

Module 14 - Lab 1: Implementing an Azure App Service web app with a staging slot

? Scenario

This module includes the following topics:

- Deploying Web Apps
- Managing Web Apps

Web Apps overview

Azure App Service Web Apps (or just Web Apps) is a service for hosting web applications, REST APIs, and mobile back ends. You can develop in your favorite language, be it .NET, .NET Core, Java, Ruby, Node.js, PHP, or Python. Applications run and scale with ease on Windows-based environments.

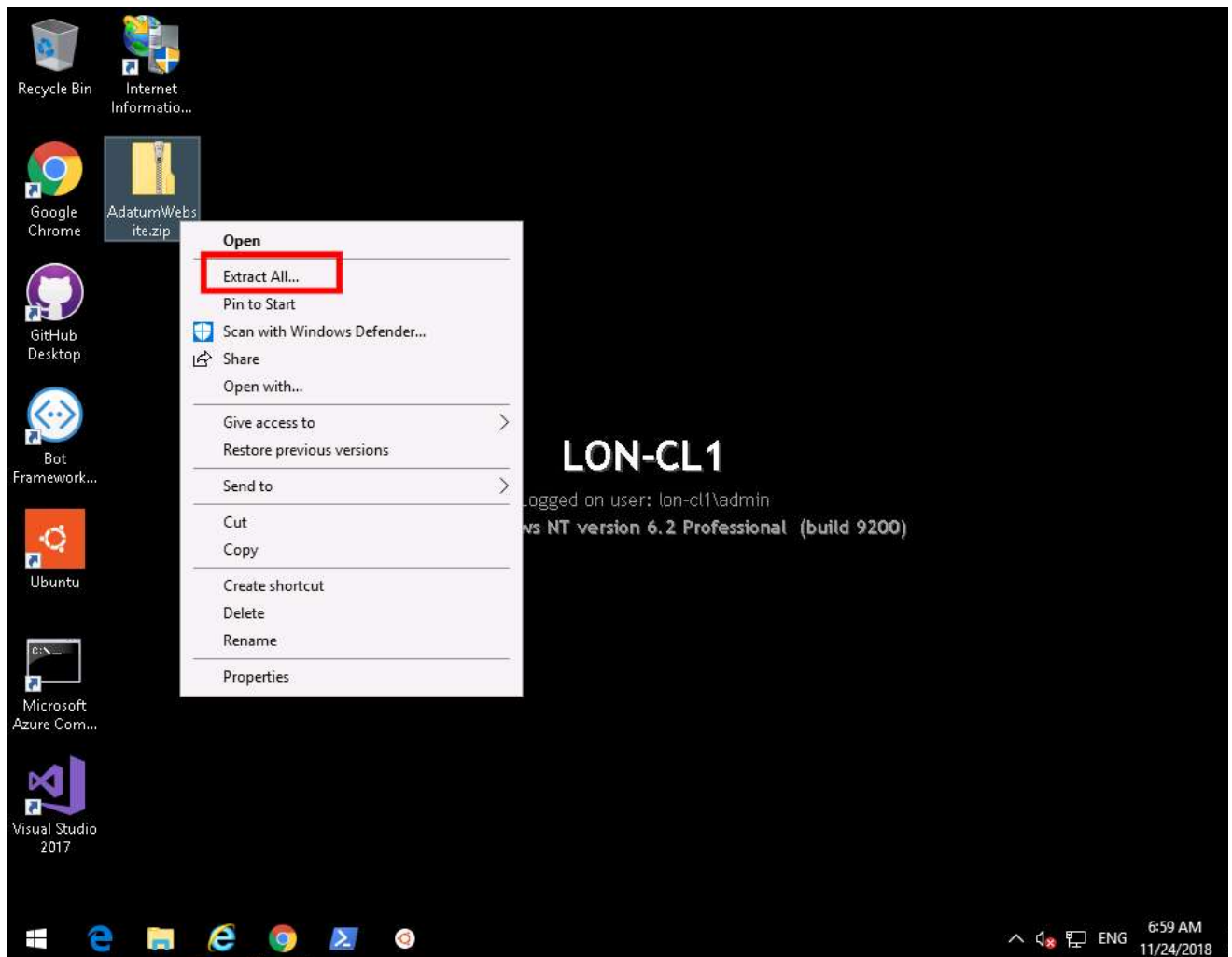
Web Apps not only adds the power of Microsoft Azure to your application, such as security, load balancing, autoscaling, and automated management. You can also take advantage of its DevOps capabilities, such as continuous deployment from Azure DevOps, GitHub, Docker Hub, and other sources, package management, staging environments, custom domain, and SSL certificates.

With App Service, you pay for the Azure compute resources you use. The compute resources you use is determined by the *App Service plan* that you run your Web Apps on.

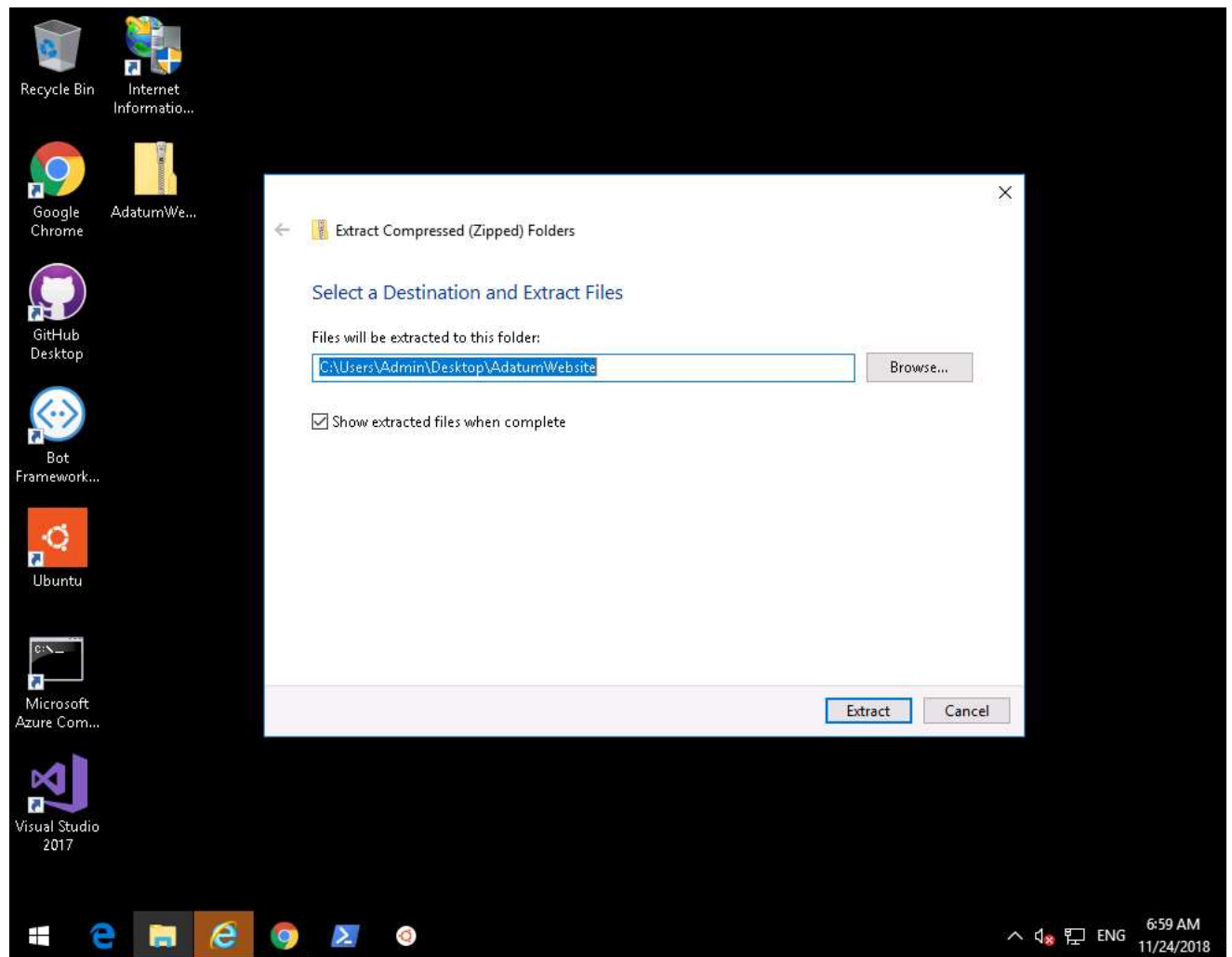
Exercise 1 - Practice - Deploying Web Apps (Azure Portal)

Task 0: Lab Preparation

- ☐ 1. Open a browser and navigate to the following URL <https://aka.gd/AZ303Website>: (Note this is case sensitive)
- ☐ 2. Save the file to your desktop.
- ☐ 3. Right click the file and click **Extract All...**



- ☐ 4. Click **Extract**.



Task 1: Create a web app

- ☐ 1. Open a browser, browse to the Azure portal at <http://portal.azure.com> and use username sheikhnasir9PU9@gdcs1.com and password [fMxJ7unlvdo3Up0u](#).
- ☐ 2. In the top-left corner of the portal, click + **Create resource**, and then click **Web**.
- ☐ 3. On the New blade, click **Create Web App**.

New

Azure Marketplace

See all

Featured

See all

Get started

Recently created

AI + Machine Learning

Analytics

Blockchain

Compute

Containers

Databases

Developer Tools

DevOps

Identity

Integration

Internet of Things

Media

Mixed Reality

IT & Management Tools


Networking


Software as a Service (SaaS)


Security


Storage


Web


 **Web App**
[Quickstarts + tutorials](#)


 **Logic App**
[Quickstarts + tutorials](#)


 **App Service Plan**
[Learn more](#)


 **App Service Environment**
[Learn more](#)


 **API Management**
[Quickstarts + tutorials](#)

 **CDN**
[Quickstarts + tutorials](#)

 **Media Services**
[Quickstarts + tutorials](#)

 **Azure Cognitive Search**
[Quickstarts + tutorials](#)

 **ClickCompliance (preview)**
[Learn more](#)

 **Sitecore® Experience Cloud**
[Learn more](#)

- ☐ 4. On the Web App blade, under **Resource Group**, select **myRG-83DE1E9MPB** and click **OK**.
- ☐ 5. On the Web App blade, in the **Web App name** text box, type a unique name. If the name is unique and valid, a green check mark appears.
- ☐ 6. Ensure **Code** is selected and select **ASP.NET v4.8 or later** as the Runtime stack.
- ☐ 7. Click **Windows** and change the location to **East US**.

Web App

Basics Monitoring Tags Review + create

App Service Web Apps lets you quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. Meet rigorous performance, scalability, security and compliance requirements while using a fully managed platform to perform infrastructure maintenance. [Learn more](#)

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ go deploy - Dev Test Subs

Resource Group * ⓘ myRG-KY8CHK4LC9
[Create new](#)

Instance Details

Name * godeployazurelab .azurewebsites.net

Publish * Code Docker Container

Runtime stack * ASP.NET V4.7

Operating System * Linux Windows

Region * East US
ⓘ Not finding your App Service Plan? Try a different region.

- ☐ 8. In the the **App Service plan** section, click **Create New**.
- ☐ 9. On the New App Service Plan box, in the **App Service plan** text box, type **myAppServicePlan** and click **OK**.

App Service Plan

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. [Learn more](#)

Windows Plan (East US) * ⓘ (New) ASP-myRGKY8CHK4LC9-b027
[Create new](#)

Sku and size *

New App Service Plan

Name * myAppServicePlan

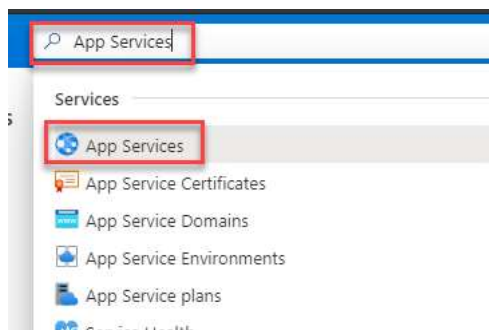
OK Cancel

[Review + create](#) < Previous

- ☐ 10. The **Sku and size**, should automatically be set to **Standard S1** if not select it.
- ☐ 11. On the New App Service Plan blade, click **Review + create**.
- ☐ 12. On the summary screen, click **Create**. Wait until the web app is provisioned.

Task 2: Add a deployment slot

- ☐ 1. At the top of the Azure Portal search for and select **App Services**.



- ☐ 2. On the App Services blade, click the **Web App** that you created in the first task.

-
- The screenshot shows the Azure portal interface. In the left-hand navigation pane, the 'Deployment' section is expanded, and the 'Deployment slots' option is highlighted with a red rectangle. Other visible options in the navigation pane include Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Events, Quickstart, and Deployment Center. The main content area on the right shows details for a resource group, including links to 'Browse', 'Click here to a...', 'Resource group (ch...', 'Status', 'Location', 'Subscription (chang...', 'Subscription ID', and 'Tags (change)'. At the bottom right, there is a 'Diagnose' button with the text 'Our self-servi helps you ide'.

- ## Add a slot

Name
Staging

Clone settings from:
Do not clone settings

- ```
Get-AzWebApp -ResourceGroupName 'myRG-83DE1E9MPB'
```

- ```
PowerShell | ? | ⚙ | 📄 | 📁 | {} | 📄
PS /home/test_studentx7uox> Get-AzWebApp -ResourceGroupName 'myRG-KY8CHK4LC9'

Name                               State      ResourceGroup      EnabledHostNames
----                               -
godeployazurelab                  Running    myRG-KY8CHK4LC9    {godeployazurelab.azurewebsites.net, ...}

PS /home/test_studentx7uox>
```

- ```
Get-AzWebAppSlot -ResourceGroupName 'myRG-83DE1E9MPB' -Name 'Name of your web app'
```

- ☐ 11. Verify that the web app staging slot you created in this task is listed in the output

```
PowerShell | ? | ? | ? | ? | ? | ?
PS /home/test_studentx7uox> Get-AzWebAppSlot -ResourceGroupName 'myRG-KY8CHK4LC9' -Name 'godeployazurelab'

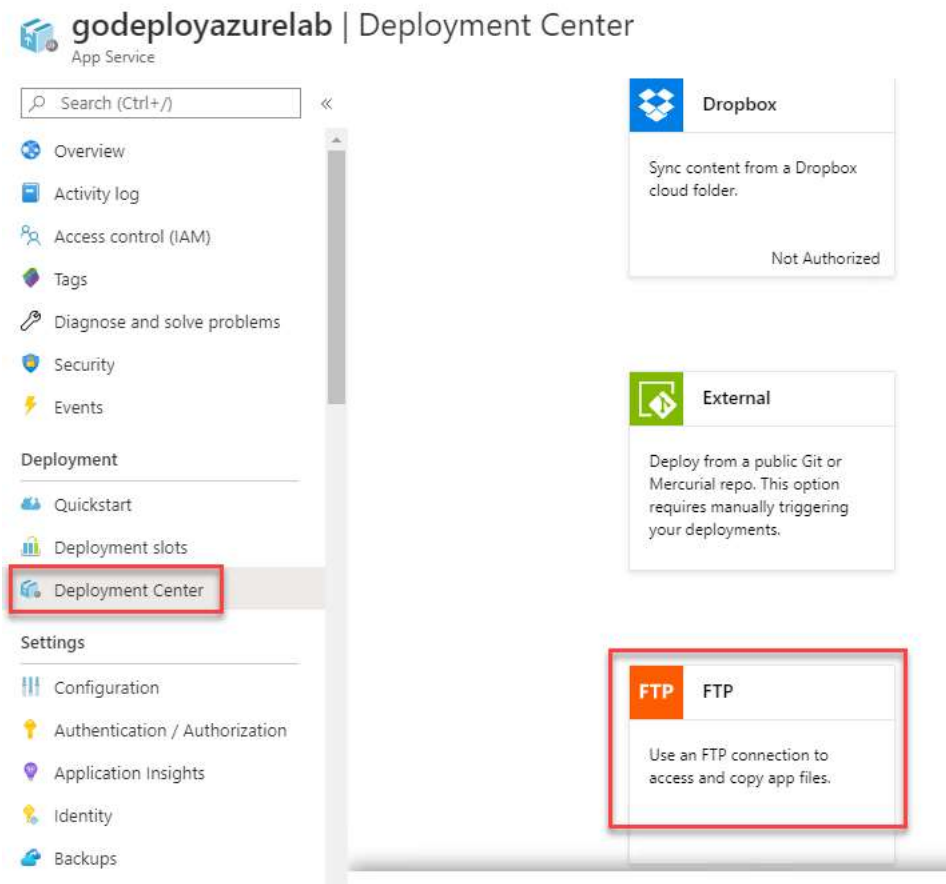
Name State ResourceGroup EnabledHostNames
---- -
godeployazurelab/Staging Running myRG-KY8CHK4LC9 {godeployazurelab-staging.azurewebsite

PS /home/test_studentx7uox>
```

- ☐ 12. Close the Cloud Shell.

### Task 3: Configure deployment credentials

- ☐ 1. In the Azure portal, in the web app that you created in the first task, click the **Deployment Center** then click **FTP Credentials** and then click **Dashboard**.



- ☐ 2. In the FTP dashboard, Scroll down to **User Credentials**, and in the **user name** text box, type **FTPUserXXXXXXXX** (Where XXXXXXXX is a unique number).
- ☐ 3. In the **Password** text box, type a unique password **Save credentials** ensuring you remember the password you typed.

App Service enables you to access your app content through FTP/S. [Learn more](#)

FTPS Endpoint  [Copy](#)

### App Credentials

### User Credentials

User Credentials are defined by you, the user, and can be used with all the apps to which you have access. These credentials can be used with FTP, Local Git and WebDeploy. [Learn more](#)

Username

Password

Confirm Password

Save Credentials

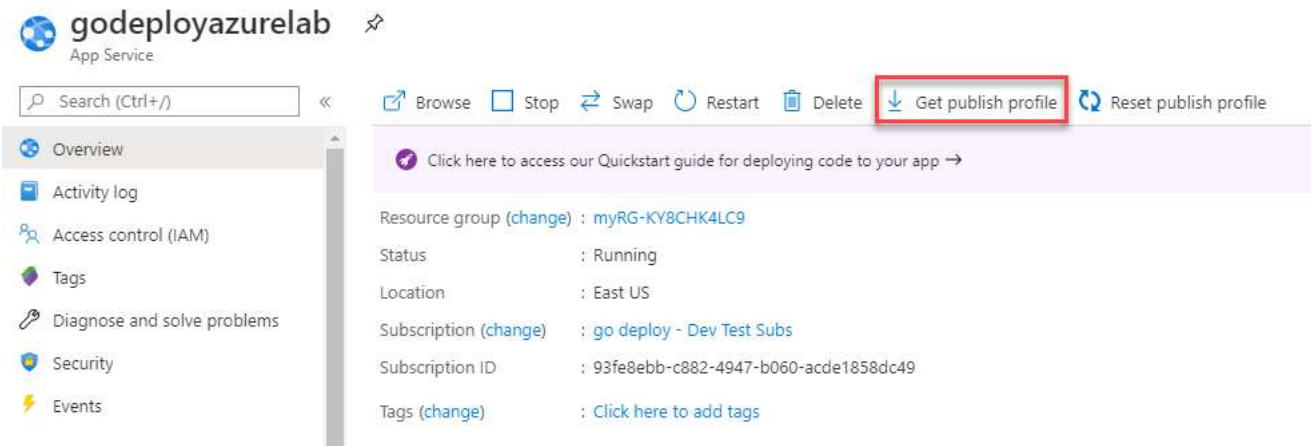
- ☐ 4. Click **Save**

✓ **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

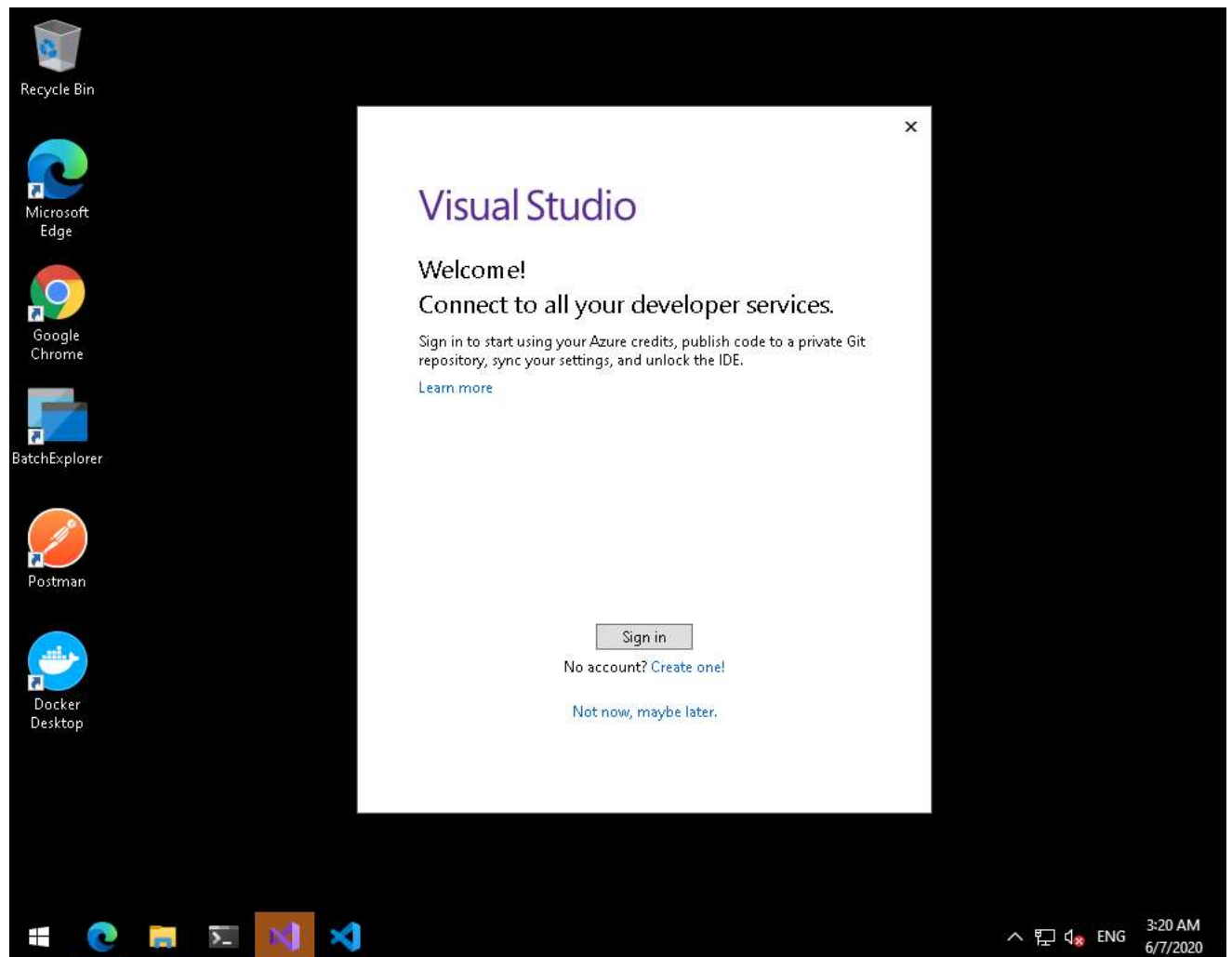
- ☐ 1. In the **Azure portal**, on the blade of the web app that you created in the previous practice, click **Overview** and then click **Get publish profile**. You might need to first click ... **More** if the **Get publish profile** option does not appear in the toolbar at the top of the blade.



- ☐ 2. When prompted whether to open or save the **.PublishSettings** file, click **Save**. browser will save the publishing profile in the **Downloads** folder on your lab computer.
- ☐ 3. On your Virtual Machine, click **Start**, in the Start menu, click **Visual Studio 2022**

⚠ **Note:** You may need to sign in with your Azure subscription credentials. If you do not have Visual Studio installed you can install the free community edition from this link <https://visualstudio.microsoft.com/downloads/>

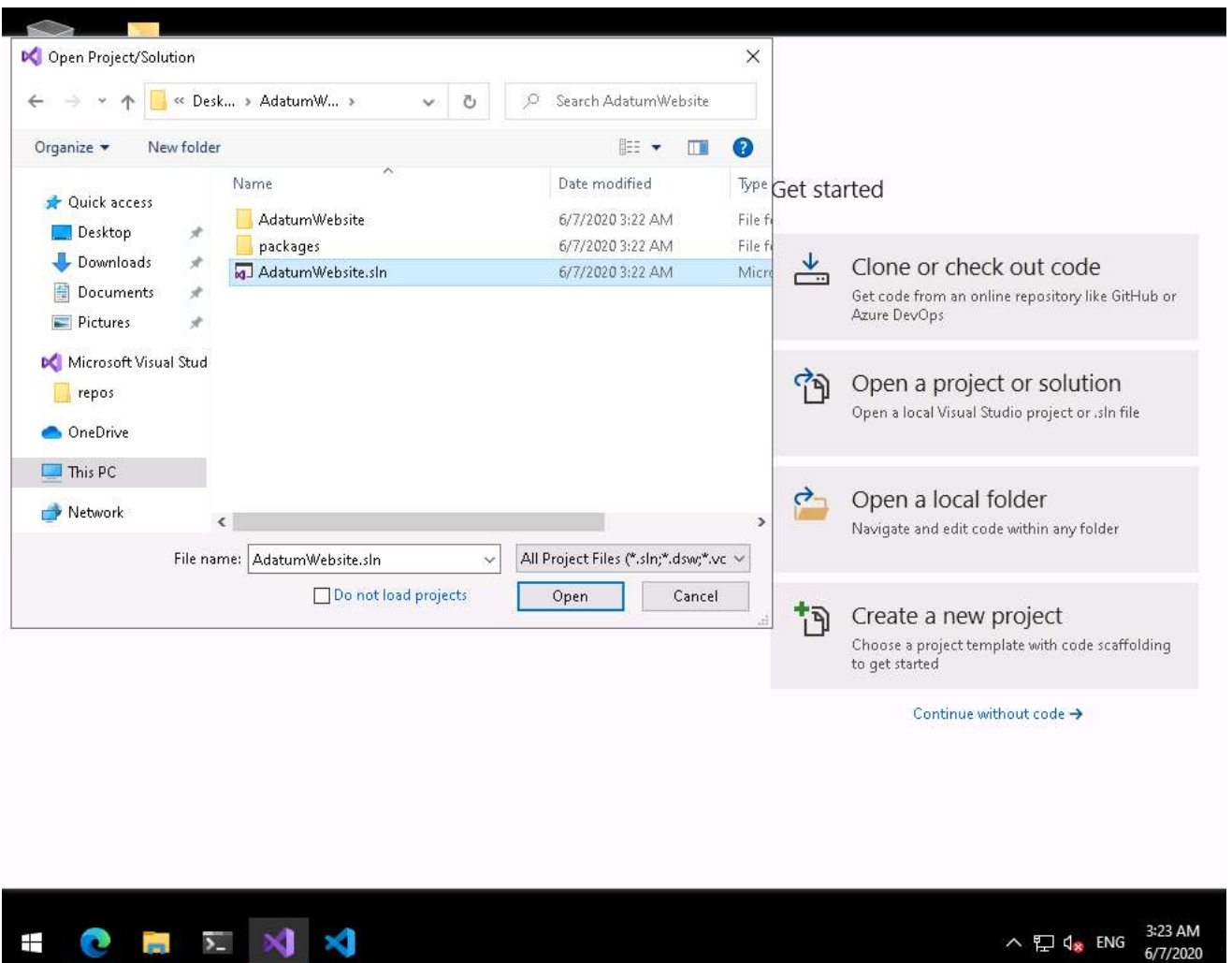
- ☐ 4. When prompted click Sign In and enter the username  **sheikhnasir9PUJ9@gdcs1.com** and password  **fMxJ7unlvdo3Up0u**



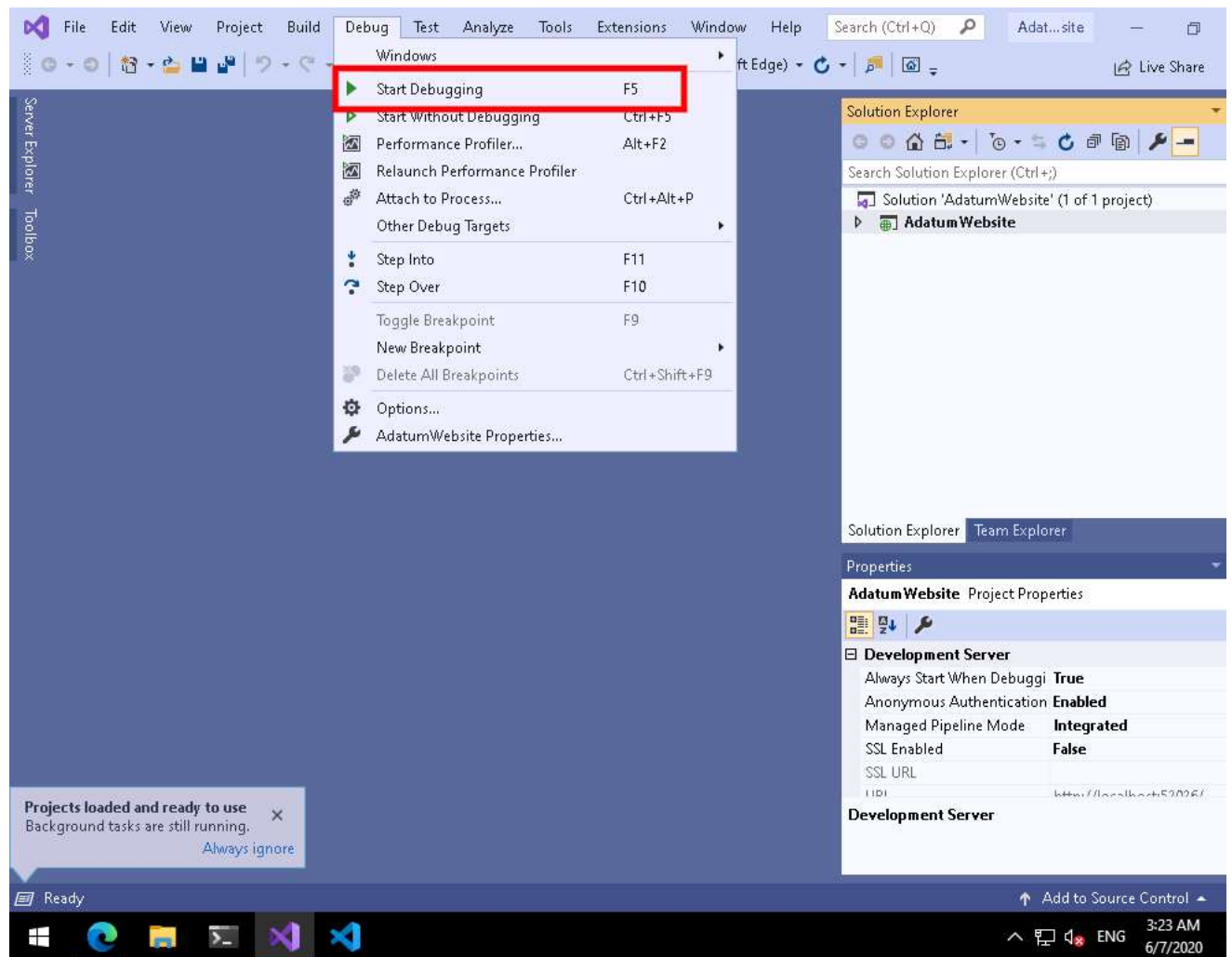
- ☐ 5. On the **Get Started** screen, click **Open a project or solution**. Browse to the folder **NewAdatumWebsite** on your desktop and inside the **AdatumWebsite**, click **AdatumWebsite.sln**, and then click **Open**.

If prompted, in the **Security Warning for AdatumWebsite** dialog box, clear the checkbox **Ask me for every project in this solution** and click **OK**.

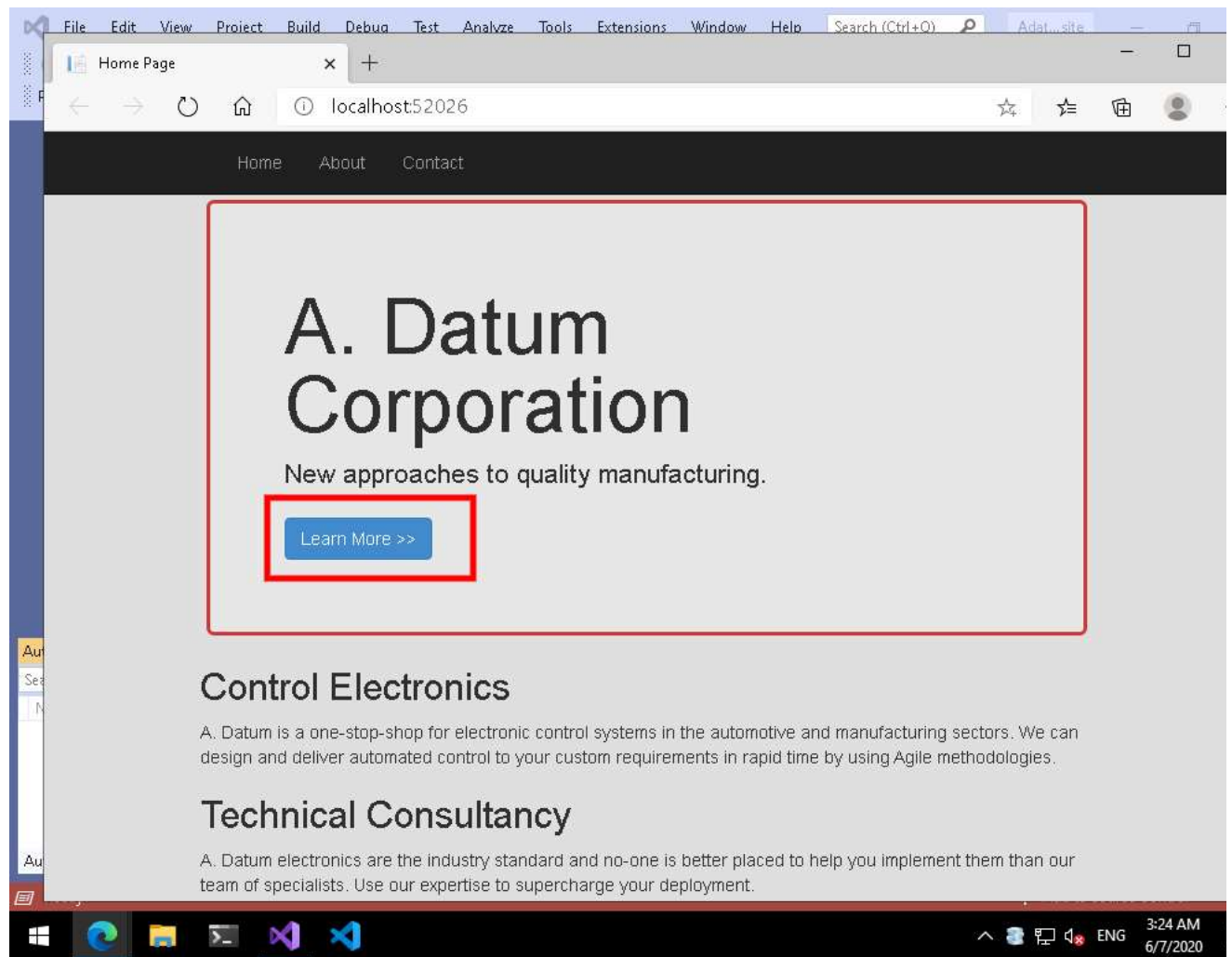




- ☐ 6. In the **Debug** menu, click **Start Debugging**. This will open a new tab in the browser window.



- ☐ 7. On the new browser tab, under **A. Datum Corporation**, click **Learn More**.



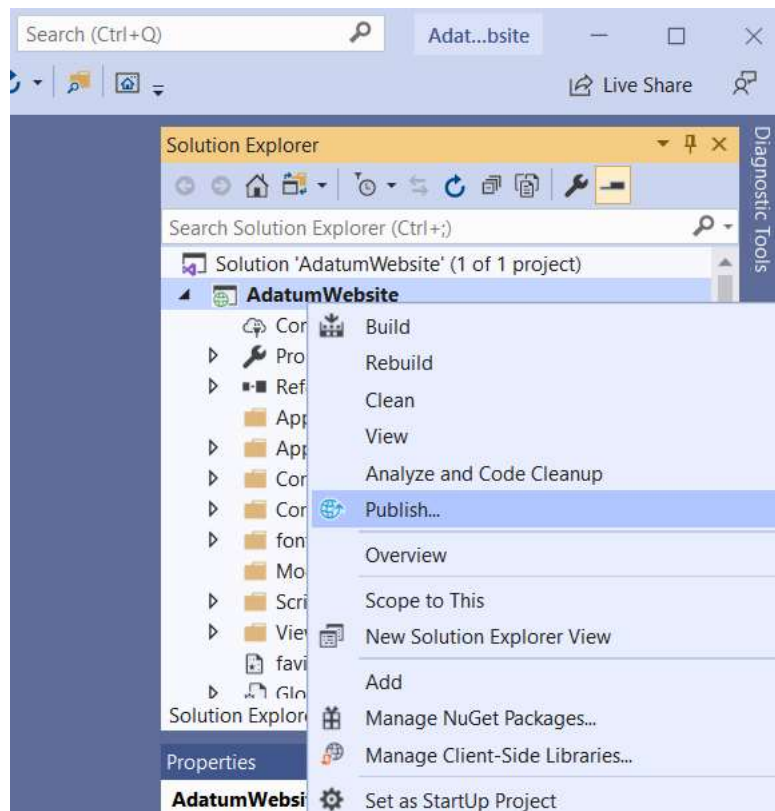
☐ 8. In the browser window, click **Contact**.

☐ 9. Close the new browser tab.

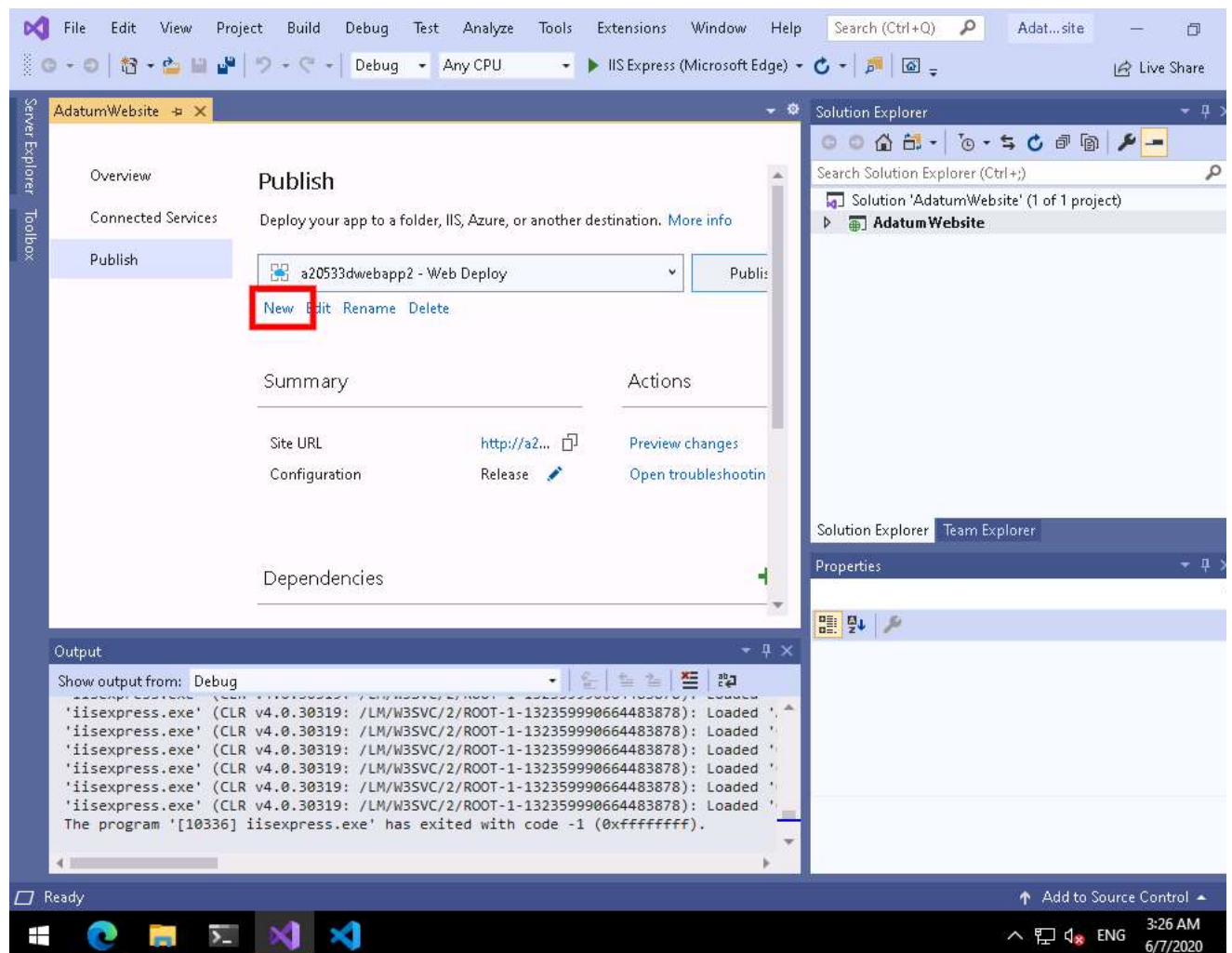
### Task 2: Deploy a web app

☐ 1. In Visual Studio, click **Debug** and, in the **Debug** menu, click **Stop Debugging**.

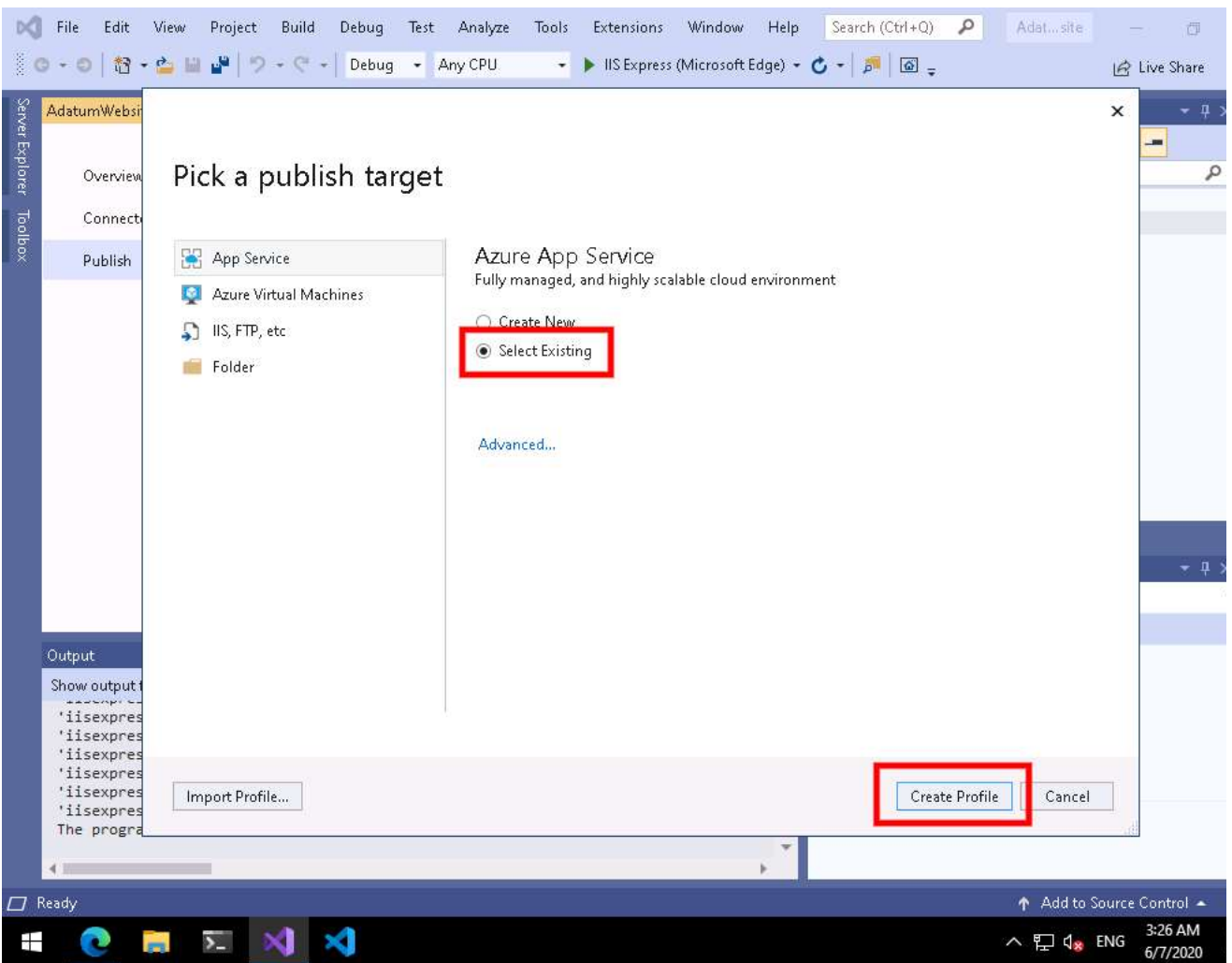
☐ 2. In the Solution Explorer, right-click the **AdatumWebsite** project, and then click **Publish**.



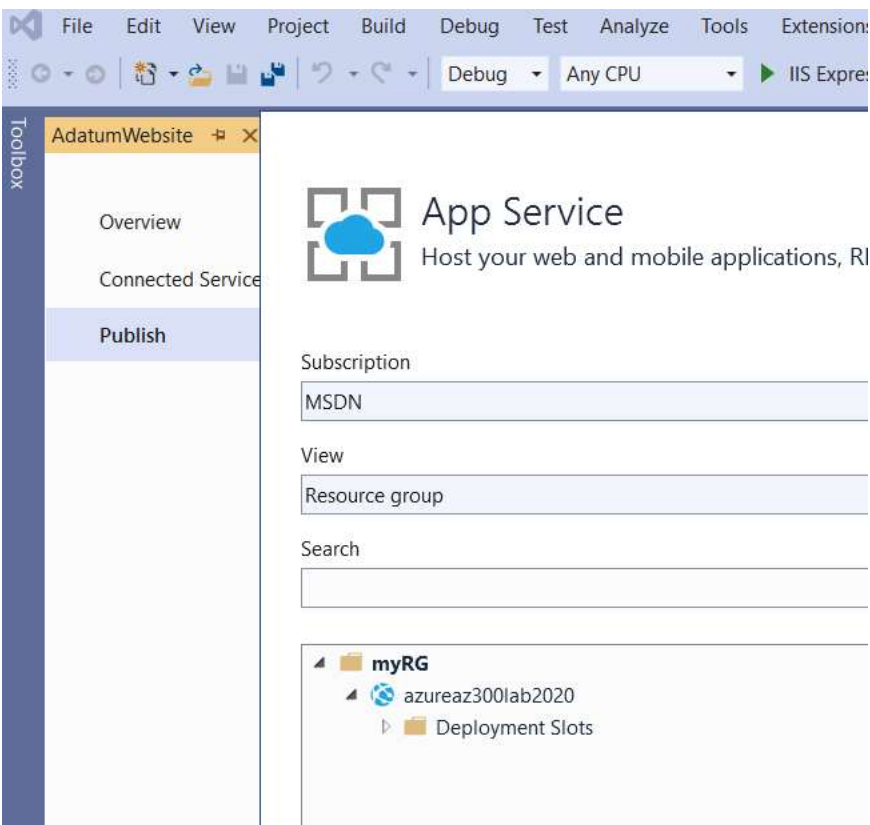
☐ 3. Select **New**



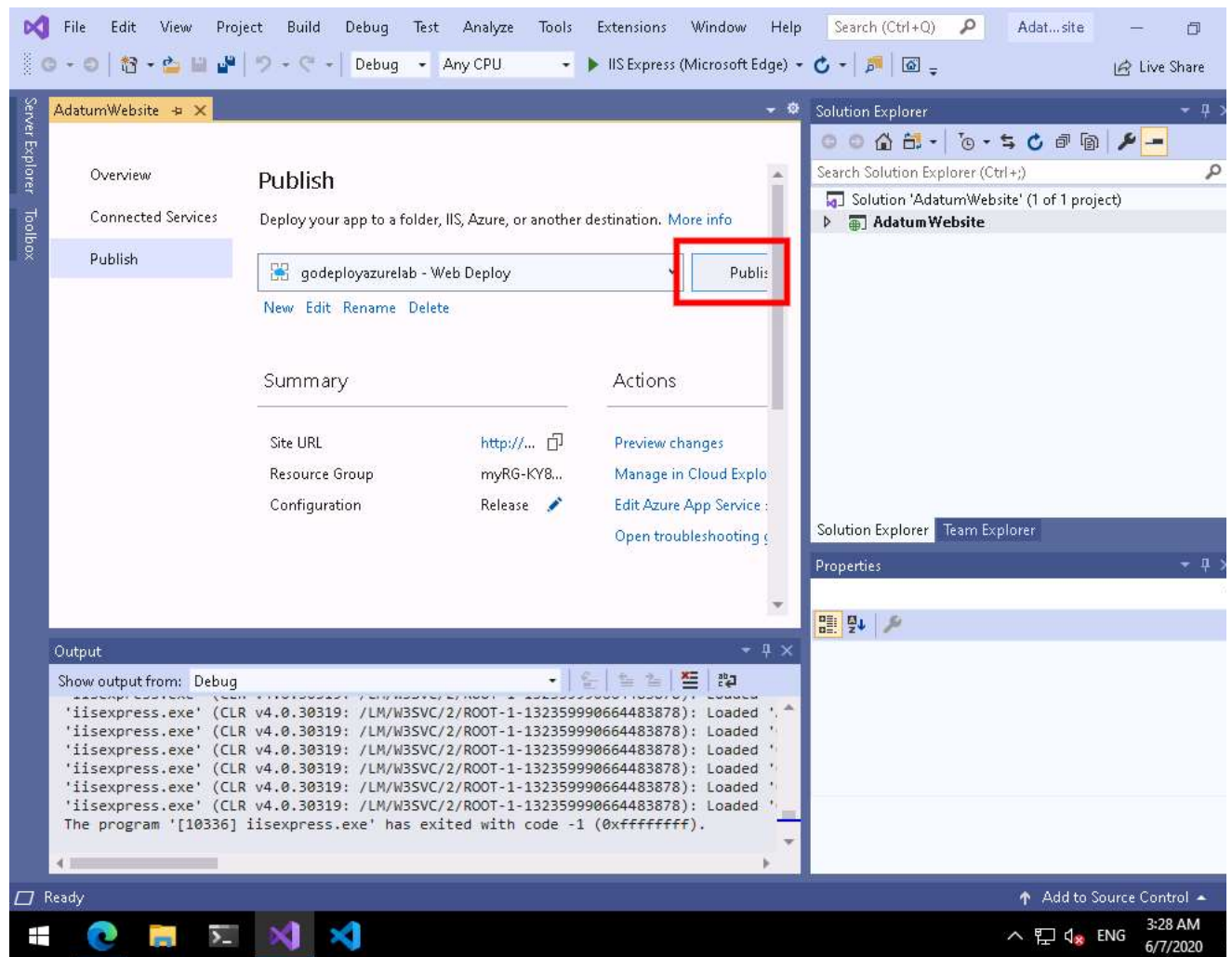
☐ 4. On the pick a publish target page **select existing** click **create profile**



- ☐ 5. Select your resource group and your web app



- ☐ 6. Click **Publish**.



- ☐ 7. Leave Visual Studio open.
- ☐ 8. Close the browser tab with the new Adatum Website.

✓ **Result:** After completing this practice, 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 a browser, in the Azure portal, navigate to the blade of the web app you created in the first practice.
- ☐ 2. On the web app blade, in the **Deployment** section, click **Deployment slots**.



Search (Ctrl+/) Save Discard Add Slot Swap Logs Refresh

Overview  
Activity log  
Access control (IAM)  
Tags  
Diagnose and solve problems  
Security  
Events  
Deployment  
Quickstart  
**Deployment slots**  
Deployment Center  
Settings

### Deployment Slots

Deployment slots are live apps with their own hostnames. App content and configurations elements can be swapped between two deployment slots, including the production slot.

| NAME                                  | STATUS  | APP SERVICE PLAN | TRAFFIC % |
|---------------------------------------|---------|------------------|-----------|
| godeployazurelab<br><b>PRODUCTION</b> | Running | myAppServicePlan | 100       |
| godeployazurelab-Staging              | Running | myAppServicePlan | 0         |

- ☐ 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.

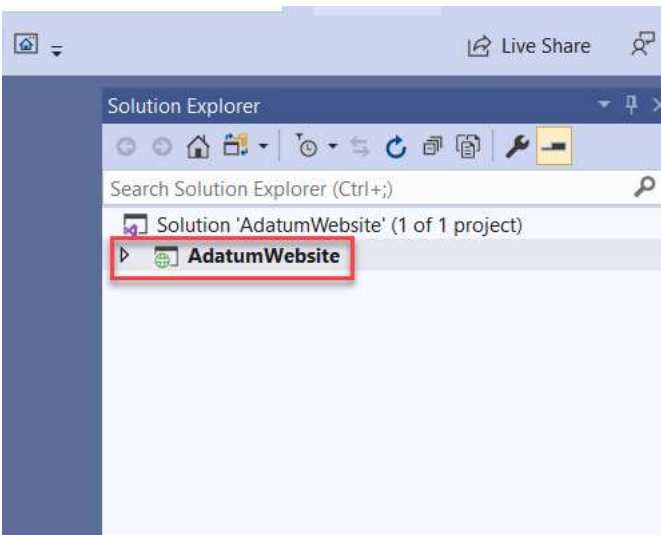
Staging (godeployazurelab/Staging) App Service (Slot)

Search (Ctrl+/) Browse Stop Swap Restart Delete **Get publish profile** Reset publish profile

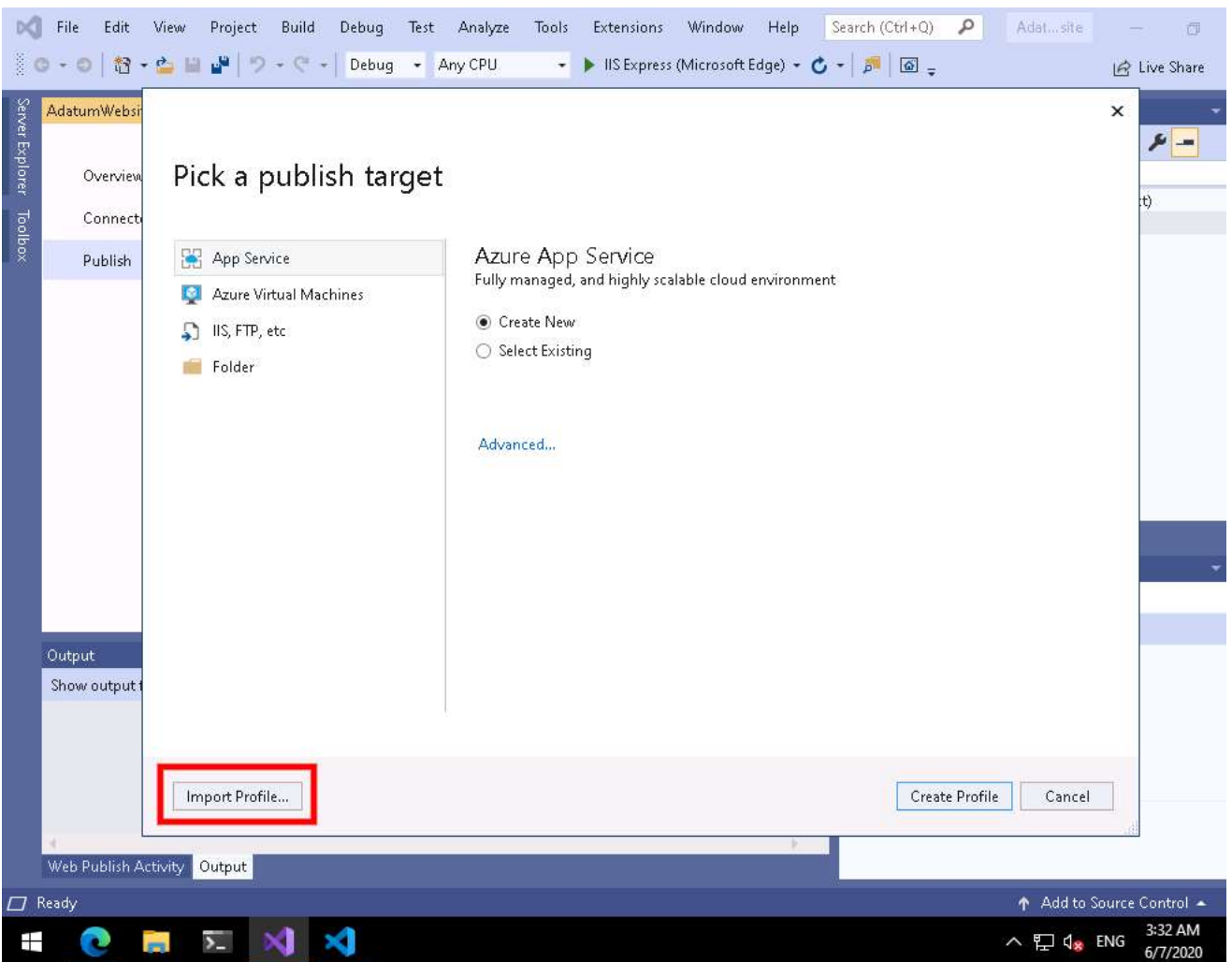
Click here to access our Quickstart guide for deploying code to your app →

|                                                         |                                                                                                                            |
|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| Resource group (change)<br>myrg-ky8chk4lc9              | URL<br><a href="https://godeployazurelab-staging.azurewebsites.net">https://godeployazurelab-staging.azurewebsites.net</a> |
| Status<br>Running                                       | App Service Plan<br>myAppServicePlan (S1: 1)                                                                               |
| Location<br>East US                                     | FTP/deployment username<br>godeployazurelab_Staging\FTPUser25084294                                                        |
| Subscription (change)<br>go deploy - Dev Test Subs      | FTP hostname<br>ftp://waws-prod-blu-159.ftp.azurewebsites.window...                                                        |
| Subscription ID<br>93fe8ebb-c882-4947-b060-acde1858dc49 | FTPS hostname<br>https://waws-prod-blu-159.ftp.azurewebsites.windo...                                                      |
| Tags (change)<br><a href="#">Click here to add tags</a> |                                                                                                                            |

- ☐ 5. The file will be saved to your downloads folder.
- ☐ 6. Switch to **Visual Studio**.
- ☐ 7. On the **File** menu, click **Open**, and then click **Project/Solution**.
- ☐ 8. Browse to the folder **AdatumWebsite** on your desktop and then navigate to the **NewAdatumWebsite** folder.
- ☐ 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**.

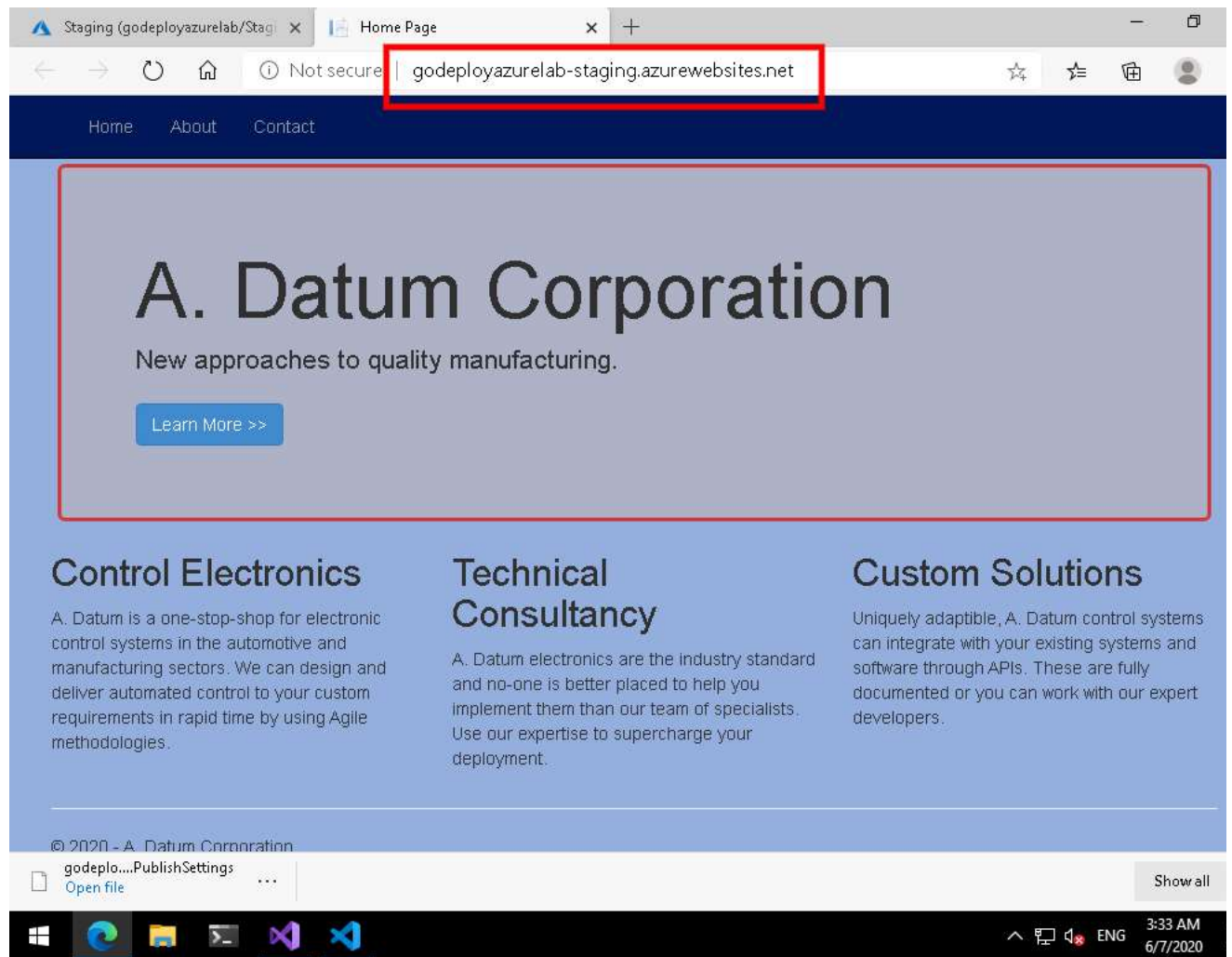


- ☐ 11. On the Publish page, select **New ...**
- ☐ 12. Click **Import Profile**.



- ☐ 13. In the **Import Publish Settings** dialog box, browse to the **Downloads** folder.
- ☐ 14. Select the **XXXXXXXX(staging).PublishSettings** file that you downloaded, click **Open** then click **Publish**. The staging Web App 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 browser window displaying it.
- ☐ 15. Verify that A. Datum's web app is open in a new browser tab and verify the web app's URL.

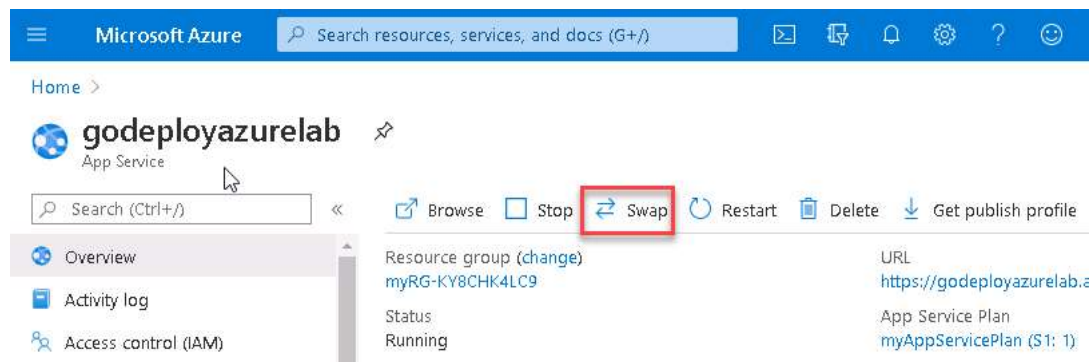




- ☐ 16. Close the new tab.
- ☐ 17. Leave Visual Studio open.

## Task 2: Swap deployment slots

- ☐ 1. In the Azure portal, navigate to the web app blade and, in the **Overview** blade, click the **URL** link for your web app. This will open another tab in a new browser 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. In the **Source** drop-down list, ensure that **Staging** is selected.
- ☐ 5. In the **Destination** drop-down list, ensure that the original WebApp is selected, and then click **Swap**.

# Swap


Source

godeployazurelab-Staging

Target

PRODUCTION

godeployazurelab

 Swap with preview can only be used with sites that have depl

☐ Perform swap with preview

### Config Changes

This is a summary of the final set of configuration changes on the source after the swap has completed.

| Source Changes       |            |           |
|----------------------|------------|-----------|
| SETTING              | TYPE       | OLD VALUE |
| PhpVersion           | General    | 5.6       |
| WEBSITE_NODE_DEF...  | AppSetting | 6.9.1     |
| APPINSIGHTS_INSTR... | AppSetting | Not set   |

Swap

Close

- ☐ 6. Wait until swap operation completes.
- ☐ 7. On the web app blade, in the **Overview** section/blade, click the URL link for your web app. This will open another tab in a new browser tab. Note the color scheme. Notice that the color scheme has changed.
- ☐ 8. 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. Wait until Swap operation completes.
- ☐ 3. On the web app blade, in **Overview** blade, click the URL link for your web app. This will open another tab in a new browser tab. Notice that the color scheme reverted to the original one.
- ☐ 4. Close the A. Datum tab in the browser.

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