service** step, define
 - the WebApp Name: "SUFFIX-webapp1"
 - a New Resource Group "SUFFIX-webapp1",
 - a New App Service Plan "SUFFIX-webapp1ServicePlan"

×

Create App Service

Host your web and mobile applications, REST APIs, and more in Azure

Microsoft account

Hosting

Services

Web App Name

Change Type ▾

ADS-webapp1

Subscription

Mic... Microsoft Azure Subscription

Resource Group

ADS-WebApp1 (centralus) ▾

New...

App Service Plan

ADS-webapp1Plan* ▾

New...

Clicking the Create button will create the following Azure resources

[Explore additional Azure services](#)

App Service - ADS-webapp1

App Service Plan - ADS-webapp1Plan

If you have removed your spending limit or you are using Pay as You Go, there may be monetary impact if you provision additional resources.

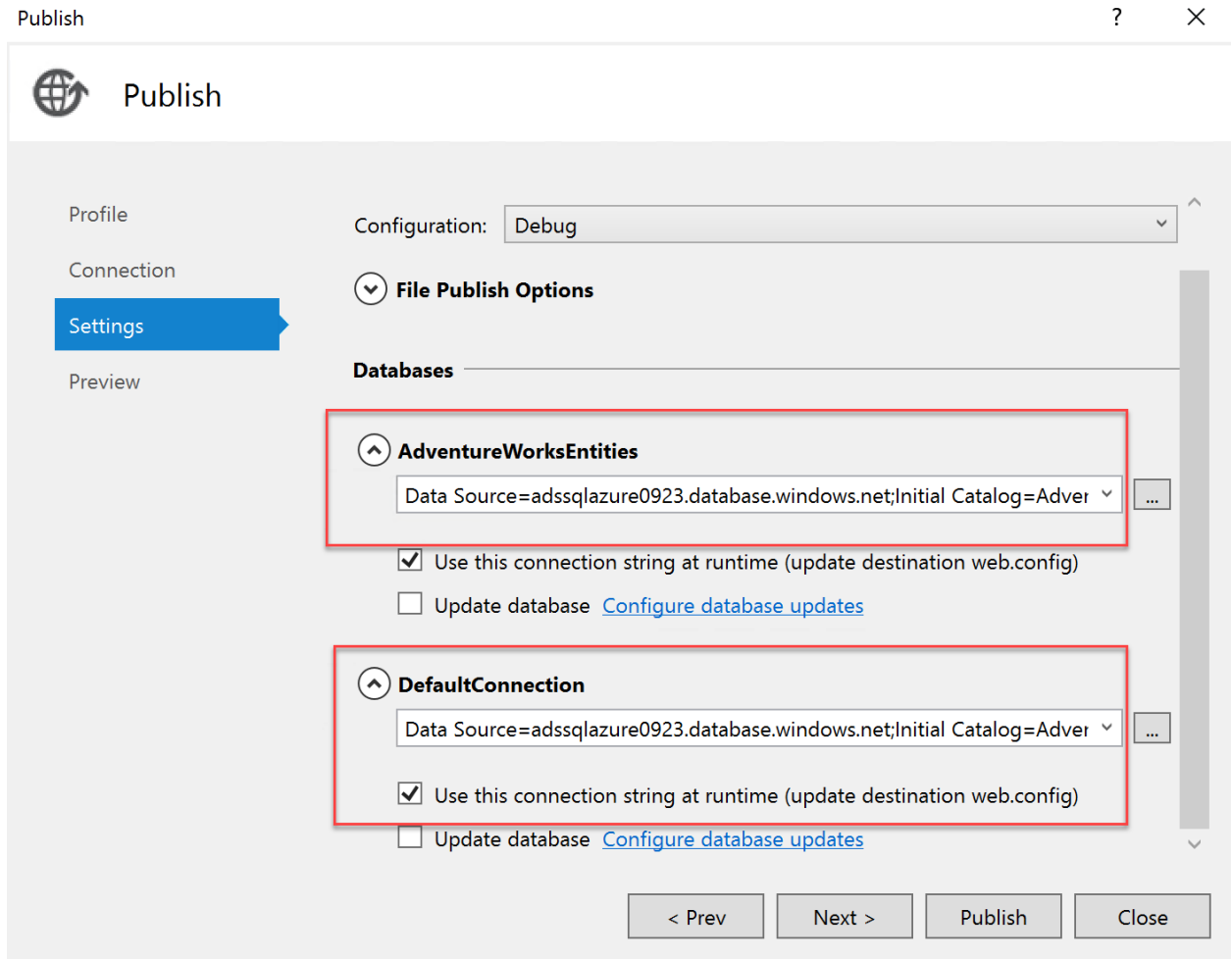
[Learn More](#)

Export...

Create

Cancel

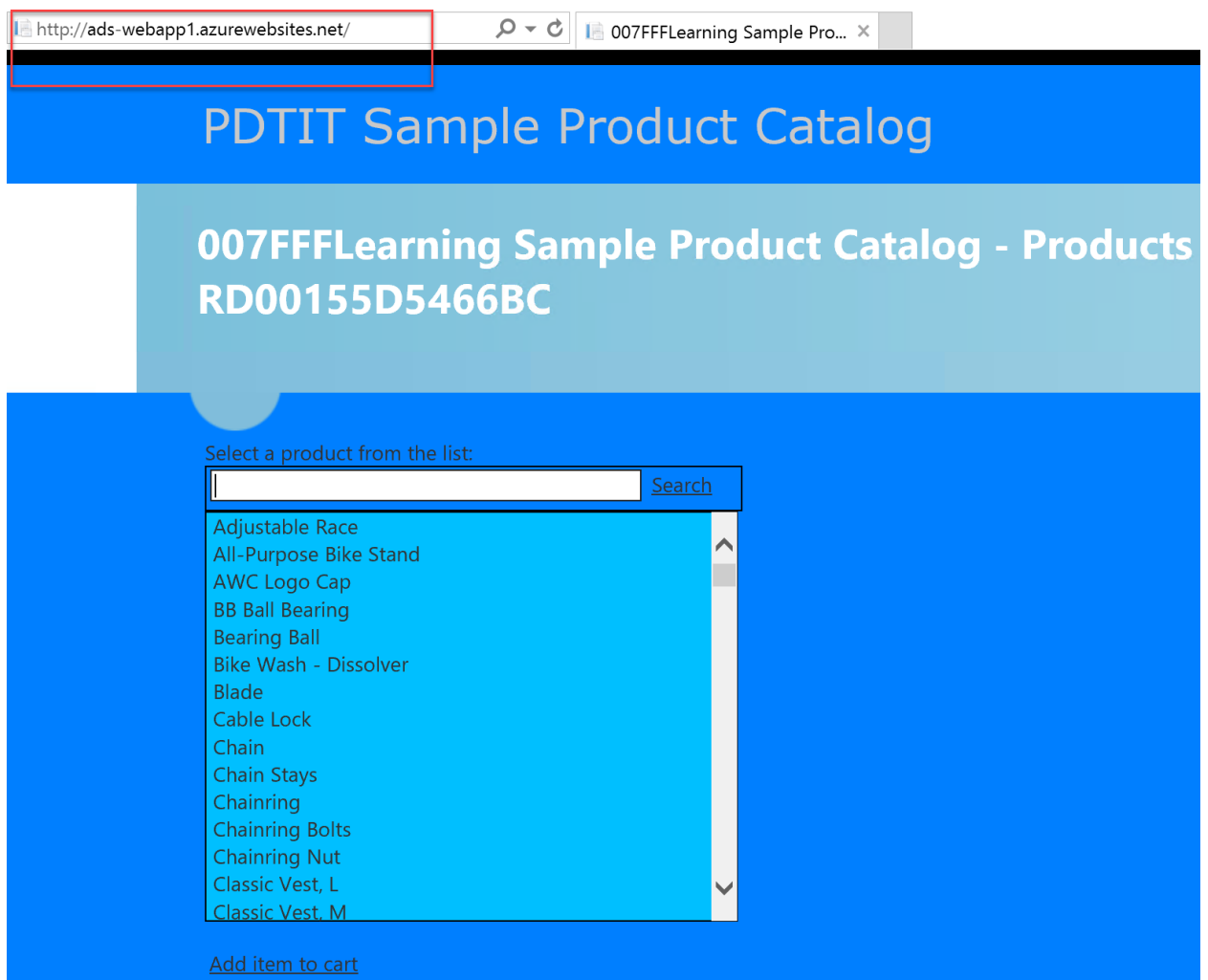
13. Click the **Create** button to kick off the actual Web App resource creation. This creates the different Azure Resource objects (Resource Group and App Service Plan) in the back-end. After which you get redirected back to the **Publish** wizard.



16. Accept the presented information. (Note this gives interesting options towards developers to test the web app with test/dev databases, right from within this web app publishing wizard).
17. Click **Publish** to finalize this publish wizard step, and wait for the Web App publish steps to be completed successfully.
18. From the Visual Studio Output window, you can follow the publishing sequence in a log file.

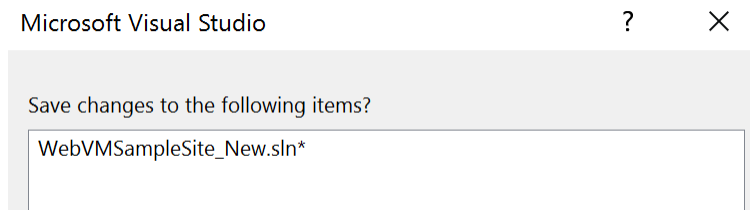
```
Output
Show output from: Build
----- Publish started: Project: WebVMSampleSite_New, Configuration: Debug Any CPU -----
Transformed Web.config using C:\WebVMSampleSite_New\Web.Debug.config into C:\Users\labadmin\AppData\Local\Temp\2\WebSitePublish\We
Auto ConnectionString Transformed obj\Release\Package\PackageTmp\Web.config into C:\Users\labadmin\AppData\Local\Temp\2\WebSitePub
Auto ConnectionString Transformed obj\Release\Package\PackageTmp\Views\Web.config into C:\Users\labadmin\AppData\Local\Temp\2\WebS
Auto ConnectionString Transformed obj\Release\TransformWebConfig\assist\Web.config into C:\Users\labadmin\AppData\Local\Temp\2\We
Auto ConnectionString Transformed obj\Release\TransformWebConfig\original\Web.config into C:\Users\labadmin\AppData\Local\Temp\2\W
Auto ConnectionString Transformed obj\Release\TransformWebConfig\transformed\Web.config into C:\Users\labadmin\AppData\Local\Temp\
Auto ConnectionString Transformed Views\Web.config into C:\Users\labadmin\AppData\Local\Temp\2\WebSitePublish\WebVMSampleSite_New-
Auto ConnectionString Transformed C:\Users\labadmin\AppData\Local\Temp\2\WebSitePublish\WebVMSampleSite_New--476791108\obj\Debug\T
Copying all files to temporary location below for package/publish:
C:\Users\labadmin\AppData\Local\Temp\2\WebSitePublish\WebVMSampleSite_New--476791108\obj\Debug\Package\PackageTmp.
Start Web Deploy Publish the Application/package to https://ads-webapp1.scm.azurewebsites.net/msdeploy.axd?site=ADS-webapp1 ...
```

19. Once the publish step is complete, Visual Studio will open your internet browser and connect to the URL of the new Azure Web App.

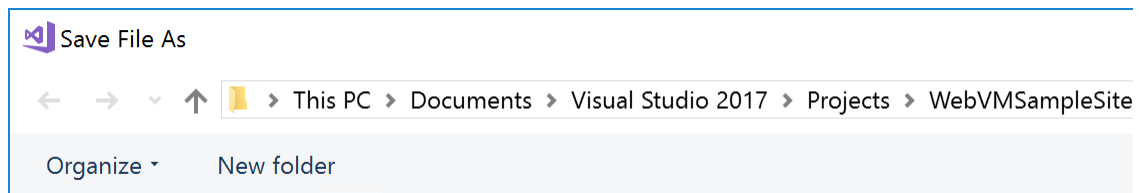


20. This confirms our Azure Web is published successfully, communicating with the SQL Azure database.

21. Close your internet browser.
22. Go back to **Visual Studio 2017**, and **close** the application. This will **prompt you to save** the Visual Studio application (.sln).



23. Click **Yes**, and accept the default location that is offered.



24. This closes Visual Studio 2017.

This completes this lab.

Summary

In this lab exercise, you learned how to update connection string settings in a web.config file, and how to publish your website code to an Azure Web App, by using the Visual Studio Web App Publishing wizard. Lastly, you saved your web site content as a new Visual Studio Project.