## Building a web application on Azure platform as a service offering

# 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:

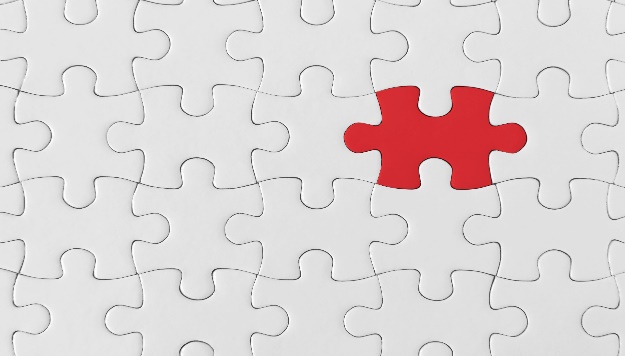
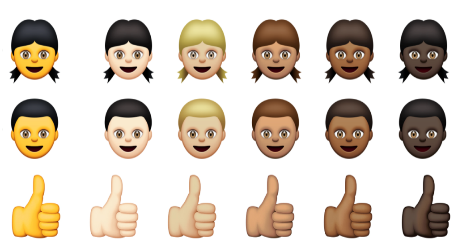
* Create various apps by using App Service.
* Configure application settings for an app.
* Deploy apps by using Kudu, the Azure Command-Line Interface (CLI), and zip file deployment.



Container in the Storage account should store the uploaded image

Website should display all the pictures uploaded by users and in the container

Users would be uploading pic on the site.



API to connect frontend Webapp and storage container

## Lab setup

* Estimated time: **45 minutes**

### Tools to be used in this Lab

* Windows 10
* Azure subscription
* Microsoft Edge
* File Explorer
* Windows PowerShell
* Visual Studio Code
* Fork the following github repository as we are to use this for app deployment.

[https://github.com/TPCSAcademy/Webapp\_Demo.git (github.com)](https://github.com/ShrutiSinhaa/AZ-204-DevelopingSolutionsforMicrosoftAzure)



[This Photo](http://asuyatuyolar.org/2013/05/ack-kaynak-makale.html) by Unknown Author is licensed under [CC BY](https://creativecommons.org/licenses/by/3.0/)

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

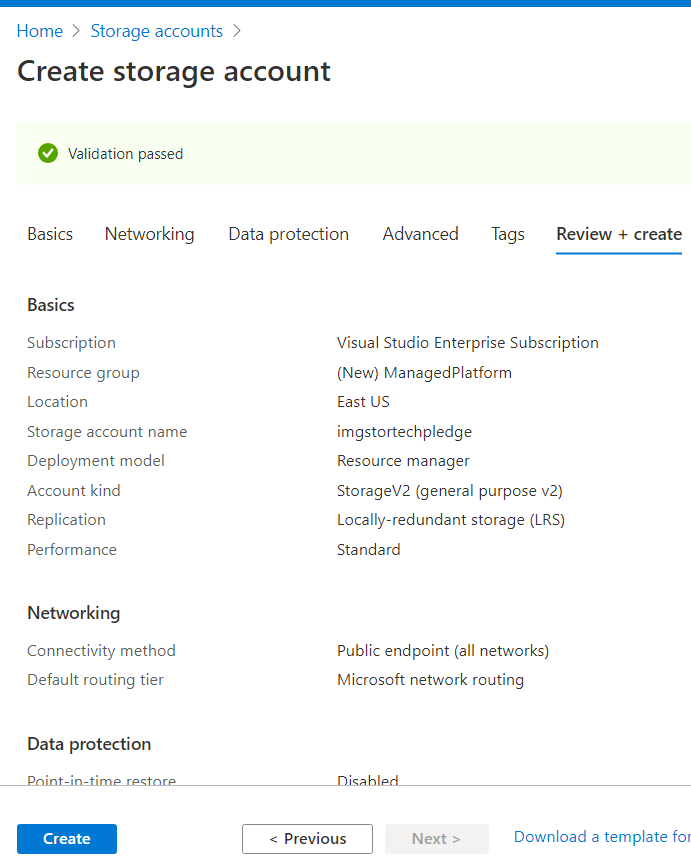
#### Task 1: Open the Azure portal

Sign in to the Azure portal ([https://portal.azure.com](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)**
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.



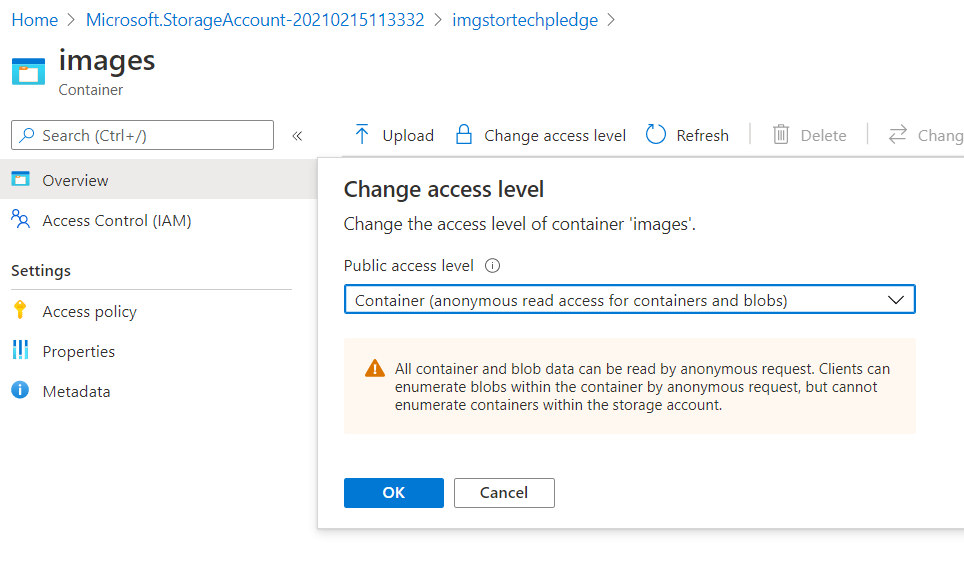
1. Access the **Access Keys** blade of your newly created storage account instance.
2. Record the value of the **Connection string** text box. You'll use this value later in this lab.

**Sample Connection String:**

DefaultEndpointsProtocol=https;AccountName=imgstortechpledge;AccountKey=VNzkTN2KBJleZFk8jkl9OL/NY1hvysoH41Yu2Woy6wPqldUi1SwHFrAvRHQwtr1gd3YK/p/mDdj7imYw/9zLzQ==;EndpointSuffix=core.windows.net

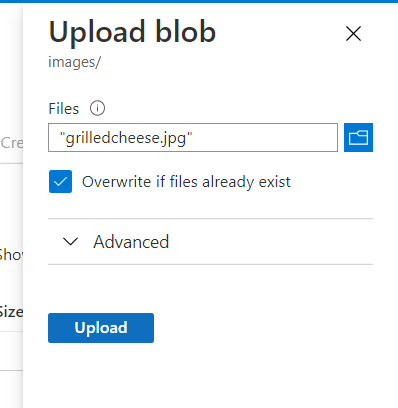
#### Task 3: Upload a sample blob

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)**



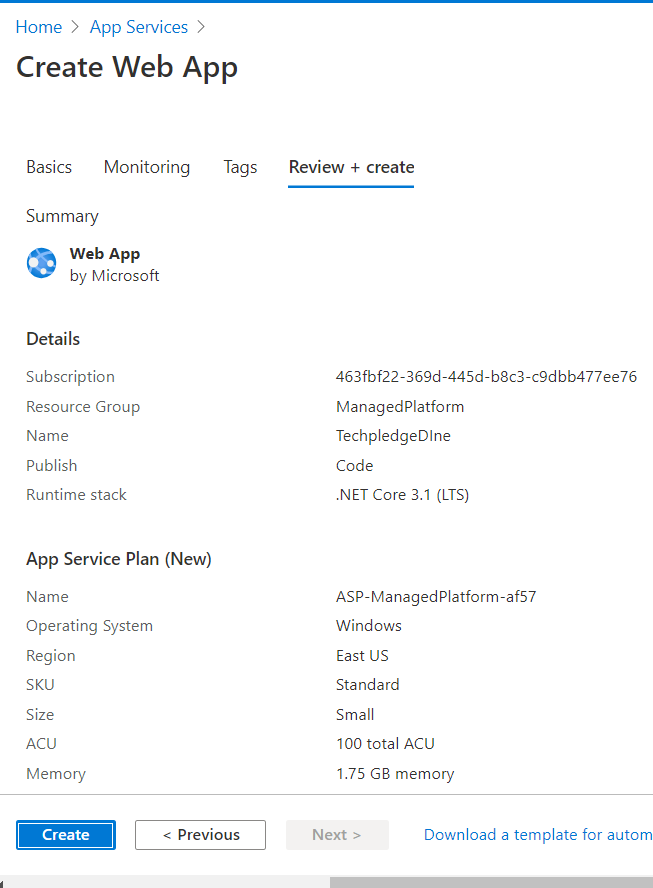
1. Go to the new **images** container, and then use the **Upload** button to upload the **grilledcheese.jpg** file in the **Allfiles (F):\Allfiles\Labs\01\Starter\Images** folder on your lab machine.

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



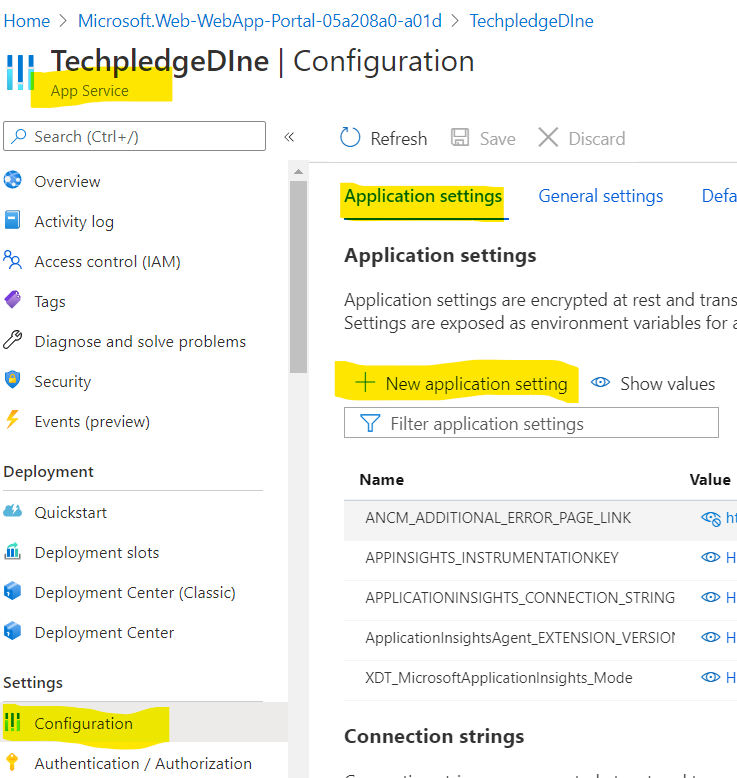
#### Task 4: Create a web app

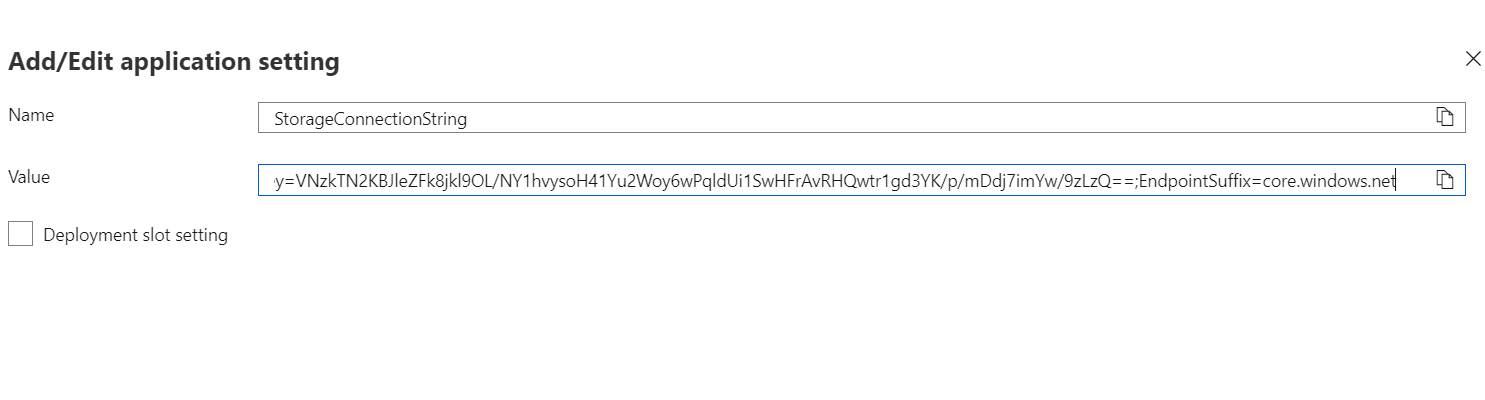
1. Create a new web app with the following details:
   * Existing resource group: **ManagedPlatform**
   * Web App name: imgapi[yourname]\*\*
   * Publish: **Code**
   * Runtime stack: **.NET Core 3.1 (LTS)**
   * Operating System: **Windows**
   * Region: **East US**
   * New App Service plan: **ManagedPlan**
   * SKU and size: **Standard (S1)**
   * Application Insights: **Disabled**

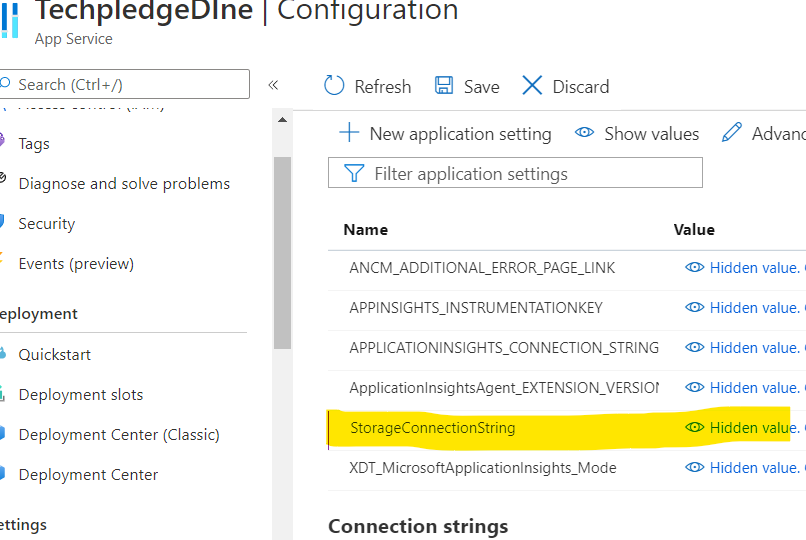


#### Task 5: Configure the web app

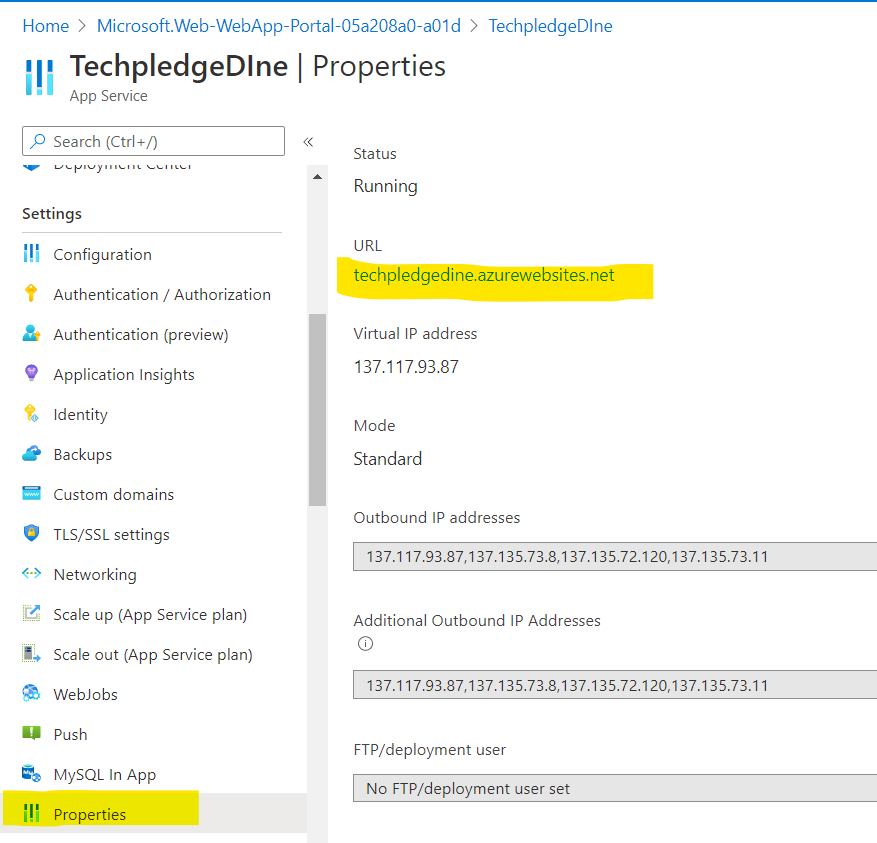
1. Access the imgapi[yourname]\*\* web app that you created earlier in this lab.
2. In the **Settings** section, find the **Configuration** section, and then create a new application setting by using the following details:
   * Name: **StorageConnectionString**
   * Value: ***Storage Connection String copied earlier in this lab***
   * Deployment slot setting: **Not selected**







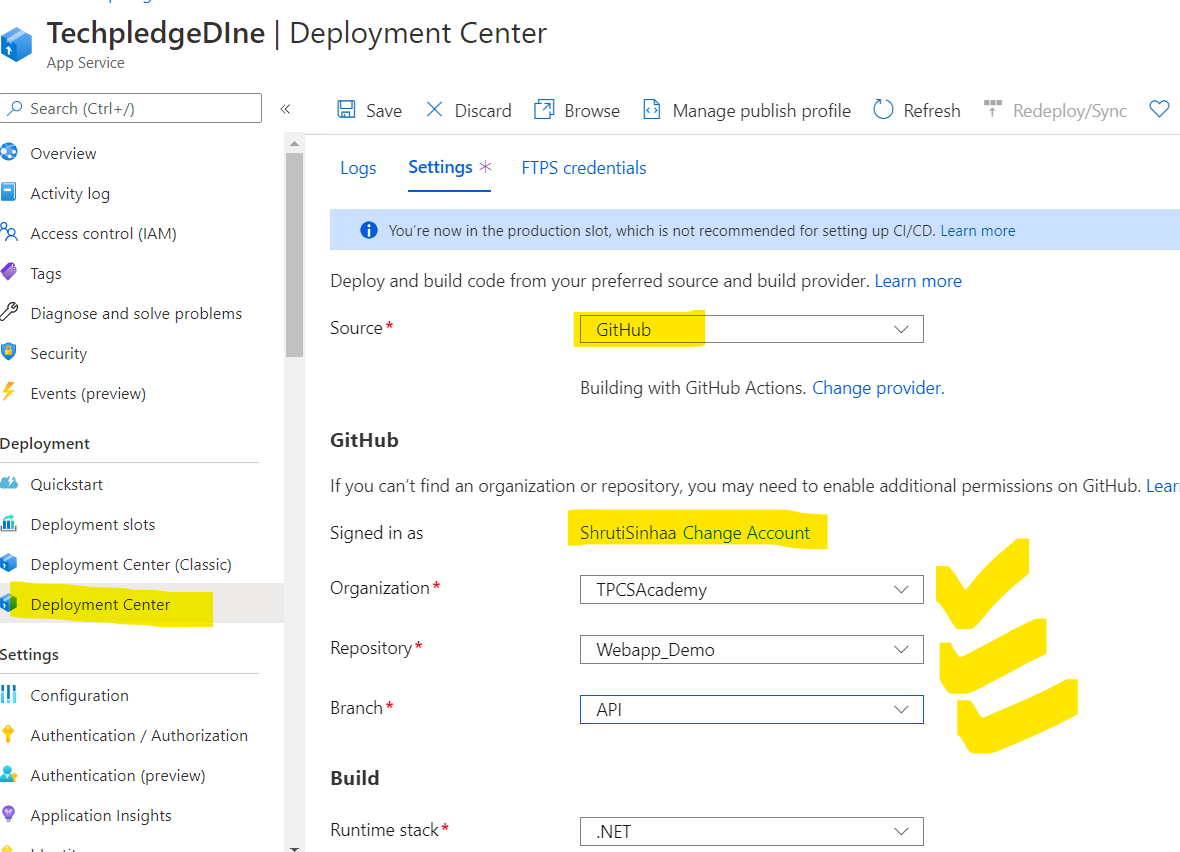
1. Save your changes to the application settings.
2. In the **Settings** section, find the **Properties** section.
3. In the **Properties** section, copy the value of the **URL** text box. You'll use this value later in the lab.



[techpledgedine.azurewebsites.net](http://techpledgedine.azurewebsites.net/)

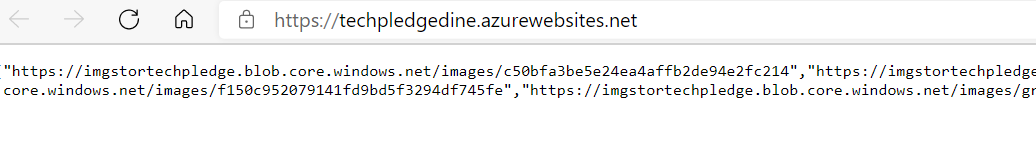
**Note**: At this point, the web server at this URL will return a 404 error. You have not deployed any code to the Web App yet. You will deploy code to the Web App later in this lab.

#### Task 6: Deploy an ASP.NET web application from GitHub

* Select Deployment center
* Select Github
* Login to github
* Select the repository and API branch
* 

Task 7: Verify the web app:

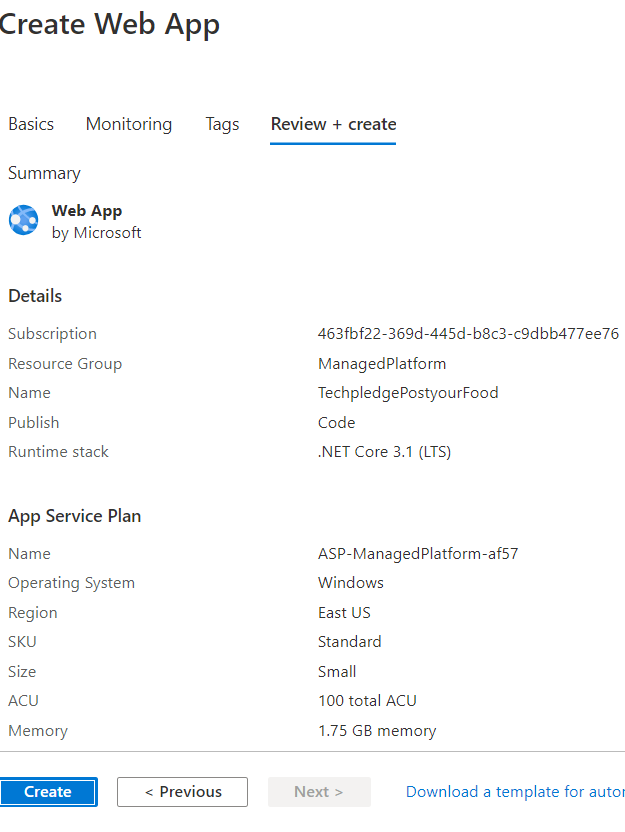
* Go to Overview
* Select browse:
* You should be able to see the path for all the blobs uploaded in the container:



### 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**



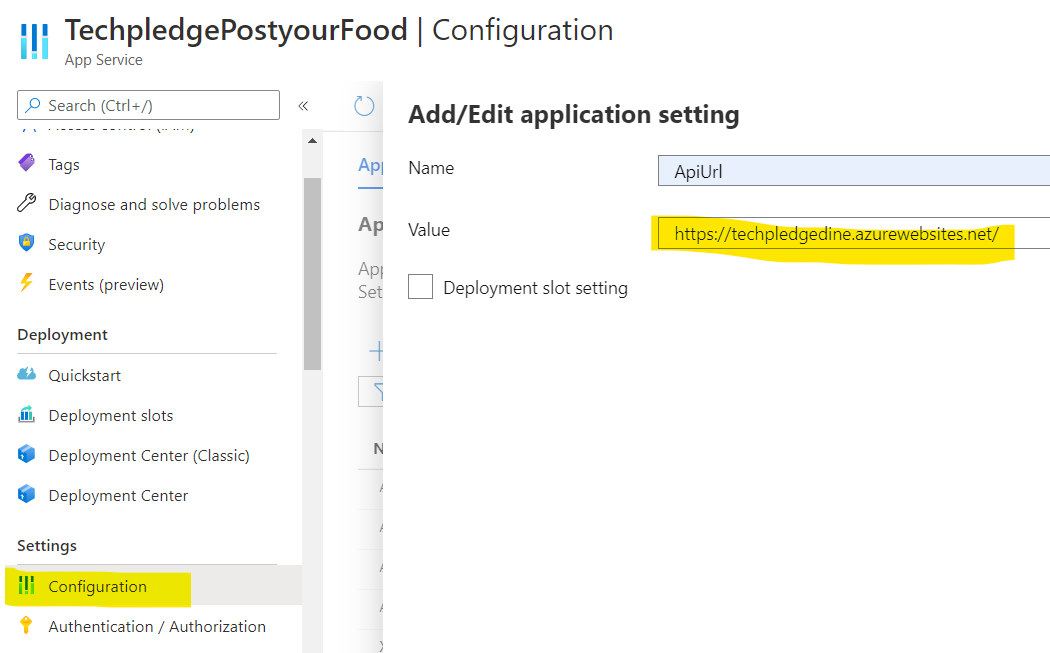
1. Wait for Azure to finish creating the web app before you move forward with the lab. You'll receive a notification when the app is created.

#### Task 2: Configure a web app

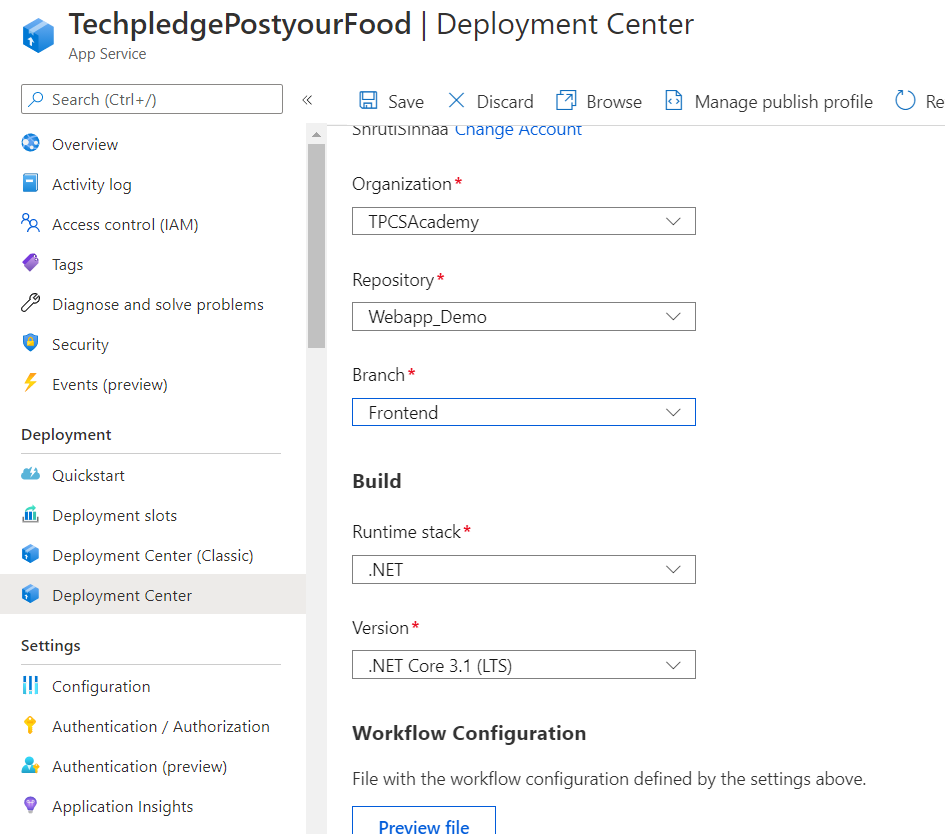
1. Access the imgweb[yourname]\*\* web app that you created in the previous task.
2. In the **Settings** section, find the **Configuration** settings.
3. Create a new application setting by using the following details:
   * Name: **ApiUrl**
   * Value: ***Web app URL copied earlier in this lab***
  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

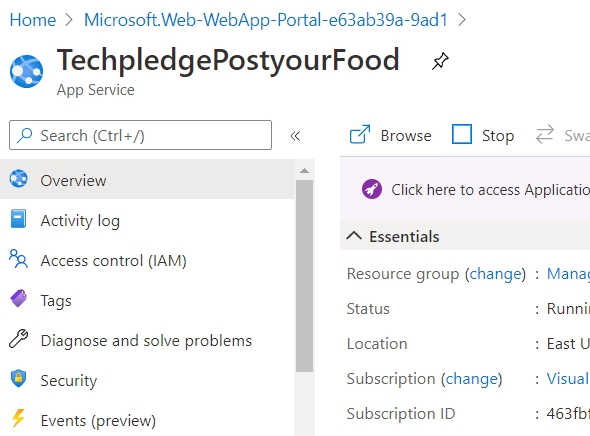
**Note**: Make sure you include the protocol, such as **https://**, in the URL that you copy into the **Value** text box for this application setting.

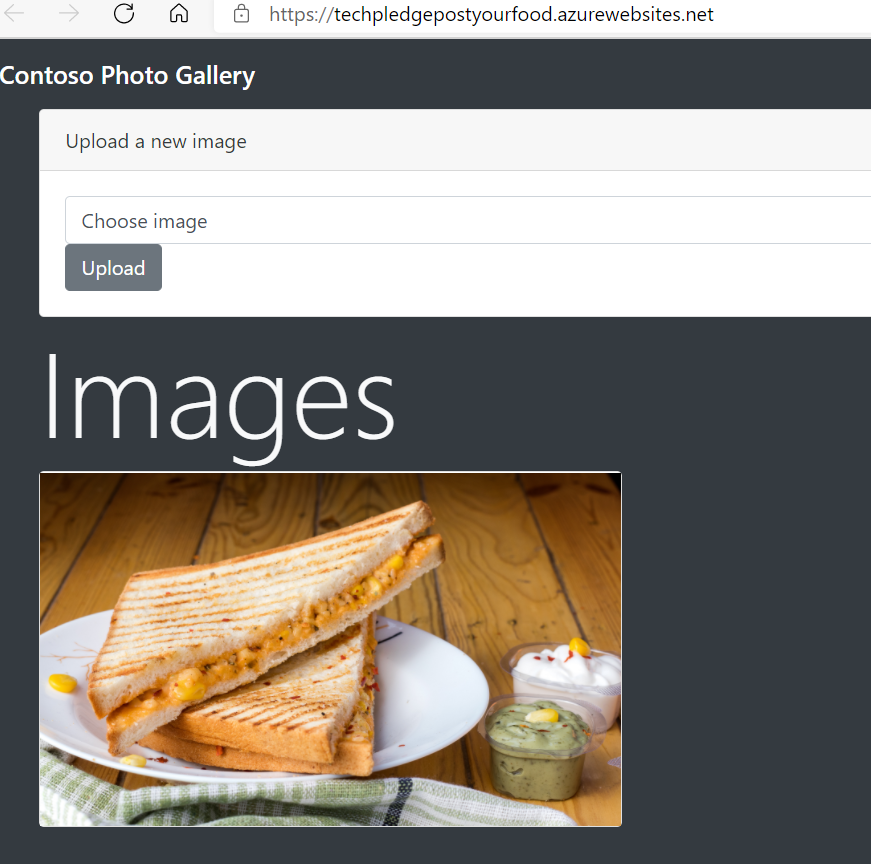
1. Save your changes to the application settings.



#### Task 3: Deploy an ASP.NET web application From GitHUB

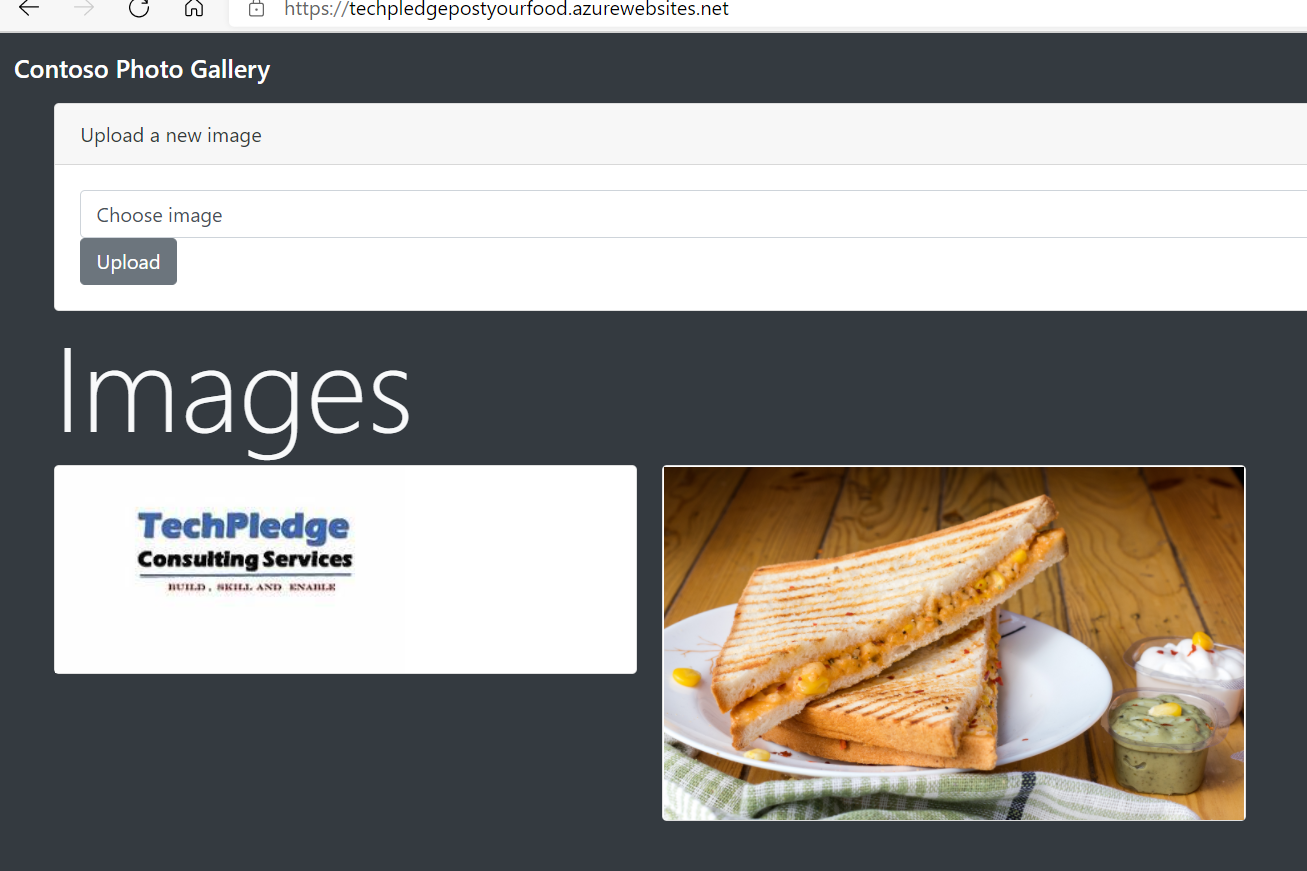
1. Select Deployment center
2. Select Github
3. Login to github
4. Select the repository and API branch
5. 
6. Access the web app that you created earlier in this lab. Open the web app in your browser.





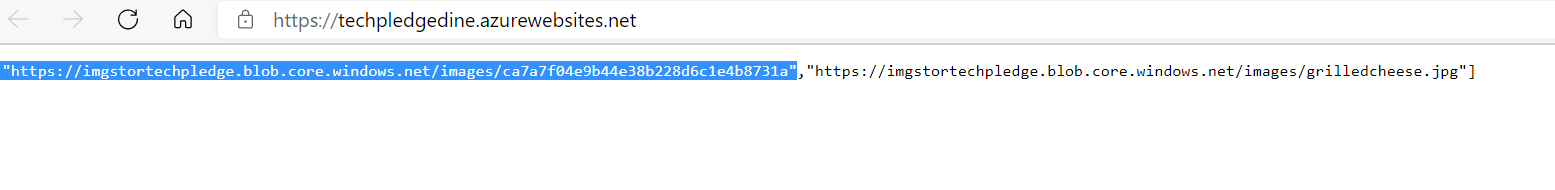
1. From the **Contoso Photo Gallery** webpage, find the **Upload a new image** section, and then upload the **bahnmi.jpg** file in the **Allfiles (F):\Allfiles\Labs\01\Starter\Images** folder on your lab machine.

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

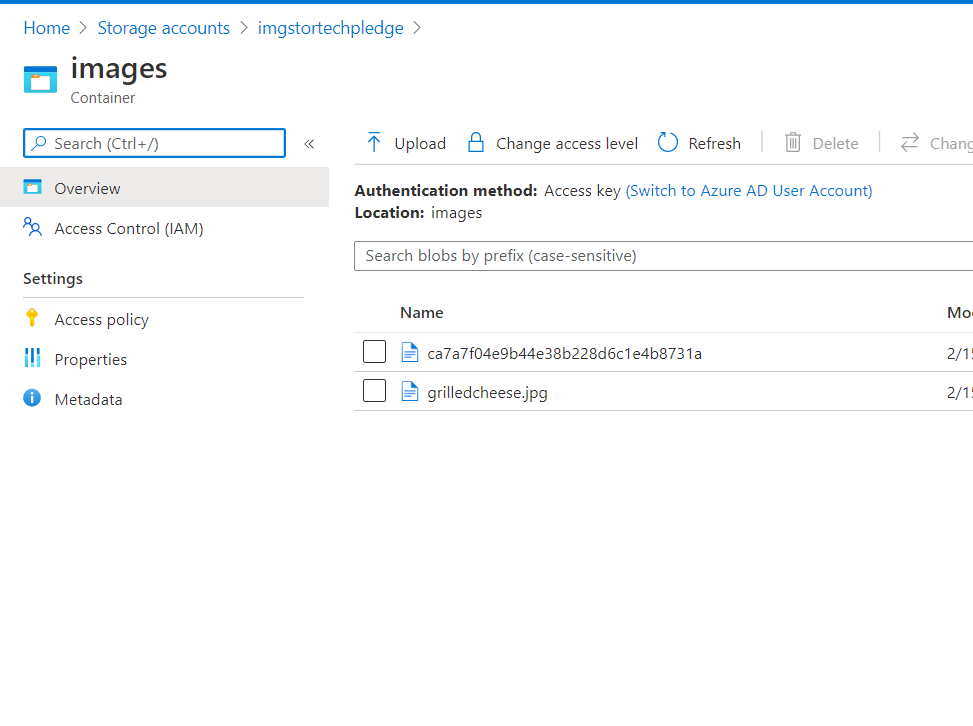


1. Observe that the list of gallery images has updated with your new image.
   1. Observer the API webapp

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



* 1. Observer the new blob in the storage container:



1. 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

1. 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.

Top of Form