Lab 01: Build a web application on Azure platform as a service offering

Microsoft Azure user interface

Given the dynamic nature of Microsoft cloud tools, you might experience Azure UI changes that occur after the development of this training content. As a result, the lab instructions and lab steps might not align correctly.

Microsoft updates this training course when the community alerts us to needed changes. However, cloud updates occur frequently, so you might encounter UI changes before this training content updates. **If this occurs, adapt to the changes, and then work through them in the labs as needed.**

Note: Your instructor will provide instructions to connect to the virtual lab environment.

Review the installed applications

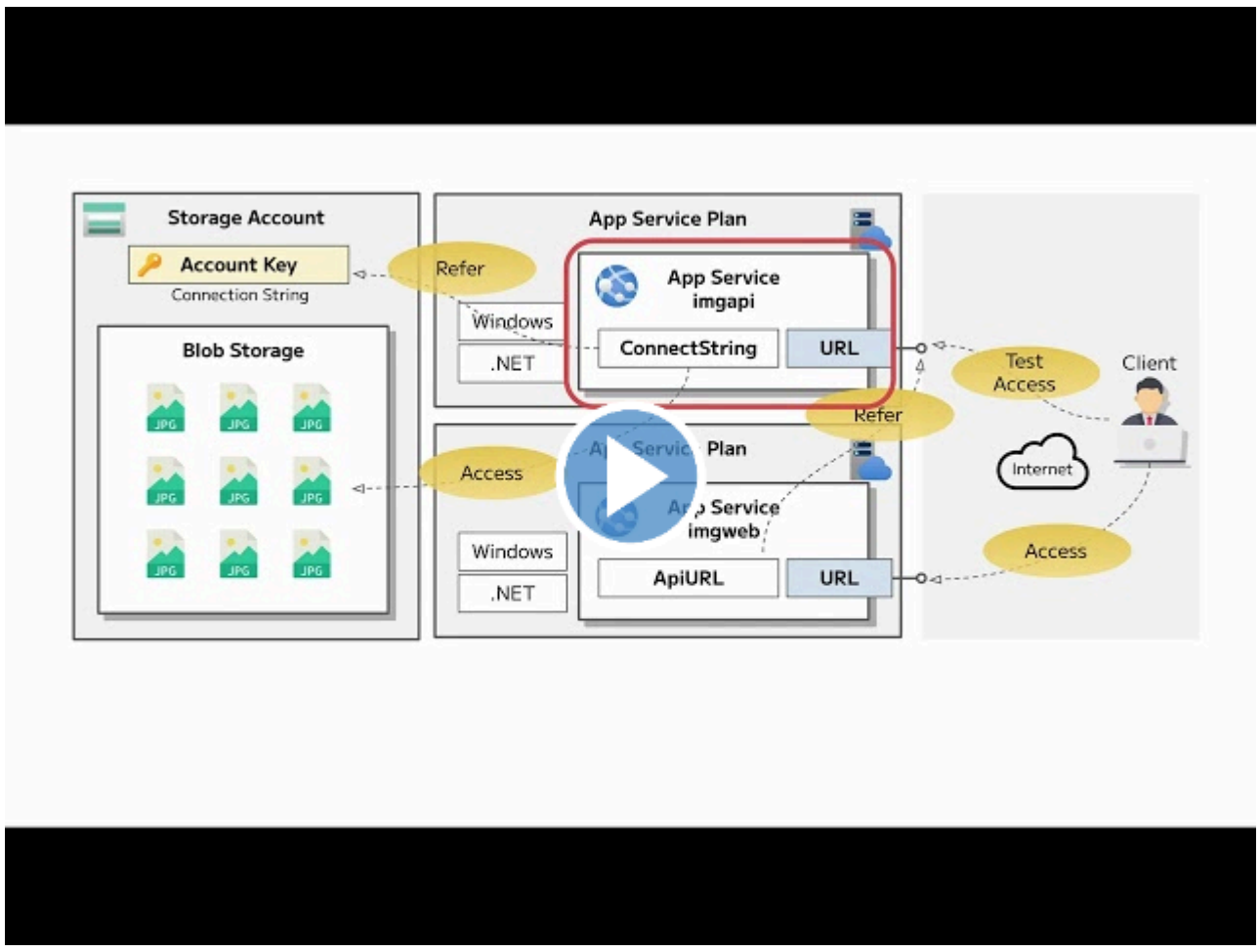
Find the taskbar on your Windows 11 desktop. The taskbar contains the icons for the applications that you'll use in this lab, including:

- Microsoft Edge
- File Explorer
- Terminal
- Visual Studio Code

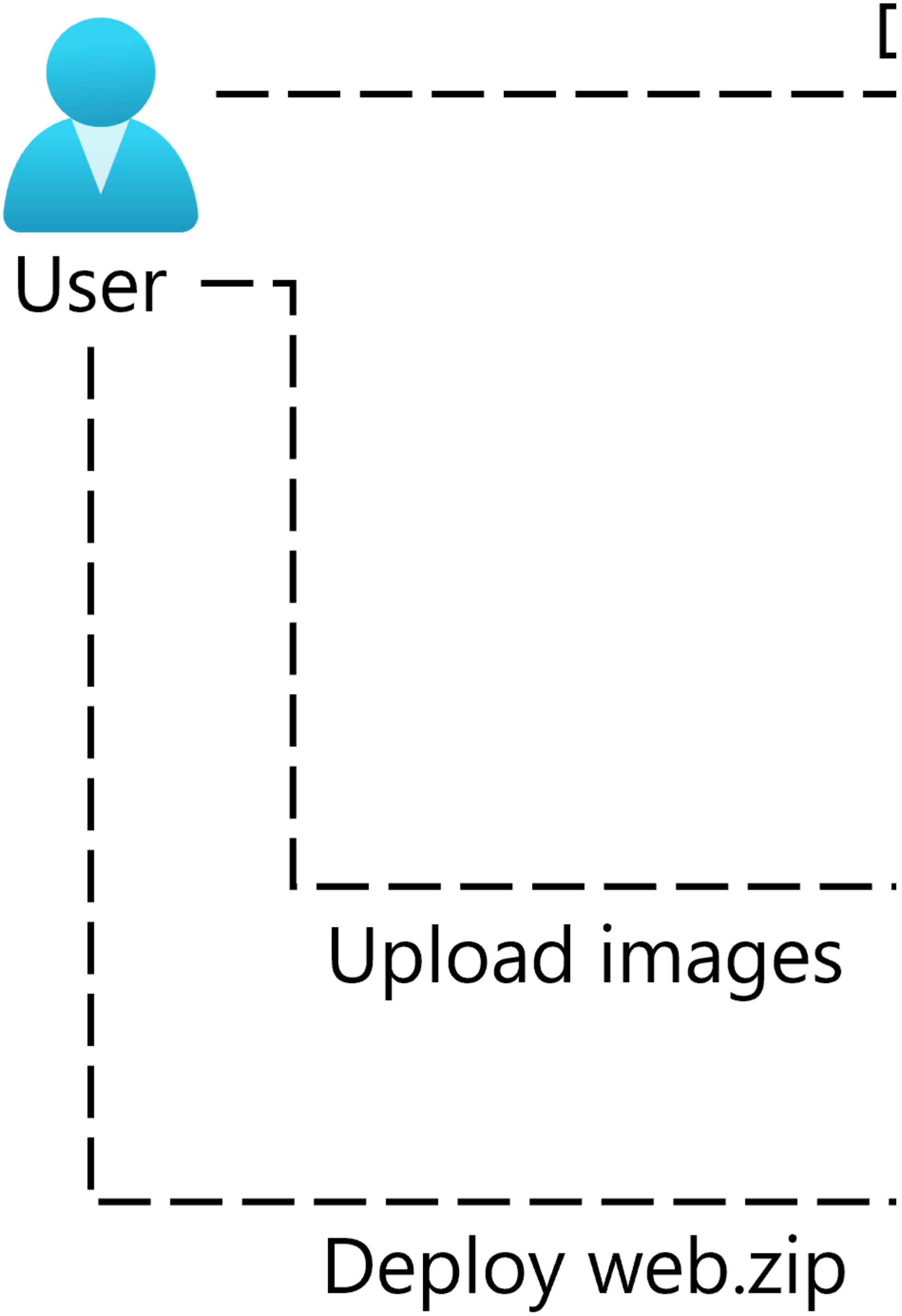
Scenario

In this lab, you will explore how to create a web application on Azure by using the PaaS model. After the web application is created, you will learn how to upload existing web application files by using the Apache Kudu zip deployment option. You will then view and test the newly deployed web application.

*View this video by right-clicking this **video link** and select 'Open link in a new tab / new window'.*



Architecture diagram


User
Upload images
Deploy web.zip

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

Task 0: Download the lab files.

- [ ] 1. From the lab virtual machine, click **Start** and search for **PowerShell** then open **PowerShell as Administrator**.
- [ ] 2. Run the following commands to download the latest version of the lab files to the virtual machine.

```
New-Item -Path "D:\" -Name "Allfiles" -ItemType "directory"
```

```
([System.Net.WebClient]::new()).DownloadFile('https://github.com/MicrosoftLearning/AZ-204-DevelopingSolutionsforMicrosoftAzure/archive/m
```

```
Expand-Archive -Path 'D:\Allfiles\master.zip' -DestinationPath 'D:\Allfiles'
```

```
Move-item -Path "D:\Allfiles\AZ-204-DevelopingSolutionsforMicrosoftAzure-master\Allfiles\*" -Destination "D:\Allfiles" -confirm:$false
```

- [ ] 3. Close the powershell window.

Task 1: Open the Azure portal

- [ ] 1. On the taskbar, select the **Microsoft Edge** icon.
- [ ] 2. In the browser window, browse to the Azure portal **https://portal.azure.com**, and sign in as **studentANDGQ@t001.gdazcs.com** with the password **8bnA6ATjjtq5plok**

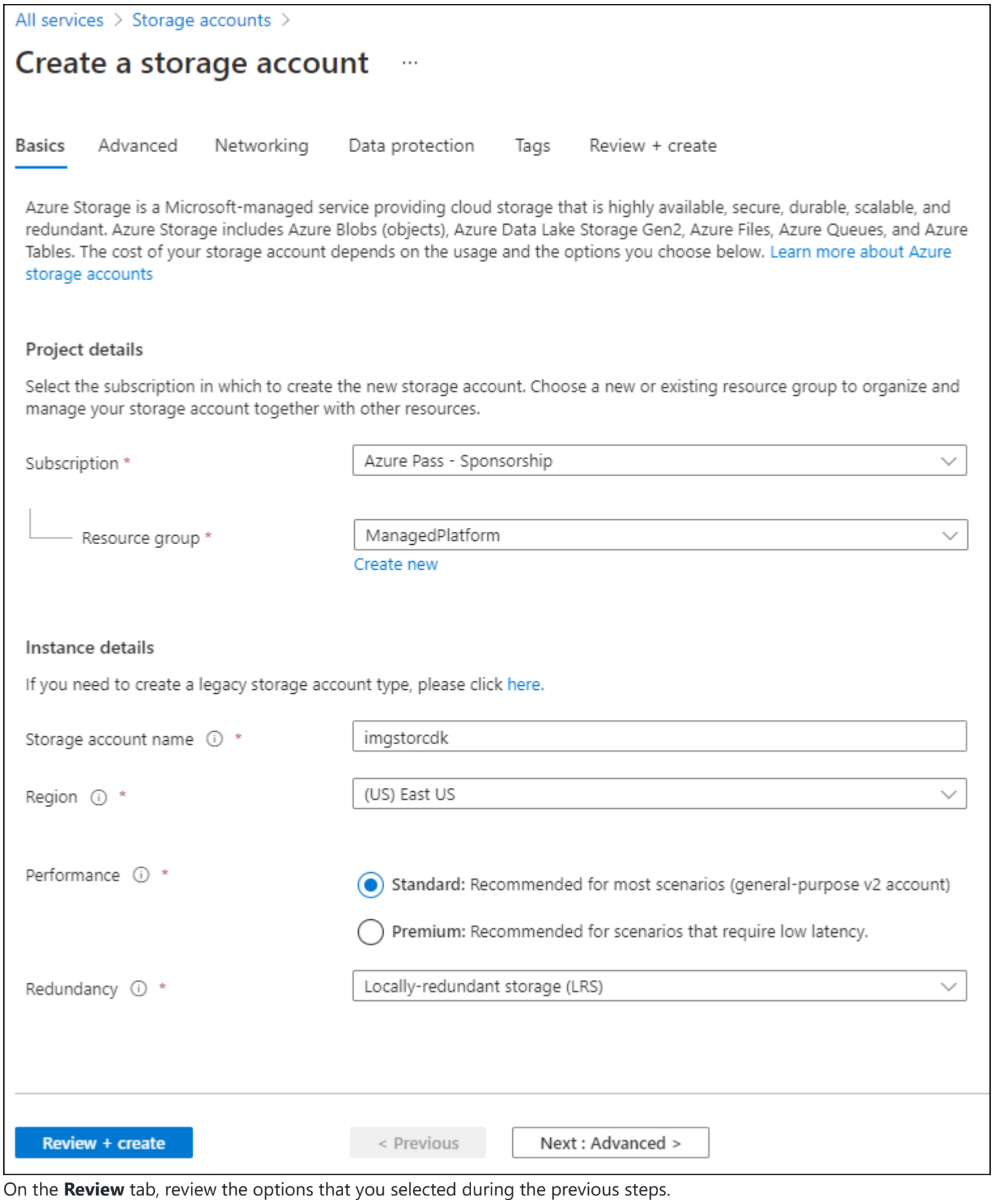
> ⚠ **Note**: If this is your first time signing in to the Azure portal, you'll be offered a tour of the portal. If you prefer to skip the tour, select **Maybe later** to begin using the portal.

Task 2: Create a Storage account

- [ ] 1. In the Azure portal, use the **Search resources, services, and docs** text box to search for **Storage Accounts**, and then in the list of results, select **Storage Accounts**.
- [ ] 2. On the **Storage accounts** blade, select **+ Create**.
- [ ] 3. On the **Create a storage account** blade, on the **Basics** tab, perform the following actions, and then select **Review**:

Setting	Action
Subscription drop-down list	**CloudShare1A**
Resource group section	**ManagedPlatform-MPAYZBHZ5D**
Storage account name text box	Enter **imgstor***[yourname]*
Region drop-down list	Select **(US) East US**
Performance section	Select the **Standard** option
Redundancy drop-down list	Select **Locally-redundant storage (LRS)**
Advanced tab - Security	**Allow enabling anonymous access to individual containers**

The following screenshot displays the configured settings on the **Basics** tab of the **Create a storage account** blade.



- [ ] 4. On the **Review** tab, review the options that you selected during the previous steps.
- [ ] 5. Select **Create** to create the storage account by using your specified configuration.

> ⚠ **Note**: Wait for the creation task to complete before you proceed with this lab.

- [ ] 6. On the **Overview** blade, select the **Go to resource** button to navigate to the blade of the newly created storage account.
- [ ] 7. On the **Storage account** blade, in the **Security + networking** section, select **Access keys**.
- [ ] 8. On the **Access keys** blade, review any one of the keys, and then copy the value of either of the **Connection string** boxes to the clipboard.

> ⚠ **Note**: It doesn't matter which connection string you choose. They are interchangeable.

- [ ] 9. Open Notepad, and then paste the copied connection string value to Notepad. You'll use this value later in this lab.

Task 3: Upload a sample blob

- [ ] 1. On the **Storage Account** blade, in the **Data storage** section, select the **Containers** link.
- [ ] 2. On the **Containers** blade, select **+ Container**.
- [ ] 3. In the **New container** window, perform the following actions:

Setting	Action
Name text box	Enter **images**
Public access level list	Select **Private (no anonymous access)**, and then select **Create**

- [ ] 4. On the **Containers** blade, select the newly created **images** container.
- [ ] 5. On the **images** blade, select **Upload**.
- [ ] 6. In the **Upload blob** window, perform the following actions:

Setting	Action
Files section	Select **Browse for files**
File Explorer window	Browse to **Allfiles (D):\Allfiles\Labs\01\Starter\Images**, select the **grilledcheese.jpg** file, and then select **Open**
Overwrite if files already exist check box	Ensure that the check box is selected, and then select **Upload**

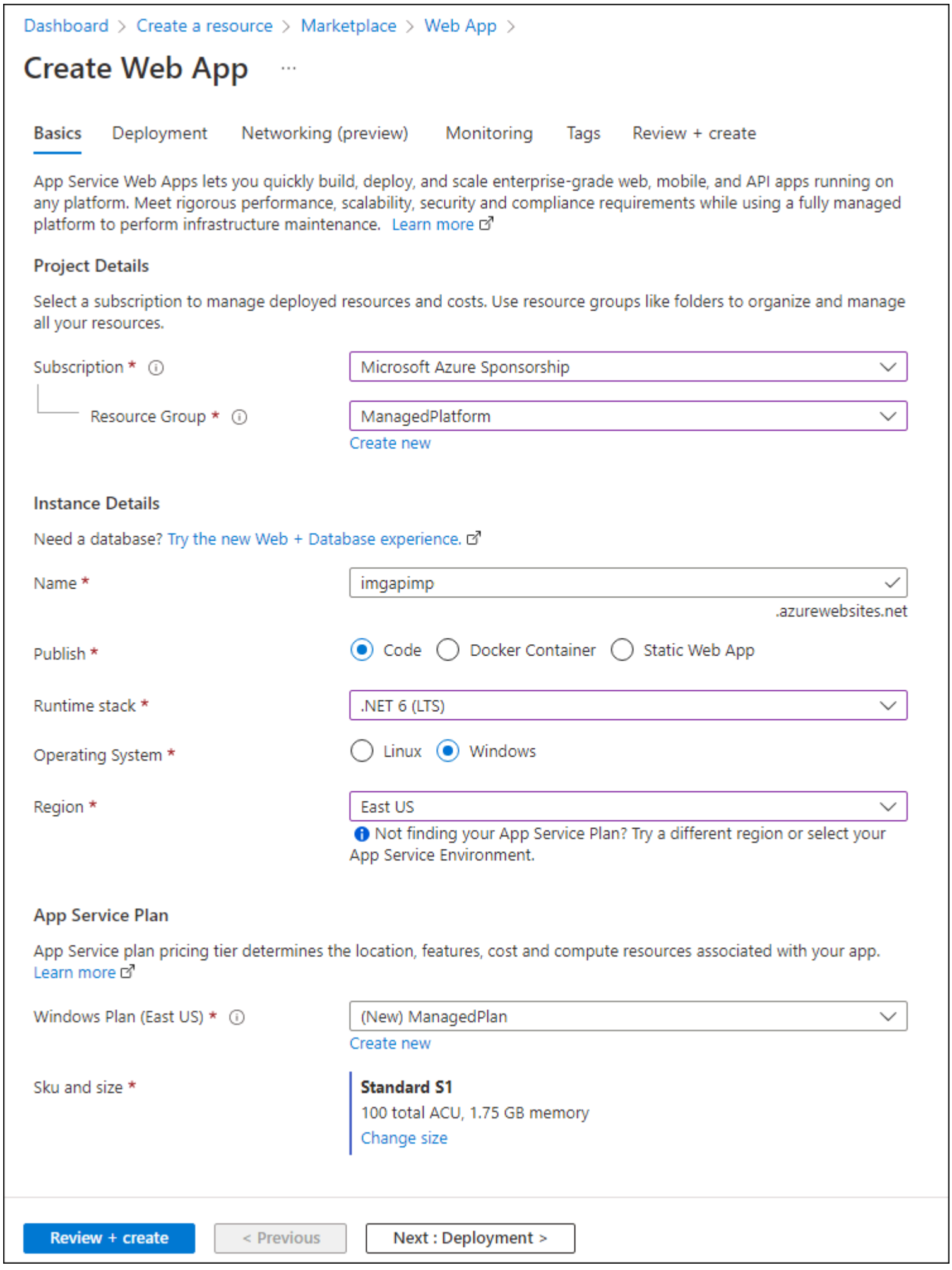
> ⚠ **Note**: Wait for the blob to upload before you continue with this lab.

Task 4: Create a web app

- [ ] 1. On the Azure portal's navigation pane, select **Create a resource**.
- [ ] 2. On the **Create a resource** blade, in the **Search services and marketplace** text box, enter **Web App**, and then select Enter.
- [ ] 3. On the **Marketplace** search results blade, select the **Web App** result.
- [ ] 4. On the **Web App** blade, select **Create**.
- [ ] 5. On the **Create Web App** blade, on the **Basics** tab, perform the following actions, and then select the **Monitoring** tab:

Setting	Action
Subscription drop-down list	Retain the default value
Resource group section	Select **ManagedPlatform**
Name text box	Enter **imgapi***[yourname]*
Publish section	Select **Code**
Runtime stack drop-down list	Select **.NET 8 (LTS)**
Operating System section	Select **Windows**
Region drop-down list	Select the **East US** region
Windows Plan (East US) section	Select **Create new**, enter the value **ManagedPlan** in the **Name** text box, and then select **OK**
Pricing plan section	Select **Standard S1**

The following screenshot displays the configured settings on the **Create web app** blade.



- [ ] 6. On the **Monitoring** tab, in the **Enable Application Insights** section, select **No**, and then select **Review + create**.
- [ ] 7. On the **Review + create** tab, review the options that you selected during the previous steps.
- [ ] 8. Select **Create** to create the web app by using your specified configuration.

> ⚠ **Note**: Wait for the web app to be created before you continue with this lab.

- [ ] 9. On the **Overview** blade, select the **Go to resource** button to navigate to the blade of the newly created web app.

Task 5: Configure the web app

- [ ] 1. On the **App Service** blade, in the **Settings** section, select the **Configuration** link.
- [ ] 2. In the **Configuration** section, perform the following actions, select **Save**, and then select **Continue**.

Setting	**Action**
Application settings tab	Select **New application setting**
Add/Edit application setting pop-up dialog	In the **Name** text box, enter **StorageConnectionString**
Value text box	Paste the storage connection string that you previously copied to Notepad
Deployment slot setting check box	Retain the default value, and then select **OK** to close the pop-up dialog and return to the **Configuration** section

Setting	Action
Click **Save** in the top menu	This will save the configuration value you just entered

Wait for your application settings to save before you continue with the lab.

- [ ] 3. On the **App Service** blade in the **Settings** section, select the **Properties** link.
- [ ] 4. In the **Properties** section, copy the value of the **URL** hyperlink, and then paste it to Notepad. You'll use this value later in the lab.

> ⚠ **Note**: At this point, the web server at this URL will return a placeholder webpage. You haven't deployed any code to the Web App yet. You'll deploy code to the Web App later in this lab.

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

- [ ] 1. In the start menu, search for and select **Visual Studio Code**.
- [ ] 2. On the **File** menu, select **Open Folder**.
- [ ] 3. In the **File Explorer** window, browse to **Allfiles (D):\Allfiles\Labs\01\Starter\API**, and then select **Select Folder**.

> ⚠ **Note**: Ignore any prompts to add required assets to build and debug and to run the restore command to address unresolved dependencies.

- [ ] 4. On the **Explorer** pane of the **Visual Studio Code** window, expand the **Controllers** folder, and then select the **ImagesController.cs** file to open the file in the editor.
- [ ] 5. In the editor, in the **ImagesController** class on line 26, observe the **GetCloudBlobContainer** method and the code used to retrieve a container.
- [ ] 6. In the **ImagesController** class on line 36, observe the **Get** method and the code used to retrieve all blobs asynchronously from the **images** container.
- [ ] 7. In the **ImagesController** class on line 68, observe the **Post** method and the code used to persist an uploaded image to Storage.
- [ ] 8. On the taskbar, select the **Terminal** icon.
- [ ] 9. At the open command prompt, enter the following command, and then select Enter to sign in to the Azure Command-Line Interface (CLI):

```
az login
```

- [ ] 10. In the **Microsoft Edge** browser window, enter the email address and password for your Microsoft account, and then select **Sign in**.
- [ ] 11. Return to the currently open Terminal **Command Prompt** window. Wait for the sign-in process to finish.
- [ ] 12. At the command prompt, enter the following command, and then select Enter to list all the apps in your **ManagedPlatform** resource group:

```
az webapp list --resource-group ManagedPlatform-MPAYZBHZ5D
```

- [ ] 13. Enter the following command, and then select Enter to find the apps that have the **imgapi*** prefix:

```
az webapp list --resource-group ManagedPlatform-MPAYZBHZ5D --query "[?starts_with(name, 'imgapi')]"
```

- [ ] 14. Enter the following command, and then select Enter to render only the name of the single app that has the **imgapi*** prefix:

```
az webapp list --resource-group ManagedPlatform-MPAYZBHZ5D --query "[?starts_with(name, 'imgapi')].{Name:name}" --output tsv
```

- [ ] 15. Enter the following command, and then select Enter to change the current directory to the **Allfiles (D):\Allfiles\Labs\01\Starter\API** directory that contains the lab files:

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

- [ ] 16. Enter the following command, and then select Enter to deploy the **api.zip** file to the web app that you created previously in this lab:

```
az webapp deployment source config-zip --resource-group ManagedPlatform-MPAYZBHZ5D --src api.zip --name <name-of-your-api-app>
```

> ⚠ **Note**: Replace the *<name-of-your-api-app>* placeholder with the name of the web app that you created previously in this lab. You recently queried this app's name in the previous steps.

Wait for the deployment to complete before you continue with this lab.

- [ ] 17. On the Azure portal's **navigation** pane, select the **Resource groups** link.
- [ ] 18. On the **Resource groups** blade, select the **ManagedPlatform** resource group that you created previously in this lab.

- [ ] 19. On the **ManagedPlatform** blade, select the **imgapi***[yourname]* web app that you created previously in this lab.
- [ ] 20. From the **App Service** blade, select **Browse**.

> ⚠ **Note**: The **Browse** command will perform a GET request to the root of the website, which returns a JavaScript Object Notation (JSON) array. This array should contain the URL for your single uploaded image in your Storage account.

- [ ] 21. Return to your browser window that contains the Azure portal.
- [ ] 22. Close the currently running Visual Studio Code and 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 Apache Kudu zip file deployment utility.

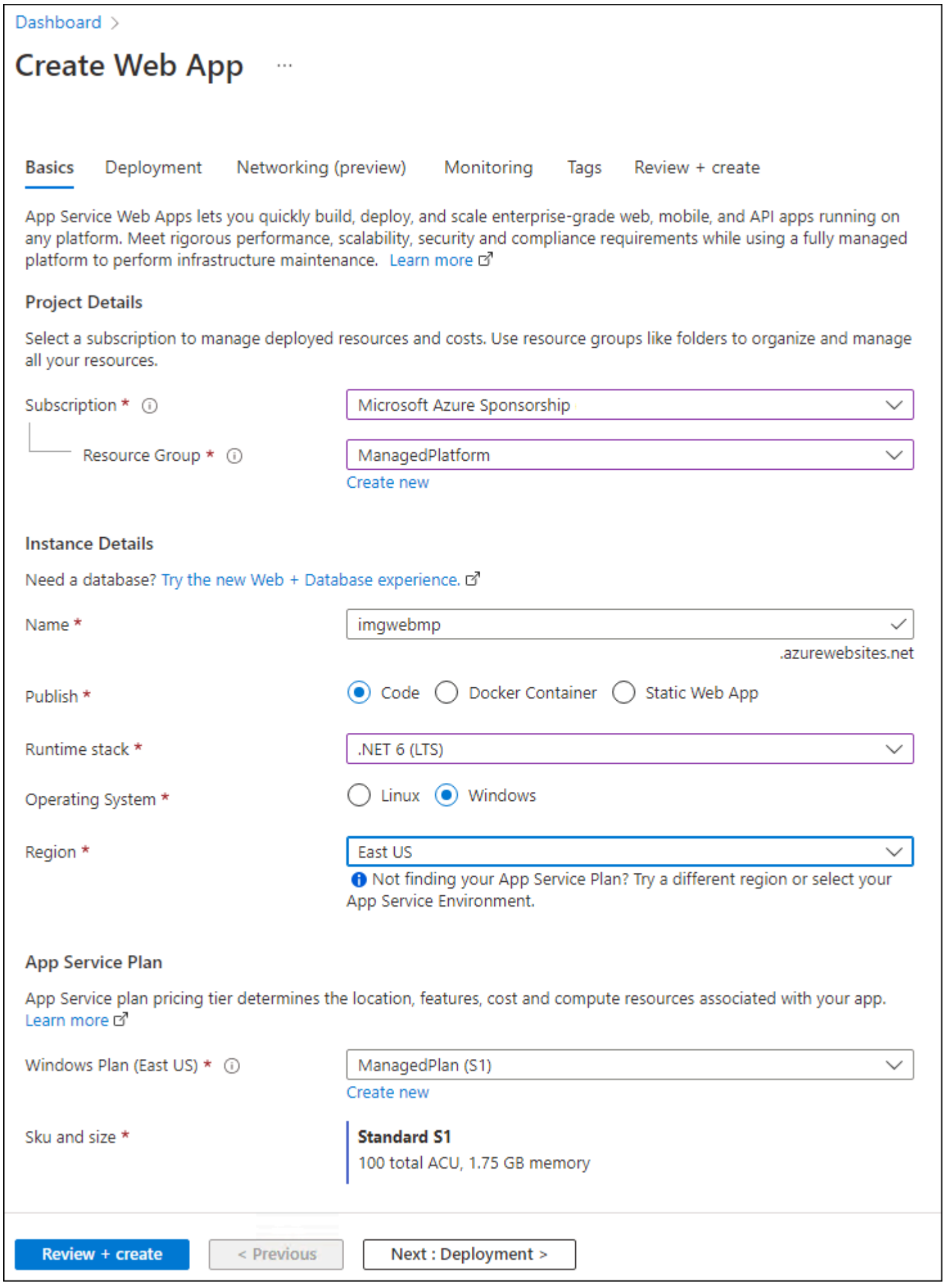
Exercise 2: Build a front-end web application by using Azure Web Apps

Task 1: Create a web app

- [ ] 1. On the Azure portal's **navigation** pane, select **Create a resource**.
- [ ] 2. On the **Create a resource** blade, in the **Search services and marketplace** text box, enter **Web App**, and then select Enter.
- [ ] 3. On the **Marketplace** search results blade, select **Web App**.
- [ ] 4. On the **Web App** blade, select **Create**.
- [ ] 5. On the **Create Web App** blade, on the **Basics** tab, perform the following actions, and then select the **Monitoring** tab:

Setting	Action
Subscription drop-down list	**CloudShare1A**
Resource group section	Select **ManagedPlatform-MPAYZBHZ5D**
Name text box	Enter **imgweb***[yourname]*
Publish section	Select **Code**
Runtime stack drop-down list	Select **.NET 8 (LTS)**
Operating System section	Select **Windows**
Region drop-down list	Select the **East US** region
Windows Plan (East US) section	Select **ManagedPlan (S1)**

The following screenshot displays the configured settings on the **Create web app** blade.



- [ ] 6. On the **Monitoring** tab, in the **Enable Application Insights** section, select **No**, and then select **Review + create**.
- [ ] 7. From the **Review + create** tab, review the options that you selected during the previous steps.
- [ ] 8. Select **Create** to create the web app by using your specified configuration.

> ⚠ **Note**: Wait for the creation task to complete before you continue with this lab.

- [ ] 9. On the **Overview** blade, select the **Go to resource** button to navigate to the blade of the newly created web app.

Task 2: Configure a web app

- [ ] 1. On the **App Service** blade, in the **Settings** section, select the **Configuration** link.
- [ ] 2. In the **Configuration** section, perform the following actions, select **Save**, and then select **Continue**:

Setting	**Action**
Application settings tab	Select **New application setting**
Add/Edit application setting pop-up dialog	In the **Name** text box, enter **ApiUrl**
Value text box	Enter the web app URL that you copied previously in this lab. **Note**: Make sure you include the protocol **https://**, in the URL that you copy into the **Value** text box for this application setting

Setting	Action
Deployment slot setting check box	Retain the default value, and then select **OK**
Click **Save** in the top menu	This will save the configuration value you just entered

> ⚠ **Note**: Wait for the application settings to save before you continue with this lab.

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

- [ ] 1. On the taskbar, select the **Visual Studio Code** icon.
- [ ] 2. On the **File** menu, select **Open Folder**.
- [ ] 3. In the **File Explorer** window, browse to **Allfiles (D):\Allfiles\Labs\01\Starter\Web**, and then select **Select Folder**.

> ⚠ **Note**: Ignore any prompts to add required assets to build and debug and to run the restore command to address unresolved dependencies.

- [ ] 4. On the **Explorer** pane of the **Visual Studio Code** window, expand the **Pages** folder, and then select the **Index.cshtml.cs** file to open the file in the editor.
- [ ] 5. In the editor, in the **IndexModel** class on line 30, observe the **OnGetAsync** method and the code used to retrieve the list of images from the API.
- [ ] 6. In the **IndexModel** class on line 41, observe the **OnPostAsync** method and the code used to stream an uploaded image to the backend API.
- [ ] 7. On the taskbar, select the **Terminal** icon.
- [ ] 8. At the open command prompt, enter the following command, and then select Enter to sign in to the Azure CLI:

```
az login
```

- [ ] 9. In the **Microsoft Edge** browser window, enter the email address and password for your Microsoft account, and then select **Sign in**.
- [ ] 10. Return to the currently open Terminal **Command Prompt** window. Wait for the sign-in process to finish.
- [ ] 11. Enter the following command, and then select Enter to list all the apps in your **ManagedPlatform** resource group:

```
az webapp list --resource-group ManagedPlatform-MPAYZBHZ5D
```

- [ ] 12. Enter the following command, and then select Enter to find the apps that have the **imgweb*** prefix:

```
az webapp list --resource-group ManagedPlatform-MPAYZBHZ5D --query "[?starts_with(name, 'imgweb')]"
```

- [ ] 13. Enter the following command, and then select Enter to render only the name of the single app that has the **imgweb*** prefix:

```
az webapp list --resource-group ManagedPlatform-MPAYZBHZ5D --query "[?starts_with(name, 'imgweb')].{Name:name}" --output tsv
```

- [ ] 14. Enter the following command, and then select Enter to change the current directory to the **Allfiles (D):\Allfiles\Labs\01\Starter\Web** directory that contains the lab files:

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

- [ ] 15. Enter the following command, and then select Enter to deploy the **web.zip** file to the web app that you created previously in this lab:

```
az webapp deployment source config-zip --resource-group ManagedPlatform-MPAYZBHZ5D --src web.zip --name <name-of-your-web-app>
```

> ⚠ **Note**: Replace the *<name-of-your-web-app>* placeholder with the name of the web app that you created previously in this lab. You recently queried this app's name in the previous steps.

Wait for the deployment to complete before you continue with this lab.

- [ ] 16. On the Azure portal's **navigation** pane, select **Resource groups**.
- [ ] 17. On the **Resource groups** blade, select the **ManagedPlatform** resource group that you created previously in this lab.
- [ ] 18. On the **ManagedPlatform** blade, select the **imgweb***[yourname]* web app that you created previously in this lab.
- [ ] 19. On the **App Service** blade, select **Browse**.
- [ ] 20. Observe the list of images in the gallery. The gallery should list a single image that was uploaded to Storage previously in the lab.
- [ ] 21. In the **Contoso Photo Gallery** webpage, in the **Upload a new image** section, perform the following actions:

a. Select **Browse**.

b. In the **File Explorer** window, browse to **Allfiles (D):\Allfiles\Labs\01\Starter\Images**, select the **Banhmi.jpg** file, and then select **Open**.

c. Select **Upload**.

- [ ] 22. 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.

- [ ] 23. Return to the browser window that contains the Azure portal.

- [ ] 24. Close the currently running Visual Studio Code and 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.