

The note after Exercise 1, Task 6, Step 9 and Exercise 2, Task 3, Step 8 is incorrect. It should state "Replace the <name-of-your-api-app> placeholder..."

At the end of this lab, you can skip the **Clean Up** exercise directing you to remove the resources from your Subscription or Resource Group(s). The clean up is handled automatically, after ending your lab.

Lab: Building a web application on Azure platform as a service offerings

Student lab manual

Lab scenario

You're the owner of a startup organization and have been building an image gallery application for people to share great images of food. To get your product to market as quickly as possible, you decided to use Microsoft Azure App Service to host your web apps and APIs.

Objectives

After you complete this lab, you will be able to:

- Create various apps by using App Service.
- Configure application settings for an app.
- Deploy apps by using Kudu, the Azure Command-Line Interface (CLI), and zip file deployment.

Lab setup

- Estimated time: **45 minutes**

Instructions

Before you start

Sign in to the lab virtual machine

Ensure that you're signed in to your Windows 10 virtual machine by using the following credentials:

- Username: **Admin**
- Password: **Pa55w.rd**

Review the installed applications

Find the taskbar on your Windows 10 desktop. The taskbar contains the icons for the applications that you'll use in this lab:

- Microsoft Edge
- File Explorer
- Windows PowerShell
- Visual Studio Code

Exercise 1: Build a back-end API by using Azure Storage and the Web Apps feature of Azure App Service

Task 1: Open the Azure portal

1. Sign in to the Azure portal (<https://portal.azure.com>).

Note: If this is your first time signing in to the Azure portal, a dialog box will display offering a tour of the portal. Select **Get Started** to skip the tour.

Task 2: Create a Storage account

1. Create a new storage account with the following details:
 - New resource group: **ManagedPlatform**
 - Name: **imgstor*[yourname]***
 - Location: **(US) East US**
 - Performance: **Standard**
 - Account kind: **StorageV2 (general purpose v2)**
 - Replication: **Locally-redundant storage (LRS)**
 - Access tier: **Hot**
2. Wait for Azure to finish creating the storage account before you move forward with the lab. You'll receive a notification when the account is created.
3. Access the **Access Keys** blade of your newly created storage account instance.
4. Record the value of the **Connection string** text box. You'll use this value later in this lab.

Task 3: Upload a sample blob

1. Access the **imgstor*[yourname]*** storage account that you created earlier in this lab.
2. In the **Blob service** section, select the **Containers** link.
3. Create a new **container** with the following settings:
 - Name: **images**
 - Public access level: **Blob (anonymous read access for blobs only)**
4. Go to the new **images** container, and then use the **Upload** button to upload the **grilledcheese.jpg** file in the **Allfiles (F):\Allfiles\Labs\01\Starter\Images** folder on your lab machine.

Note: We recommended that you enable the **Overwrite if files already exist** option.

Task 4: Create a web app

1. Create a new web app with the following details:
 - Existing resource group: **ManagedPlatform**
 - Web App name: **imgapi*[yourname]***

- Publish: **Code**
- Runtime stack: **.NET Core 3.1 (LTS)**
- Operating System: **Windows**
- Region: **East US**
- New App Service plan: **ManagedPlan**
- SKU and size: **Standard (S1)**
- Application Insights: **Disabled**

2. Wait for Azure to finish creating the web app before you move forward with the lab. You'll receive a notification when the app is created.

Task 5: Configure the web app

1. Access the **imgapi*[yourname]*** web app that you created earlier in this lab.
2. In the **Settings** section, find the **Configuration** section, and then create a new application setting by using the following details:
 - Name: **StorageConnectionString**
 - Value: ***Storage Connection String copied earlier in this lab***
 - Deployment slot setting: **Not selected**
3. Save your changes to the application settings.
4. In the **Settings** section, find the **Properties** section.
5. In the **Properties** section, copy the value of the **URL** text box. You'll use this value later in the lab.

Note: At this point, the web server at this URL will return a 404 error. You have not deployed any code to the Web App yet. You will deploy code to the Web App later in this lab.

Task 6: Deploy an ASP.NET web application to Web Apps

1. Using Visual Studio Code, open the web application in the **Allfiles (F):\Allfiles\Labs\01\Starter\API** folder.
2. Open the **Controllers\ImagesController.cs** file, and then observe the code in each of the methods.
3. Open the Windows Terminal application.
4. Sign in to the Azure CLI by using your Azure credentials:

```
az login
```

5. List all the apps in your **ManagedPlatform** resource group:

```
az webapp list --resource-group ManagedPlatform
```

6. Find the apps that have the **imgapi*** prefix:

```
az webapp list --resource-group ManagedPlatform --query "[?starts_with(name
```

7. Print only the name of the single app that has the **imgapi*** prefix:

```
az webapp list --resource-group ManagedPlatform --query "[?starts_with(name
```

8. Change the current directory to the **Allfiles (F):\Allfiles\Labs\01\Starter\API** directory that contains the lab files:

```
cd F:\Allfiles\Labs\01\Starter\API\
```

9. Deploy the **api.zip** file to the web app that you created earlier in this lab:

```
az webapp deployment source config-zip --resource-group ManagedPlatform --s
```

Note: Replace the | placeholder with the name of the web app that you created earlier in this lab. You recently queried this app's name in the previous steps.

10. Access the **imgapi*[yourname]** web app that you created earlier in this lab. Open the **imgapi*[yourname]*** web app in your browser.
11. Perform a GET request to the root of the website, and then observe the JavaScript Object Notation (JSON) array that's returned. This array should contain the URL for your single uploaded image in your storage account.
12. Close the currently running Visual Studio Code and Windows Terminal applications.

Review

In this exercise, you created a web app in Azure and then deployed your ASP.NET web application to Web Apps by using the Azure CLI and the Kudu zip file deployment utility.

Exercise 2: Build a front-end web application by using Azure Web Apps

Task 1: Create a web app

1. In the Azure portal, create a new web app with the following details:
 - Existing resource group: **ManagedPlatform**
 - Web app name: **imgweb*[yourname]***
 - Publish: **Code**
 - Runtime stack: **.NET Core 3.1 (LTS)**
 - Operating system: **Windows**
 - Region: **East US**
 - Existing App Service plan: **ManagedPlan**
 - Application Insights: **Disabled**
2. Wait for Azure to finish creating the web app before you move forward with the lab. You'll receive a notification when the app is created.

Task 2: Configure a web app

1. Access the **imgweb*[yourname]*** web app that you created in the previous task.
2. In the **Settings** section, find the **Configuration** settings.
3. Create a new application setting by using the following details:
 - Name: **ApiUrl**
 - Value: ***Web app URL copied earlier in this lab***
 - Deployment slot setting: **Not selected**

Note: Make sure you include the protocol, such as **https://**, in the URL that you copy into the **Value** text box for this application setting.

4. Save your changes to the application settings.

Task 3: Deploy an ASP.NET web application to Web Apps

1. Using Visual Studio Code, open the web application in the **Allfiles (F):\Allfiles\Labs\01\Starter\Web** folder.
2. Open the **Pages\Index.cshtml.cs** file, and then observe the code in each of the methods.
3. Open the Windows Terminal application, and then sign in to the Azure CLI by using your Azure credentials:

```
az login
```

4. List all the apps in your **ManagedPlatform** resource group:

```
az webapp list --resource-group ManagedPlatform
```

5. Find the apps that have the **imgweb*** prefix:

```
az webapp list --resource-group ManagedPlatform --query "[?starts_with(name
```

6. Print only the name of the single app that has the **imgweb*** prefix:

```
az webapp list --resource-group ManagedPlatform --query "[?starts_with(name
```

7. Change your current directory to the **Allfiles (F):\Allfiles\Labs\01\Starter\Web** directory that contains the lab files:

```
cd F:\Allfiles\Labs\01\Starter\Web\
```

8. Deploy the **web.zip** file to the web app that you created earlier in this lab:

```
az webapp deployment source config-zip --resource-group ManagedPlatform --s
```

Note: Replace the | placeholder with the name of the web app that you created earlier in this lab. You recently queried this app's name in the previous steps.

9. Access the **imgweb*[yourname]** web app that you created earlier in this lab. Open the **imgweb[yourname]*** web app in your browser.
10. From the **Contoso Photo Gallery** webpage, find the **Upload a new image** section, and then upload the **bahnmi.jpg** file in the **Allfiles (F):\Allfiles\Labs\01\Starter\Images** folder on your lab machine.

Note: Ensure you click the **Upload** button to upload the image to Azure.

11. Observe that the list of gallery images has updated with your new image.

Note: In some rare cases, you might need to refresh your browser window to retrieve the new image.

12. Close the currently running Visual Studio Code and Windows Terminal applications.

Review

In this exercise, you created an Azure web app and deployed an existing web application's code to the resource in the cloud.

Exercise 3: Clean up your subscription

Task 1: Open Azure Cloud Shell

1. In the Azure portal, select the **Cloud Shell** icon to open a new shell instance.
2. If Cloud Shell isn't already configured, configure the shell for Bash by using the default settings.

Task 2: Delete resource groups

1. Enter the following command, and then select Enter to delete the **ManagedPlatform** resource group:

```
az group delete --name ManagedPlatform --no-wait --yes
```

2. Close the Cloud Shell pane in the portal.

Task 3: Close the active applications

- Close the currently running Microsoft Edge application.

Review

In this exercise, you cleaned up your subscription by removing the resource groups used in this lab.

Congratulations!

You have successfully completed this exercise. Click **End** to advance to the next exercise.