

# ラボ開始前の準備

- 演習環境（SEA-DEVのVM）を使用する場合
  - Visual Studio CodeとWindows Terminalはすでに演習環境にインストールされていますのでインストール作業は不要です。
  - **演習環境の内部で**、ブラウザを起動して、ラボのファイルをダウンロードします。

- **ご自身のPCを使用する場合**
  - Visual Studio CodeとWindows Terminalをインストールします。（または、お好きなエディタとWindows PowerShellを使用します）
  - **ご自身のPCで**、ブラウザを起動して、ラボのファイルをダウンロードします。

下記のURLをブラウザで開きます。

<https://github.com/MicrosoftLearning/AZ-204-DevelopingSolutionsforMicrosoftAzure>

The screenshot shows the GitHub repository page for 'AZ-204-DevelopingSolutionsforMicrosoftAzure'. The repository has 78 stars and 487 forks. The 'Code' dropdown menu is highlighted with a red box. Below it, the 'Download ZIP' button is also highlighted with a red box.

Repository details:

- Owner: MicrosoftLearning
- Name: AZ-204-DevelopingSolutionsforMicrosoftAzure
- Stars: 78
- Forks: 487

Code dropdown menu:

- Clone (selected)
- HTTPS
- SSH
- GitHub CLI

Download ZIP

Repository contents (partial list):

- .github
- Allfiles
- Instructions
- LICENSE
- \_build.yml
- \_config.yml
- index.md
- lab.md
- readme.md

Readme:

AZ-204: Developing solutions for Microsoft Azure

Contributors:

AZ-204: Developing solutions for Microsoft Azure

「Code」をクリックします

「Download ZIP」をクリックします。  
ZIPファイルがダウンロードされます。  
通常、「ダウンロード」フォルダに  
格納されます。

ダウンロードしたZIPファイルを  
展開し、展開された  
フォルダの場所を確認しておきます。

**演習 1: Azure アプリ サービスの Azure ストレージと Web アプリ機能を使用してバックエンド API を構築する**

## タスク 1: Azure portal を開く

1. Azure portal (<https://portal.azure.com>) にログインします。

**注:** Azure portalに初めてログインする場合、ポータルのツアーを提供するダイアログボックスが表示されます。ツアーをスキップするには、「開始」を選択します。

ホーム - Microsoft Azure

https://po...

Microsoft Azure リソース、サービス、ドキュメントの検索 (G+/)

Azure サービス

+ リソースの作成 ストレージ アカウント App Service リソース グループ

Advisor Azure Active Directory サブスクリプション Virtual Machines

SQL データベース → その他のサービス

最近のリソース

名前 種類 最終表示日

https://portal.azure.com/#create/hub

## タスク 2: ストレージ アカウントの作成

1. 次の詳細で新規ストレージ アカウントを作成します:

- 新しいリソース グループ: ManagedPlatform
- 名前: *imgstor[yourname]\*\**
- 場所: (米国) 米国東部
- パフォーマンス: Standard
- アカウント サブタイプ: StorageV2 (general purpose v2)
- レプリケーション: ローカル冗長ストレージ (LRS)

2. Azure が Storage アカウントの作成を完了するのを待ってから、ラボを進みます。アカウントの作成時に通知が届きます。

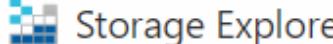
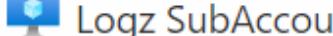
3. 新しく作成したストレージ アカウント インスタンスの**アクセス キーブレード**にアクセスします。

4. 「接続文字列」 テキストボックスの値を記録します。この値は、この課題の後半で使用します。

5. 「構成」 セクションを開き、「BLOBパブリックアクセス」を**有効**にします。

 リソースの作成 ホーム ダッシュボード すべてのサービス お気に入り すべてのリソース リソース グループ App Service 関数アプリ

## サービス

 ストレージ アカウント Storage Explorer Purview accounts SendGrid Accounts Logz SubAccount Logz Main Account Data Lake Storage Gen1

## リソース

該当結果が見つかりませんでした。

## Marketplace

 Storage account

## ドキュメント

[Azure キューの概要 - Azure Storage | Microsoft Docs](#)[Azure 向けの Microsoft マルウェア対策 | Microsoft Docs](#)[Azure Storage ファイアウォールおよび仮想ネットワークの構成 | Microsoft Docs](#)[マネージド ID をサポートする Azure サービス - Azure AD | Microsoft Docs](#)

## リソース グループ

該当結果が見つかりませんでした。

&gt;&gt;

ホーム &gt;



## ストレージ アカウント



AZ204-2020-1214



+ 追加



ビューの管理



更新



CSV にエクスポート



名前でフィルター...



サブスクリプション == すべて

リソース



0 件中 0 ~ 0 件のレコードを表示しています。



名前 ↑↓



種類 ↑↓

サブタイプ ↑↓

Azure Storage は、高可用性、セキュリティ、耐久性、スケーラビリティ、冗長性を備えたクラウドストレージを提供する Microsoft が管理するサービスです。Azure Storage には、Azure BLOB (オブジェクト)、Azure Data Lake Storage Gen2、Azure Files、Azure Queues、Azure Tables が含まれます。ストレージ アカウントのコストは、使用量と、下で選ぶオプションに応じて決まります。 [Azure ストレージ アカウントの詳細](#)

## プロジェクトの詳細

デプロイされているリソースとコストを管理するサブスクリプションを選択します。フォルダーのようなリソース グループを使用して、すべてのリソースを整理し、管理します。

サブスクリプション \*

Azure Pass - AZ204-2020-1214



リソース グループ \*

新規作成



## インスタンスの詳細

既定の展開モデルは Resource Manager であり、選択できます。 [クラシック展開モデルを選択します](#)

リソース グループは、Azure のソリューションに関連するリソースを保持するコンテナーです。

モデルを使った展開も

ストレージ アカウント名 \* ⓘ

名前 \*

ManagedPlatform



OK

キャンセル

場所 \*

確認および作成

< 前へ

次: ネットワーク >

## インスタンスの詳細

既定の展開モデルは Resource Manager であり、これは最新の Azure 機能をサポートしています。代わりに、従来の展開モデルを使った展開も選択できます。 [クラシック展開モデルを選択します](#)

ストレージ アカウント名 \* ⓘ

imgstoryamada



場所 \*

(US) 米国東部



パフォーマンス ⓘ

Standard  Premium

アカウントの種類 ⓘ

StorageV2 (汎用 v2)



レプリケーション ⓘ

ローカル冗長ストレージ (LRS)



**確認および作成**

< 前へ

次: ネットワーク >



## 通知

X

アクティビティ ログのその他のイベント →

すべて無視 ▼



展開が成功しました

X

リソース グループ 'ManagedPlatform' への 'Microsoft.StorageAccount-20201214114116' のデプロイが成功しました。

リソースに移動

↗ ダッシュボードにピン留めする

数秒前



Azure Advisor からの無償の推奨事項をご覧ください。

X

Azure Advisor は、Azure の使用を分析して、節約、パフォーマンスの向上、セキュリティ保護の強化、Azure で既に実行されているソリューションの信頼性の向上を実現するためのレコメンデーションを提供する無料のサービスです。 [詳細情報](#)

# imgstoryamada | アクセス キー

ストレージ アカウント

検索 (Ctrl+ /)



この Azure ストレージ アカウントに対して要求を行う際は、アプリケーションの認証にアクセス キーを使用します。アクセス キーは、(たとえば Azure Key Vault を使用して) 安全に保存し、共有しないでください。アクセス キーは、定期的に再生成することをお勧めします。アクセス キーは 2 つ提供されるため、一方のキーを再生成しながら、もう一方のキーを使用して接続を維持することができます。

アクセス キーを再生成する場合、その新しいキーを使用するには、このストレージ アカウントにアクセスする Azure リソースとアプリケーションをすべて更新する必要があります。この操作によって、ご使用の仮想マシンからディスクへのアクセスが中断されることはありません。[ストレージ アクセス キーの再生成に関する詳細情報](#)

ストレージ アカウント名

imgstoryamada



キーの表示



key1



キー

.....

接続文字列

.....



key2



キー

.....

接続文字列

.....

コピーした接続文字列を、  
メモ帳などに貼り付けしておきます

概要

アクティビティ ログ

タグ

問題の診断と解決

アクセス制御 (IAM)

データ転送

イベント

Storage Explorer (プレビュー)

設定

アクセス キー

geo レプリケーション

CORS

構成

暗号化

Shared Access Signature

トライアル



# imgstoryamada | 構成

ストレージ アカウント

検索 (Ctrl+ /)

保存

破棄

最新の情報に更新

概要

アクティビティ ログ

タグ

問題の診断と解決

アクセス制御 (IAM)

データ転送

イベント

Storage Explorer (プレビュー)

設定

アクセス キー

geo レプリケーション

CORS

構成

暗号化

Shared Access Signature

ストレージ アカウントのコストは、使用量と、下で選ぶオプションに応じて決まります。ストレージの価格に関する詳

アカウントの種類

StorageV2 (汎用 v2)

パフォーマンス ⓘ

 Standard  Premium

● ストレージ アカウントが作成された後にこの設定を変更することはできません。

安全な転送が必須 ⓘ

 無効  有効

BLOB パブリック アクセスを許可する ⓘ

 無効  有効

TLS の最小バージョン ⓘ

バージョン 1.2

BLOB のアクセス レベル (既定) ⓘ

 クール  ホット

レプリケーション ⓘ

ローカル冗長ストレージ (LRS)

「無効」になっている場合は、  
「有効」を選択して「保存」します

## タスク 3: サンプル BLOB のアップロード

1. この実習ラボで作成済みの `imgstor[yourname]**` ストレージ アカウントにアクセスします。
2. 「Blob サービス」セクションで、「コンテナー」を選択します。
3. 次の設定で新しいコンテナーを作成してください:
  - 名前: `images`
  - パブリック アクセス レベル: `BLOB (BLOB の場合のみ匿名読み取りアクセス)`
4. 新しい `images` コンテナーを開き、`アップロード`をクリックして、ラボ マシン上の `Allfiles\Labs\01\Starter\Images` フォルダにある `grilledcheese.jpg` ファイルをアップロードします。

| 注: ファイルが既に存在する場合は、上書きを有効にすることをお勧めします。

# imgstoryamada | コンテナー

ストレージ アカウント

 検索 (Ctrl+ /)[+ コンテナー](#) [アクセス レベルを変更します](#) [コンテナーを復元する](#)

プレフィックスによるコンテナーの検索

名前

最終変更日

まだコンテナーがありません。最初に、[+ コンテナー] をクリックしてください。

Blob service

コンテナー

カスタム ドメイン

データ保護

オブジェクト レプリケーション

Azure CDN

Azure Search の追加

ライフサイクル管理

File サービス

ファイル共有

Table service

テーブル

Queue サービス

キュー

監視

# 新しいコンテナー

名前 \*

images



パブリック アクセス レベル

BLOB (BLOB 専用の匿名読み取りアクセス)

⚠️ コンテナー内の BLOB は匿名要求によって読み取ることができます。コンテナーのデータは使用できません。匿名クライアントはコンテナー内の BLOB を列挙できません。

▼ 詳細設定

作成

破棄

## imgstoryamada | コンテナー

ストレージ アカウント

検索 (Ctrl+ /) << + コンテナー □ アクセス レベルを変更します ↻ コンテナーを復元する ⌂ 更新

プロパティ

Blob service

コンテナー

カスタム ドメイン

データ保護

オブジェクト レプリケーション

プレフィックスによるコンテナーの検索

名前	最終変更日時	パ
<input type="checkbox"/> images	2020/12/14 11:49:23	BL

## images

コンテナー

検索 (Ctrl+ /)



アップロード

アクセス レベルを変更します

更新

削除

層の変更

概要

アクセス制御 (IAM)

## 設定

アクセス ポリシー

プロパティ

メタデータ

認証方法: アクセス キー (Azure AD のユーザー アカウントに切り替える)

場所: images

プレフィックスによる BLOB の検索 (大文字と小文字を区別)

名前

変更日時

結果がありません

ア

アップロード

## BLOB のアップロード

images/

Files

"grilledcheese.jpg"



ファイルが既に存在する場合は上書きする

詳細設定

アップロード



← → ⌘ ⌘ ⌘ Allfiles > Labs > 01 > Starter > Images



Imagesの検索

整理 ▾ 新しいフォルダー



> ★ クイック アクセス

▼ デスクトップ

> OneDrive - Microsoft

> Hiromichi Yamada

> PC

> ライブラリ

> ネットワーク



babnmj.jpg



blt.jpg



burger.jpg



sub.jpg



veggie.jpg

ファイル名(N): grilledcheese.jpg

すべてのファイル (\*.\*)

開く(O)

キャンセル

## images

コンテナー

検索 (Ctrl+ /)



アップロード アクセス レベルを変更します 更新 削除 層の変更

認証方法: アクセス キー (Azure AD のユーザー アカウントに切り替える)

場所: images

プレフィックスによる BLOB の検索 (大文字と小文字を区別)

名前

変更日時

結果がありません

ア

## 設定

アクセス ポリシー

プロパティ

メタデータ

## BLOB のアップロード

images/

Files ⓘ

"grilledcheese.jpg"



ファイルが既に存在する場合は上書きする

詳細設定

アップロード



## タスク 4: Web アプリの構成

1. 次の詳細で新規 Web アプリを作成します:

- 既存のリソース グループ: **ManagedPlatform**
- Web アプリ名: *imgapi[yourname]\*\**
- 公開: コード
- ランタイム スタック: **.NET Core 3.1(LTS)**
- オペレーティング システム: **Windows**
- リージョン: **米国東部**
- 新規 App Services プラン: **ManagedPlan**
- SKU とサイズ: **Standard (S1)**
- Application Insights: **無効**

2. Azure が Web アプリの作成を完了するのを待ってから、ラボに進みます。アプリの作成時に通知が届きます。

»

ホーム &gt; Micro

  
image  
コンテナー

検索 (Ctrl+)



概要



アクセス制御



設定



アクセス ポリ



プロパティ



メタデータ



サービス

すべて

App Service Environment

App Service ドメイン

App Service プラン

App Service 証明書

App Service

ロジック アプリ (プレビュー)

関数 アプリ

静的 Web アプリ (プレビュー)

Application Services

Lab Services

リソース

»

ホーム &gt;



# App Service

AZ204-2020-1214

+ 追加



ビューの管理



更新



CSV にエクスポート



クエリを開く



タグ

名前でフィルター...

サブスクリプション == すべて

リソース グループ == すべて

0 件中 0 ~ 0 件のレコードを表示しています。

名前 ↑↓

状態 ↑↓

場所 ↑↓

価格レベル ↑↓



# Web アプリの作成

基本 監視 タグ 確認および作成

App Service Web Apps を使用すると、任意のプラットフォームで実行するエンタープライズ レベルの Web アプリ、モバイル アプリ、API アプリを素早くビルド、デプロイ、スケーリングできます。フル マネージド プラットフォームを使用してインフラストラクチャ メンテナンスを実行するには、パフォーマンス、スケーラビリティ、セキュリティおよびコンプライアンスの要件を確実に満たしてください。 [詳細情報](#)

## プロジェクトの詳細

デプロイされているリソースとコストを管理するサブスクリプションを選択します。フォルダーのようなリソース グループを使用して、すべてのリソースを整理し、管理します。

サブスクリプション \* ⓘ

Azure Pass - AZ204-2020-1214

リソース グループ \* ⓘ

ManagedPlatform

[新規作成](#)

## インスタンスの詳細

名前 \*

imgapiyamada

.azurewebsites.net

公開 \*

コード  Docker コンテナー

ランタイム スタック \*

.NET Core 3.1 (LTS)

オペレーティング システム \*

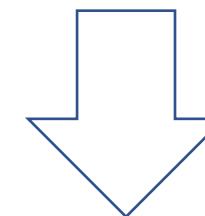
Linux  Windows

地域 \*

West US

App Service プランが見つからない場合、別のリージョンをお試しください。

画面をスクロールして  
下部を表示します



[確認および作成](#)

< 前へ

次: 監視 >

## App Service プラン

App Service プランの価格レベルによって、アプリに関連する場所、機能、コスト、コンピューティング リソースカ

Windows プラン (East US) \* ⓘ

(新規) ASP-ManagedPlatform-81cd

新規作成

SKU とサイズ \*

### 新しい App Service プラン

名前 \*

ManagedPlatform



OK

キャンセル

確認および作成

< 前へ

## App Service プラン

App Service プランの価格レベルによって、アプリに関連する場所、機能、コスト、コンピューティング リソースが決定されます。[詳細情報](#)

Windows プラン (East US) \* ⓘ

(新規) ManagedPlatform



新規作成

SKU とサイズ \*

Standard S1

ACU 合計 100, 1.75 GB メモリ

[サイズを変更します](#)

Standard S1と表示されていることを確認します。

Standard S1以外が表示されている場合は、「サイズを変更します」をクリックします。

[確認および作成](#)

< 前へ

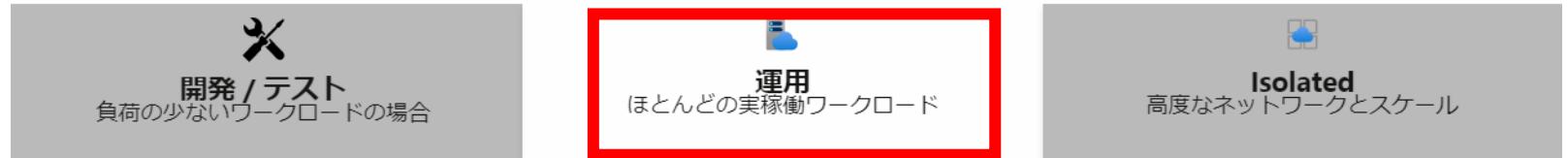
次: 監視 >

# スペックの選択

X

※「サイズを変更します」をクリックした場合のみ

「Standard S1」を選択し、  
「適用」をクリックします。



## 推奨される価格レベル

 <b>S1</b> ACU 合計 100 1.75 GB メモリ A シリーズ コンピューティングと同等 73.00 USD/月 (推定)	 <b>P1V2</b> ACU 合計 210 3.5 GB メモリ Dv2 シリーズ コンピューティングと同等 146.00 USD/月 (推定)	 <b>P2V2</b> ACU 合計 420 7 GB メモリ Dv2 シリーズ コンピューティングと同等 292.00 USD/月 (推定)
 <b>P3V2</b> ACU 合計 840 14 GB メモリ Dv2 シリーズ コンピューティングと同等 584.00 USD/月 (推定)	 <b>P1V3</b> 195 最小 ACU/vCPU 8 GB メモリ 2 個の仮想コア 244.55 USD/月 (推定)	 <b>P2V3</b> 195 最小 ACU/vCPU 16 GB メモリ 4 個の仮想コア 489.10 USD/月 (推定)
 <b>P3V3</b> 195 最小 ACU/vCPU 32 GB メモリ 8 個の仮想コア 978.20 USD/月 (推定)	<p>へ 推奨オプションのみを参照する</p>	

## 追加の価格レベル

 <b>S2</b> ACU 合計 200 3.5 GB メモリ A シリーズ コンピューティングと同等 146.00 USD/月 (推定)	 <b>S3</b> ACU 合計 400 7 GB メモリ A シリーズ コンピューティングと同等 292.00 USD/月 (推定)	 <b>P1</b> ACU 合計 100 1.75 GB メモリ A シリーズ コンピューティングと同等 219.00 USD/月 (推定)
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

適用

## App Service プラン

App Service プランの価格レベルによって、アプリに関連する場所、機能、コスト、コンピューティング リソースが決定されます。 [詳細情報](#)

Windows プラン (East US) \* ⓘ

(新規) ManagedPlatform



[新規作成](#)

SKU とサイズ \*

**Standard S1**

ACU 合計 100, 1.75 GB メモリ

[サイズを変更します](#)

[確認および作成](#)

< 前へ

次: 監視 >

# Web アプリの作成

基本    **監視**    タグ    確認および作成

Application Insights は、アプリケーションについて詳細な監視を行うためのコードなしのアタッチです。 [詳細情報](#) 

## Application Insights

Application Insights を有効にする \*

いいえ  はい

**確認および作成**

< 前へ

次: タグ >

# Web アプリの作成

基本 監視 タグ 確認および作成

## 概要



Web アプリ

Microsoft 提供

## 詳細

サブスクリプション	7b05aae1-3804-41bc-b616-1636a8711a81
リソース グループ	ManagedPlatform
名前	imgapiyamada
公開	コード
ランタイム スタック	.NET Core 3.1 (LTS)

## App Service プラン (新規)

名前	ASP-ManagedPlatform-b7a1
オペレーティング システム	Windows
地域	West US
SKU	Standard
サイズ	S
ACU	ACU 合計 100
メモリ	1.75 GB メモリ

戻る

作成

< 前へ

次へ >

Automation のテンプレートをダウンロードする

## タスク 5: ウェブ アプリの構成

1. この課題で前に作成した *imgapi[yourname]\*\** Web アプリにアクセスします。
2. 左側のメニューから 「構成」 をクリックし、次の詳細を使用して新しいアプリケーション設定を作成します。
  - 名前: StorageConnectionString
  - 値: ストレージの接続文字列
  - デプロイスロットの設定: 非選択
3. 変更内容をアプリケーション設定に保存します。
4. 左側のメニューから、「プロパティ」を見つけます。
5. プロパティ セクションで、URL フィールドの値をコピーします。この値は、このラボの後半で使用します。

ホーム >



# Microsoft.Web-WebApp-Portal-142deb34-b253 | 概要

デプロイ



削除



キャンセル



再デプロイ



最新の情報に更新



フィードバックをお待ちしています。→



デプロイが完了しました



デプロイ名: Microsoft.Web-WebApp-Portal-142deb34-b253

サブスクリプション: [Azure Pass - AZ204-2020-1214](#)

リソース グループ: [ManagedPlatform](#)

開始時刻: 2020/12/14 12:04:57

相関 ID: 2c9caf72-5211-4c50-99e6-31

▽ 展開の詳細 ([ダウンロード](#))

ヘ 次の手順

[リソースに移動](#)

# imgapiyamada | 構成

App Service

 検索 (Ctrl+ /)最新の情報に更新 保存 破棄

- 概要
- アクティビティログ
- アクセス制御 (IAM)
- タグ
- 問題の診断と解決
- セキュリティ
- イベント (プレビュー)

## デプロイメント

- クイック スタート
- デプロイ スロット
- デプロイ センター
- デプロイ センター (プレビュー)

## 設定

- 構成

- 認証/承認

アプリケーション設定 全般設定 既定のドキュメント パスのマッピング

## アプリケーション設定

アプリケーション設定は保存時に暗号化され、暗号化されたチャネルで送信されます。以下のコンソールのブラウザーにplainテキストで表示するよう選択することができます。アプリケーション設定でアクセスするための環境変数として公開されます。[詳細情報](#)

+ 新しいアプリケーション設定値を表示する高度な編集アプリケーション設定のフィルター処理

名前	値	ソース
----	---	-----

ANCM_ADDITIONAL_ERROR_PAGE_LINK	<span>非表示の値です。値を表示するには、<a href="#">アプリの構成</a>を参照してください。</span>
---------------------------------	----------------------------------------------------------------

## 接続文字列

接続文字列は保存時に暗号化され、暗号化されたチャネルで送信されます。

+ 新しい接続文字列値を表示する高度な編集

## アプリケーション設定の追加/編集

×

名前

StorageConnectionString



値

DefaultEndpointsProtocol=https;AccountName=imgstoryamada;AccountKey=haoj2bnjj3fBoHjLTE2UZtmnl7c...



デプロイ スロットの設定

チェックが付いていれば、外します。

OK

キャンセル

# imgapiyamada | 構成

App Service

検索 (Ctrl+ /)



最新の情報に更新



保存



破棄



概要



アクティビティ ログ



アクセス制御 (IAM)



タグ



問題の診断と解決

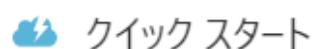


セキュリティ



イベント (プレビュー)

## デプロイメント



クイック スタート



デプロイ スロット



デプロイ センター



## アプリケーション設定 \*

### 全般設定

### 既定のドキュメント

### パスのマッピング

## アプリケーション設定

アプリケーション設定は保存時に暗号化され、暗号化されたチャネルで送信されまいのブラウザーにプレーン テキストで表示するよう選択することができます。アプリケーションでアクセスするための環境変数として公開されます。 [詳細情報](#)

+ 新しいアプリケーション設定

値を表示する

高度な編集

アプリケーション設定のフィルター処理

### 名前

### 値

ANCM\_ADDITIONAL\_ERROR\_PAGE\_LINK

非表示の値です。値を表示するには、ク

StorageConnectionString

非表示の値です。値を表示するには、ク

## 変更の保存

アプリケーション設定と接続文字列を変更すると、ア

続行

キャンセル

いのブラウザにフレーンテキストで表示するよつ選

# imgapiyamada | プロパティ

App Service

 検索 (Ctrl+ /)



状態

Running

 TLS/SSL の設定

 ネットワーク

 スケール アップ (App Service のプ...)

 スケール アウト (App Service のプ...)

 Web ジョブ

 プッシュ

 MySQL In App

 **プロパティ**

 ロック

## App Service プラン

 App Service プラン

 クオータ

 App Service プランの変更

URL

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

仮想 IP アドレス

40.112.243.47

WebアプリのURLをコピーして、  
メモ帳などに貼り付けしておきます。

モード

Standard

送信 IP アドレス

40.112.243.47,40.112.149.8,40.86.165.133,40.112.143.213,168.61.73.160

追加の送信 IP アドレス



## タスク 6: ASP.NET Web アプリケーションを Web アプリにデプロイする ステップ1~2

1. Visual Studio Code で、 Allfiles\Labs\01\Starter\API フォルダーを開きます。
2. Controllers\ImagesController.cs ファイルを開き、 各メソッドのコードを確認します。

最も一致する検索結果

Visual Studio Codeを起動します。



Visual Studio Code

アプリ

仕事および Web を検索

🔍 code - 仕事および Web の検索結果を表示 >

コマンド

🔍 code >

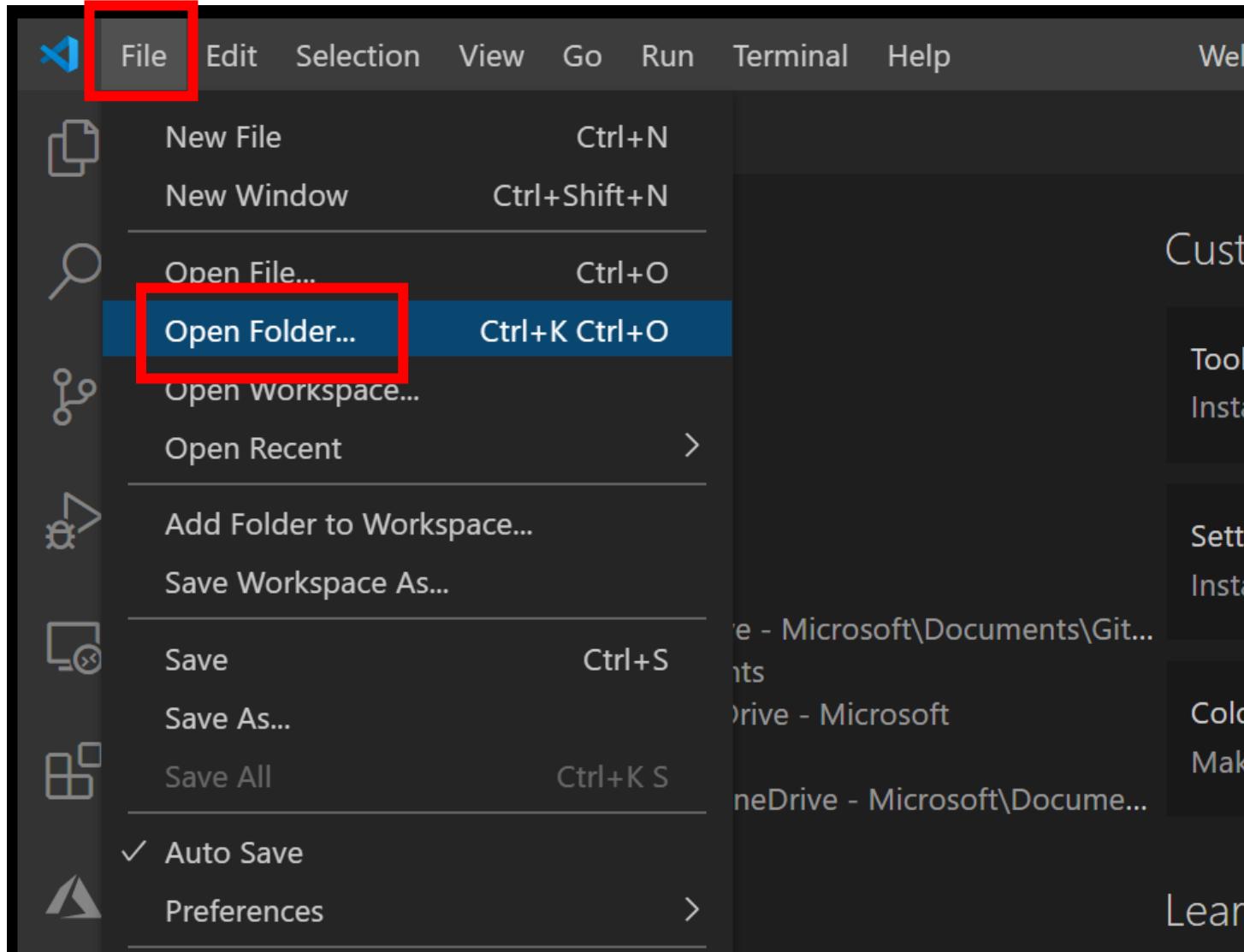
Microsoft の提案

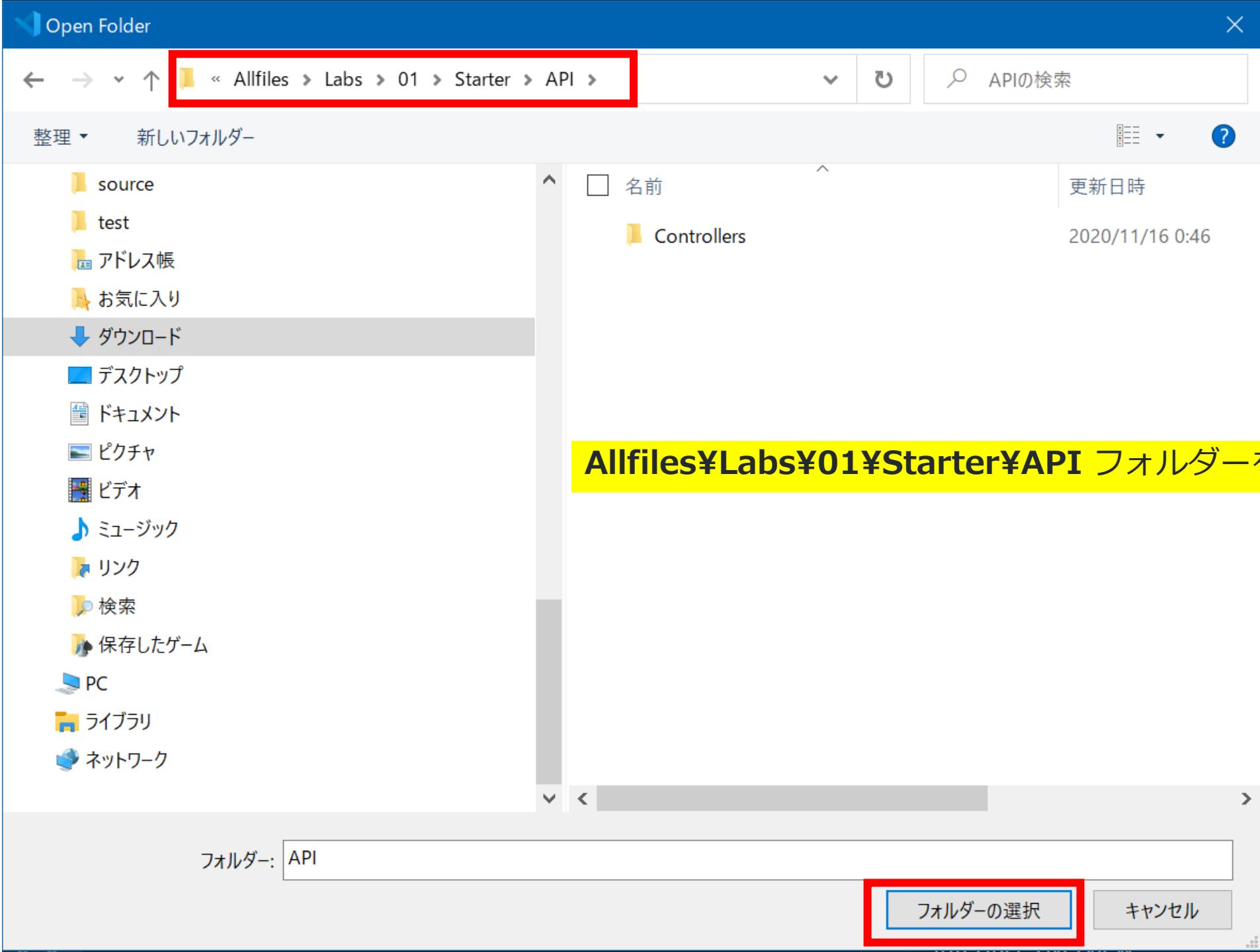
📘 CodeFlow Dashboard >

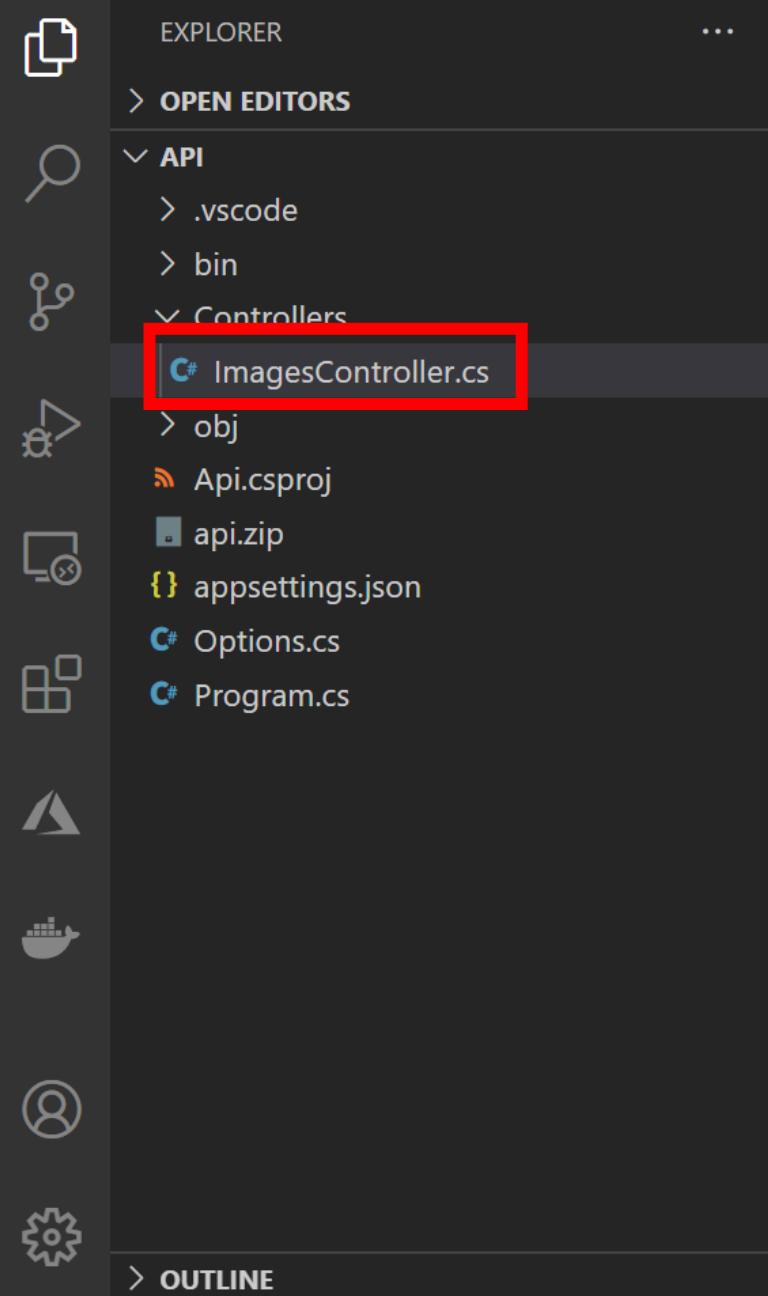
📘 Visual Studio Code >

ドキュメント - OneDrive for Business (1+)

🔍 code







## C# ImagesController.cs ×

Controllers &gt; C# ImagesController.cs &gt; ...

```
1  using Azure.Storage.Blobs;
2  using Azure.Storage.Blobs.Models;
3  using Flurl;
4  using Microsoft.AspNetCore.Mvc;
5  using System;
6  using System.Collections.Generic;
7  using System.IO;
8  using System.Net.Http;
9  using System.Threading.Tasks;
10
11 namespace Api.Controllers
12 {
13     [ApiController]
14     [Route("")]
15     public class ImagesController : ControllerBase
16     {
17         private HttpClient _httpClient;
18         private Options _options;
19
20         public ImagesController(HttpClient httpClient, Options options)
21         {
```

各メソッドの  
コードを確認します  
※理解できなくても大丈夫です

## タスク 6: ASP.NET Web アプリケーションを Web アプリにデプロイする

ステップ3~7

3. Windows Terminal アプリケーションを開きます。
4. Azure 認証情報を使用して Azure CLI にログインします。

```
az login
```

5. ManagedPlatform リソース グループ内のすべてのアプリを一覧表示します。

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

6. プレフィックス \*\*imgapi\*\*を持つアプリを見つけます。

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

7. プレフィックス imgapi\* を持つ単一のアプリの名前を表示します。

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

Windows Terminalを起動します。

最も一致する検索結果

 Windows Terminal  
アプリ

アプリ

■ コマンド プロンプト >

仕事および Web を検索 >

○ term - 仕事および Web の検索結果を表示 >

ドキュメント - PC (9+)

フォルダー (3+)

Microsoft の提案 (2)

人 (1+)

検索バー: ○ term

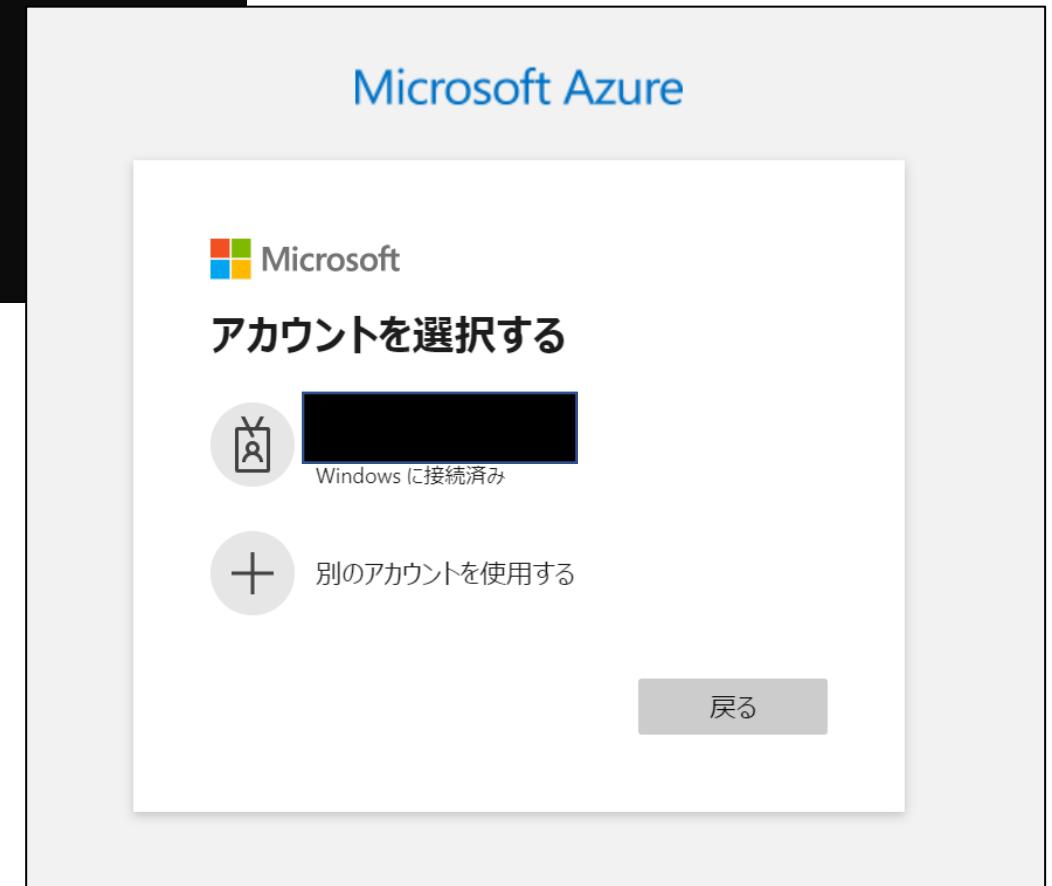
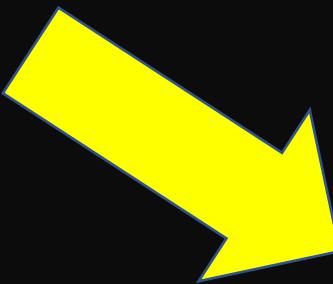
Windows PowerShell

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォームの PowerShell をお試しください。

PS C:\Users\hiryamada> az login

az login  
を実行します



Webブラウザが起動します。

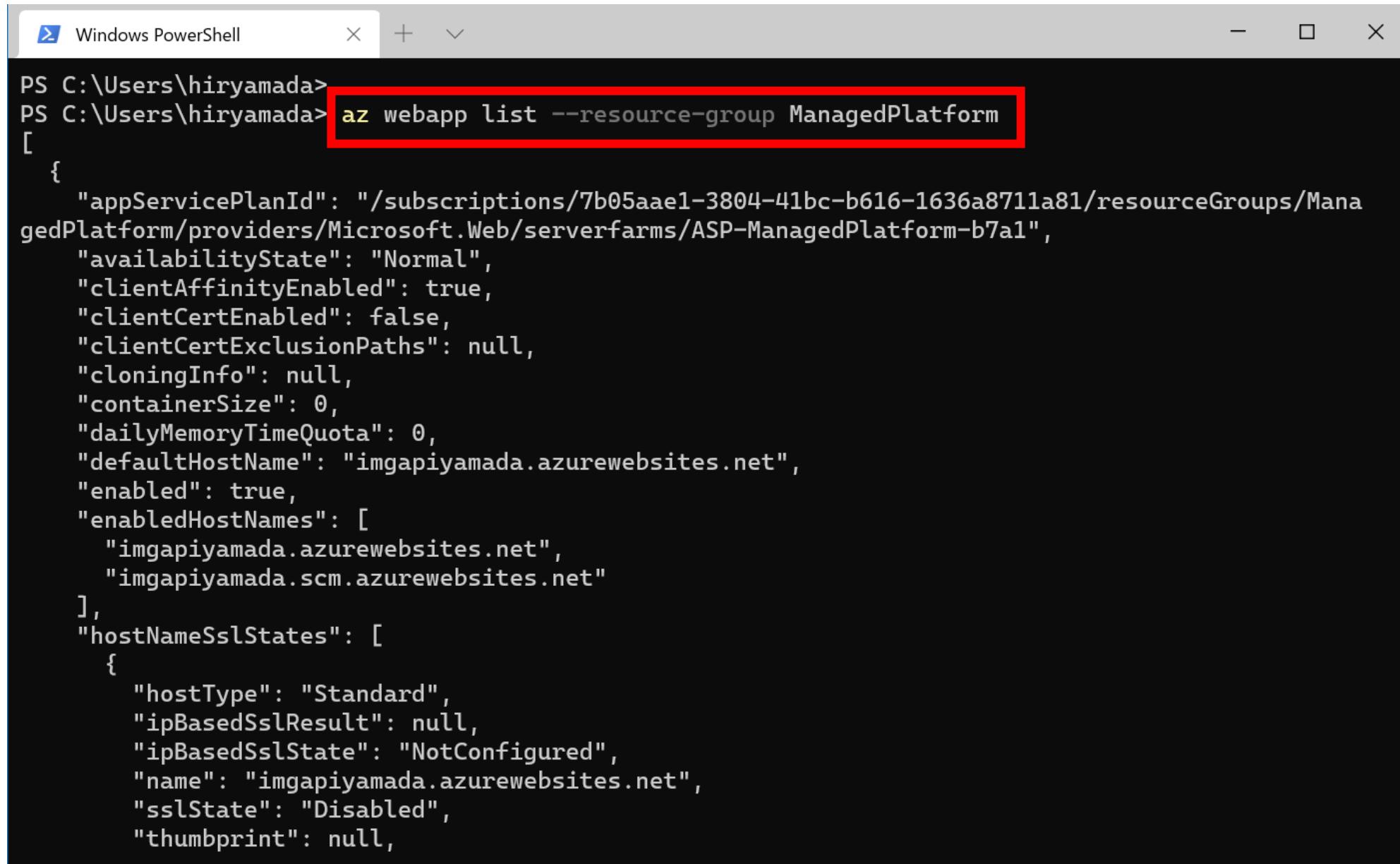
画面の指示に従い、  
トレーニング開始時に作成した  
Microsoftアカウントで  
ログインします。

以下のようなメッセージが出るので、  
しばらく待ちます。

```
PS C:\Users\hiriyamada> az login
```

```
You have logged in. Now let us find all the subscriptions to which you have access...
```

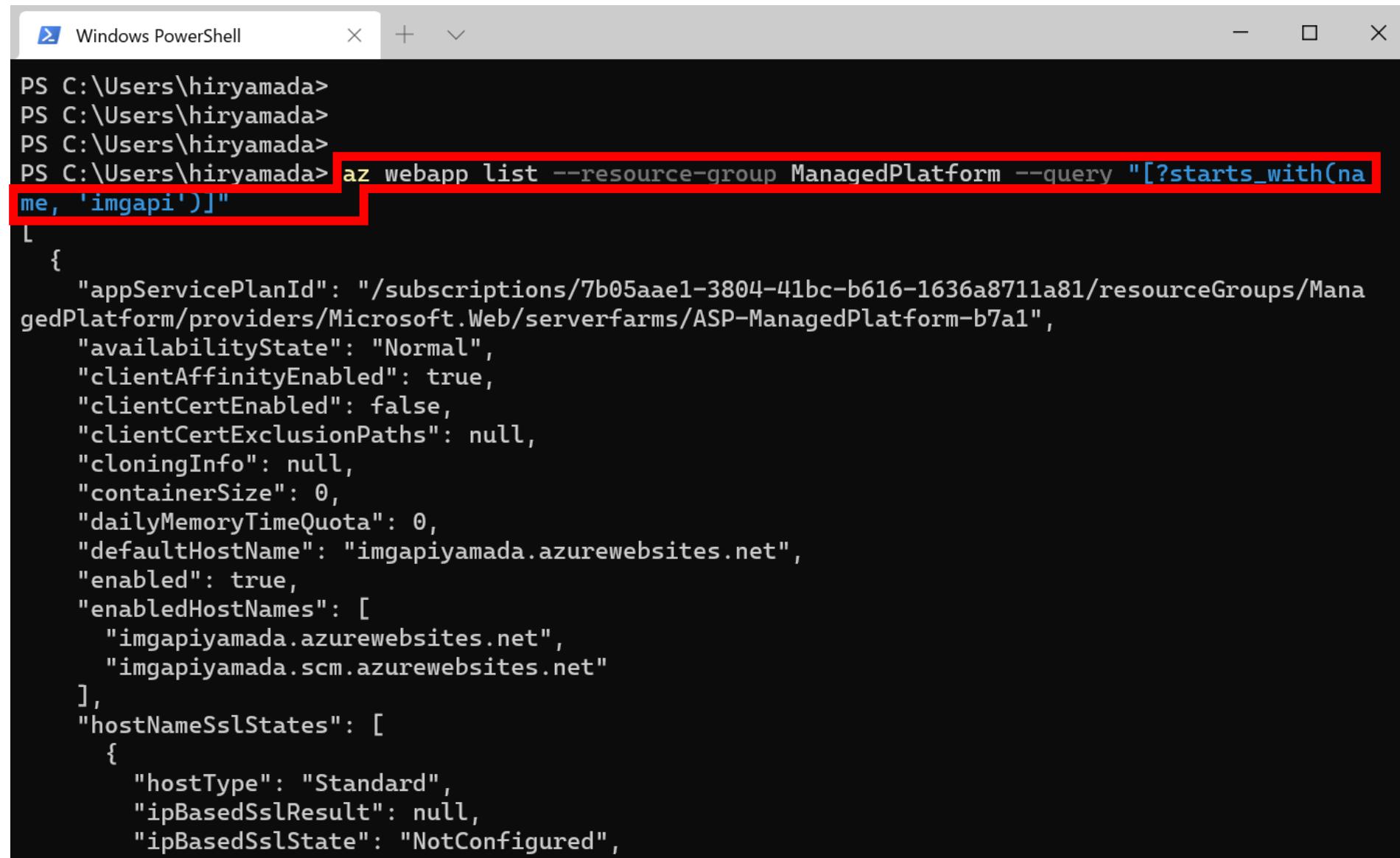
az webapp list --resource-group ManagedPlatform  
を実行します。



The screenshot shows a Windows PowerShell window with a black background and white text. The title bar says "Windows PowerShell". The command "az webapp list --resource-group ManagedPlatform" is highlighted with a red rectangle. The output of the command is displayed below, showing a JSON object representing a web application resource.

```
PS C:\Users\hiriyamada> PS C:\Users\hiriyamada> az webapp list --resource-group ManagedPlatform
[
  {
    "appServicePlanId": "/subscriptions/7b05aae1-3804-41bc-b616-1636a8711a81/resourceGroups/ManagedPlatform/providers/Microsoft.Web/serverfarms/ASP-ManagedPlatform-b7a1",
    "availabilityState": "Normal",
    "clientAffinityEnabled": true,
    "clientCertEnabled": false,
    "clientCertExclusionPaths": null,
    "cloningInfo": null,
    "containerSize": 0,
    "dailyMemoryTimeQuota": 0,
    "defaultHostName": "imgapiyamada.azurewebsites.net",
    "enabled": true,
    "enabledHostNames": [
      "imgapiyamada.azurewebsites.net",
      "imgapiyamada.scm.azurewebsites.net"
    ],
    "hostNameSslStates": [
      {
        "hostType": "Standard",
        "ipBasedSslResult": null,
        "ipBasedSslState": "NotConfigured",
        "name": "imgapiyamada.azurewebsites.net",
        "sslState": "Disabled",
        "thumbprint": null,
        "type": "Standard"
      }
    ]
  }
]
```

az webapp list --resource-group ManagedPlatform --query "[?starts\_with(name, 'imgapi')]"  
を実行します。



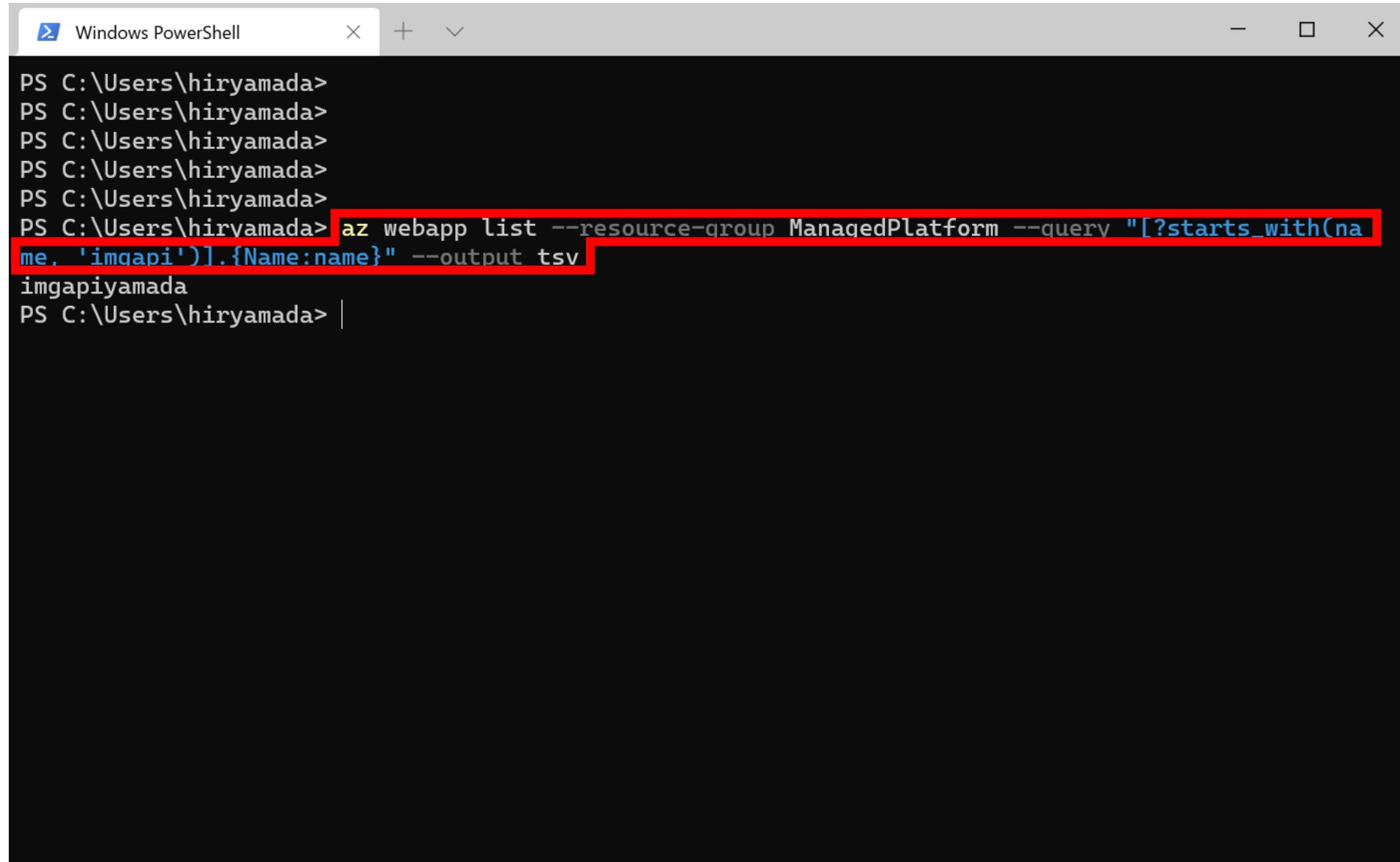
The screenshot shows a Windows PowerShell window with the title "Windows PowerShell". The command entered is:

```
PS C:\Users\hiriyamada> az webapp list --resource-group ManagedPlatform --query "[?starts_with(name, 'imgapi')]"
```

The output is a JSON object representing a single web app resource. The JSON structure includes properties such as appServicePlanId, availabilityState, clientAffinityEnabled, clientCertEnabled, clientCertExclusionPaths, cloningInfo, containerSize, dailyMemoryTimeQuota, defaultHostName, enabled, and enabledHostNames. The defaultHostName is listed as imgapiyamada.azurewebsites.net.

```
{
  "appServicePlanId": "/subscriptions/7b05aae1-3804-41bc-b616-1636a8711a81/resourceGroups/ManagedPlatform/providers/Microsoft.Web/serverfarms/ASP-ManagedPlatform-b7a1",
  "availabilityState": "Normal",
  "clientAffinityEnabled": true,
  "clientCertEnabled": false,
  "clientCertExclusionPaths": null,
  "cloningInfo": null,
  "containerSize": 0,
  "dailyMemoryTimeQuota": 0,
  "defaultHostName": "imgapiyamada.azurewebsites.net",
  "enabled": true,
  "enabledHostNames": [
    "imgapiyamada.azurewebsites.net",
    "imgapiyamada.scm.azurewebsites.net"
  ],
  "hostNameSslStates": [
    {
      "hostType": "Standard",
      "ipBasedSslResult": null,
      "ipBasedSslState": "NotConfigured"
    }
  ]
}
```

az webapp list --resource-group ManagedPlatform --query "[?starts\_with(name, 'imgapi')].{Name:name}" --output tsv  
を実行します。



The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command entered is:

```
PS C:\Users\hiryamada> az webapp list --resource-group ManagedPlatform --query "[?starts_with(name, 'imgapi')].{Name:name}" --output tsv
```

The output of the command is:

```
imgapiyamada
```

コマンドの出力に、作成したWebアプリの名前が表示されます。

この文字列は後ほど使用しますので、コピーして、メモ帳などにペーストしておきます。

```
PS C:\Users\hiriyamada> az webapp list -g myResourceGroup --query "[?name == 'imgapi'].{Name:name}" --output tsv  
imgapiyamada  
PS C:\Users\hiriyamada>
```

8. 現在のディレクトリを、演習ファイルを含むAllfiles\Labs\01\Starter\API ディレクトリに変更します。

```
cd <downloaded-files-dir>\Allfiles\Labs\01\Starter\API\
```

9. この課題で前に作成した Web アプリに **api.zip** ファイルをデプロイします。

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

注: <name-of-your-api-app> プレイスホルダーを、この課題で前に作成した Web アプリの名前に置き換えます。このアプリ名は、以前のステップで最近クエリしました。

10. この課題で前に作成した *imgapi[yourname]\*\** Web アプリにアクセスします。お使いのブラウザで *imgapi[yourname]\*\** ウェブアプリを開きます。

11. Web サイトのルートに GET 要求を実行し、返される JavaScript Object Notation (JSON) 配列を確認します。この配列には、Azure Storage アカウントにアップロードされた単一のイメージの URL が含まれている必要があります。

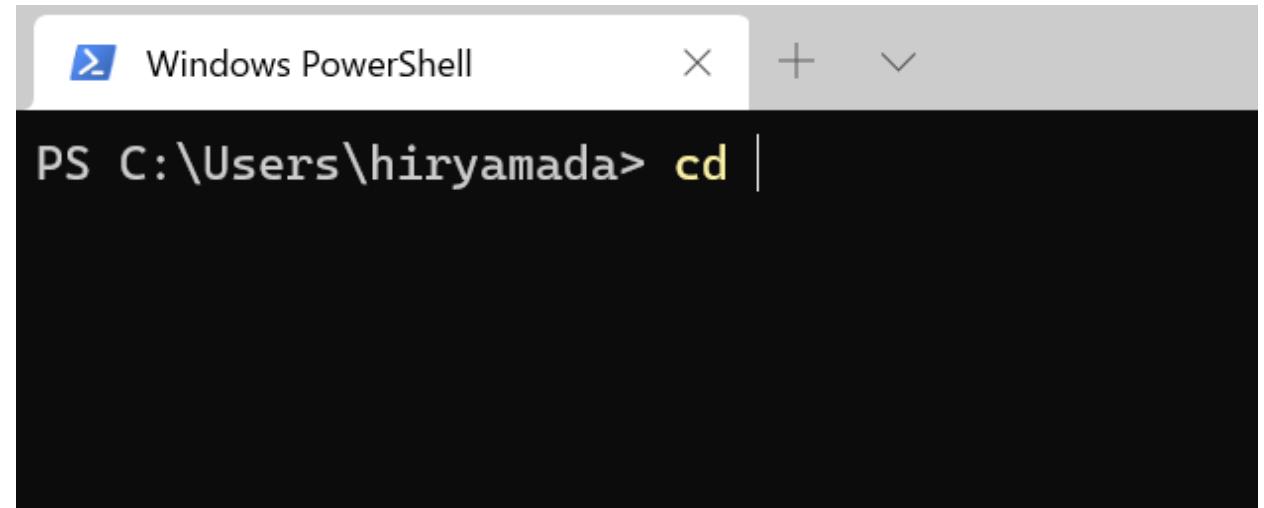
12. 現在実行中の Visual Studio Code アプリケーションと Windows Terminal アプリケーションを閉じます。

## レビュー

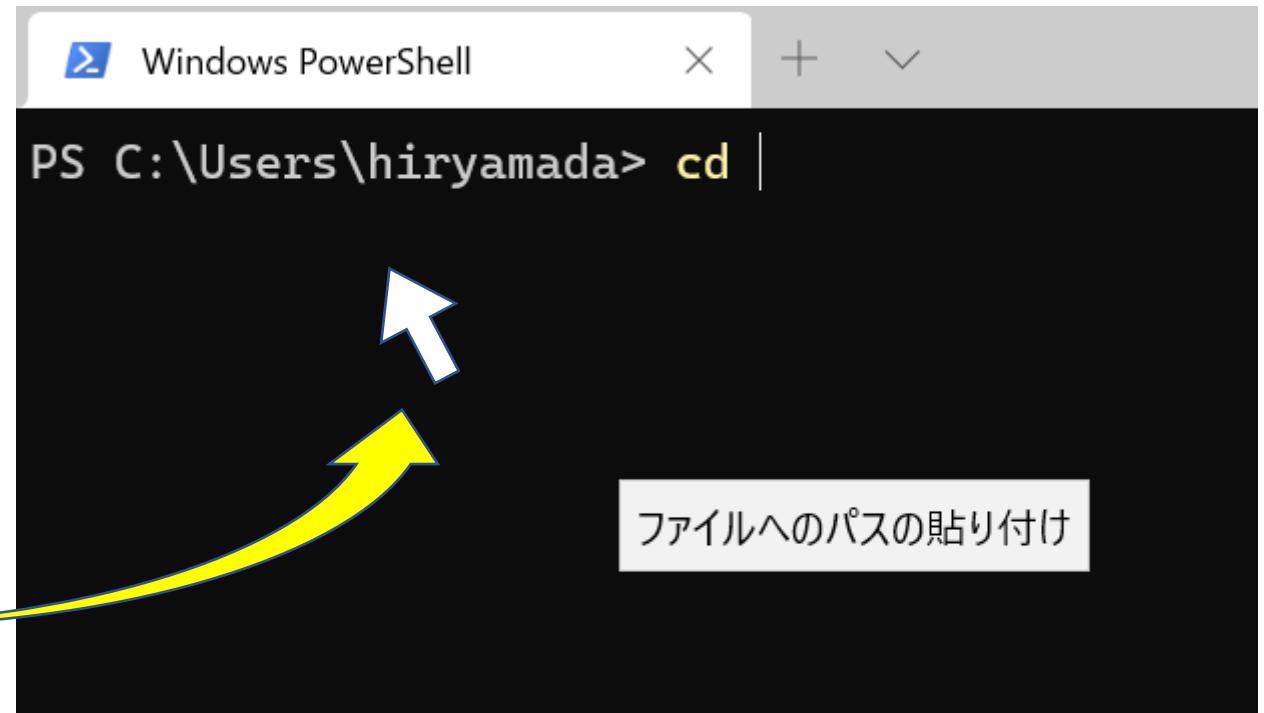
この演習では、Azure で Web アプリを作成し、Azure CLI と Kudu の zip ファイルデプロイメントユーティリティを使用して、ASP.NET Web アプリケーションを Web アプリにデプロイしました。

cd と入力し、スペースキーを押します  
(エンターキーは押さないでください)

Allfiles¥Labs¥01¥Starter¥API フォルダを  
ドラッグして、Windows Terminal にドロップします。  
(APIフォルダのパスが貼り付けされます)

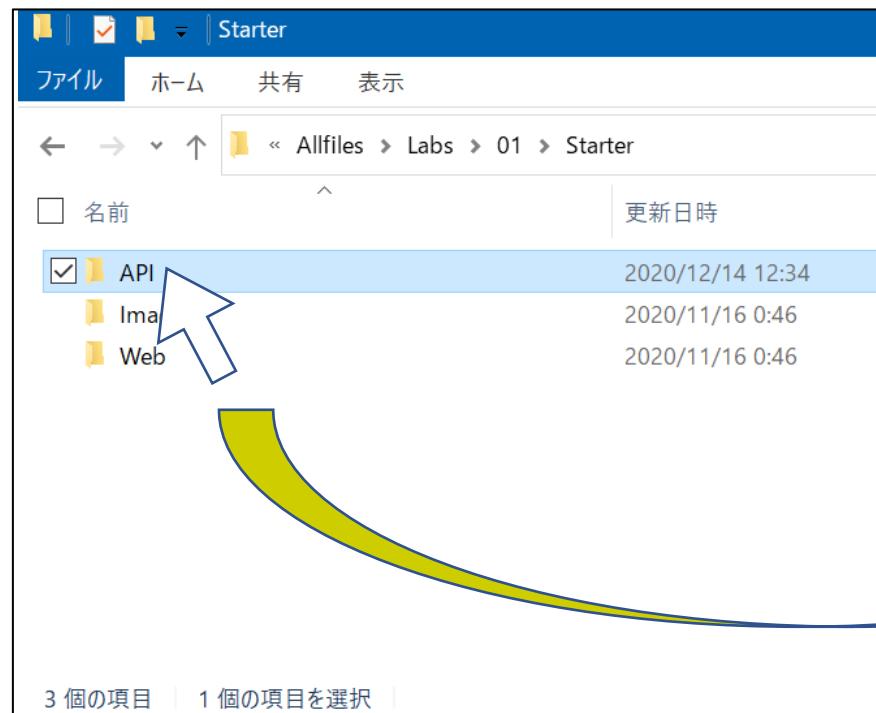


```
Σ Windows PowerShell
PS C:\Users\hiryamada> cd |
```



```
Σ Windows PowerShell
PS C:\Users\hiryamada> cd |
```

ファイルへのパスの貼り付け



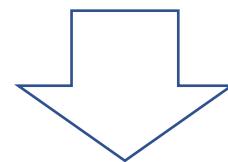
エンターキーを押します。

Windows PowerShell

X + ▾

- □ X

```
PS C:\Users\hiryamada> cd C:\Users\hiryamada\Downloads\AZ-204JA-DevelopingSolutionsforMicrosoftA  
zure-master\AZ-204JA-DevelopingSolutionsforMicrosoftAzure-master\Allfiles\Labs\01\Starter\API
```



Windows PowerShell

X + ▾

- □ X

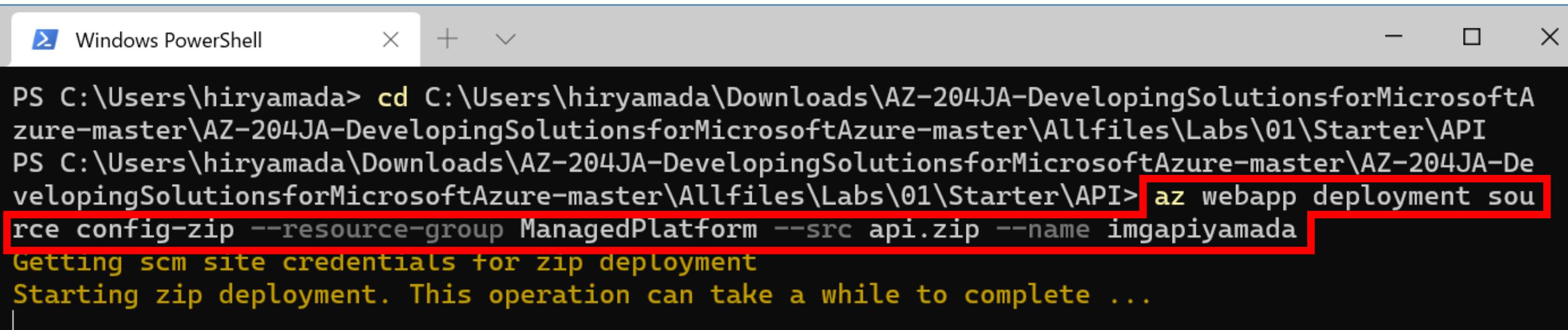
```
PS C:\Users\hiryamada> cd C:\Users\hiryamada\Downloads\AZ-204JA-DevelopingSolutionsforMicrosoftA  
zure-master\AZ-204JA-DevelopingSolutionsforMicrosoftAzure-master\Allfiles\Labs\01\Starter\API  
PS C:\Users\hiryamada\Downloads\AZ-204JA-DevelopingSolutionsforMicrosoftAzure-master\AZ-204JA-De  
velopingSolutionsforMicrosoftAzure-master\Allfiles\Labs\01\Starter\API> |
```

※カレントディレクトリが  
APIフォルダとなったことを確認します

以下のコマンドを実行します。

末尾の imgapiyamada の部分は、前の工程で確認したWebアプリの名前を指定します。

az webapp deployment source config-zip --resource-group ManagedPlatform --src api.zip --name **imgapiyamada**



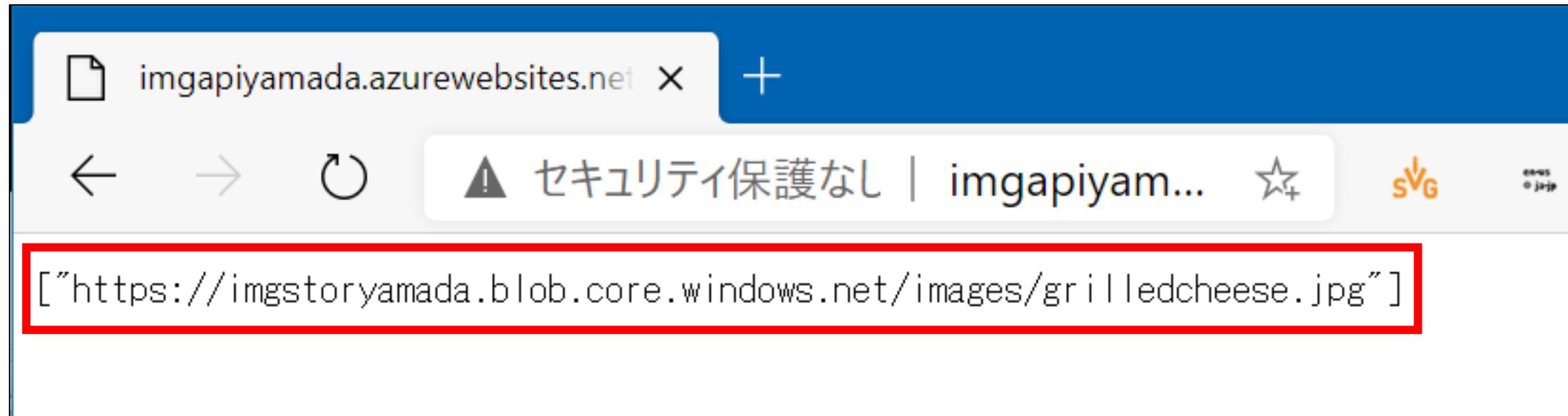
Windows PowerShell

```
PS C:\Users\hiriyamada> cd C:\Users\hiriyamada\Downloads\AZ-204JA-DevelopingSolutionsforMicrosoftAzure-master\AZ-204JA-DevelopingSolutionsforMicrosoftAzure-master\Allfiles\Labs\01\Starter\API
PS C:\Users\hiriyamada\Downloads\AZ-204JA-DevelopingSolutionsforMicrosoftAzure-master\AZ-204JA-DevelopingSolutionsforMicrosoftAzure-master\Allfiles\Labs\01\Starter\API> az webapp deployment source config-zip --resource-group ManagedPlatform --src api.zip --name imgapiyamada
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
```

前の工程でコピーしておいたWebアプリのURLをブラウザで開きます。

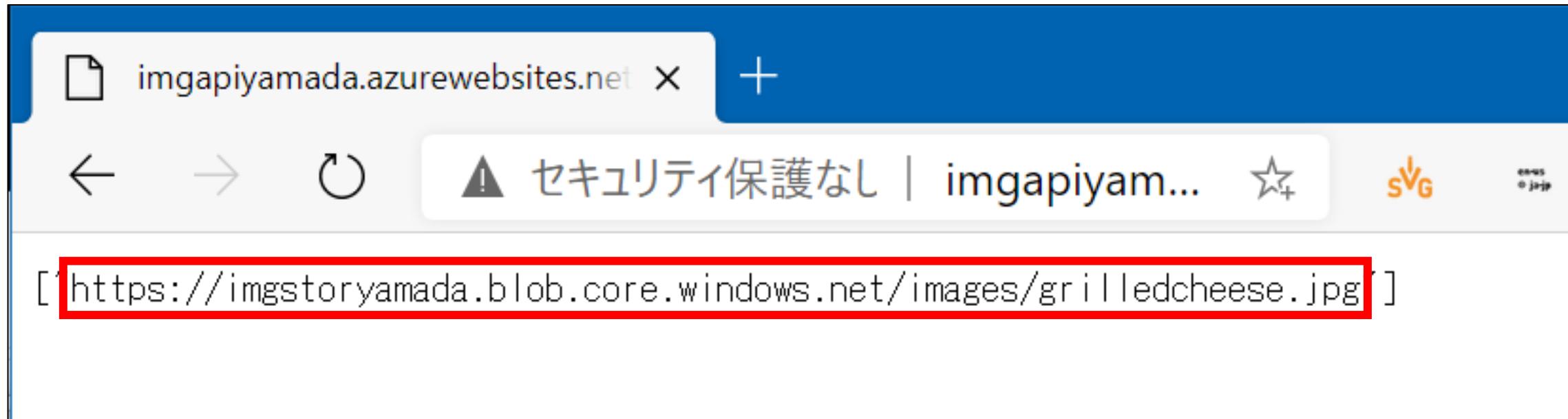
URL

imgapiyamada.azurewebsites.net

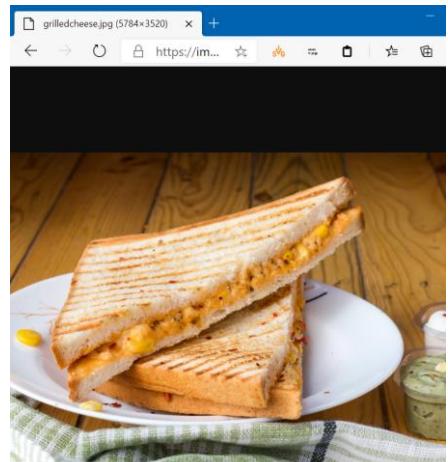


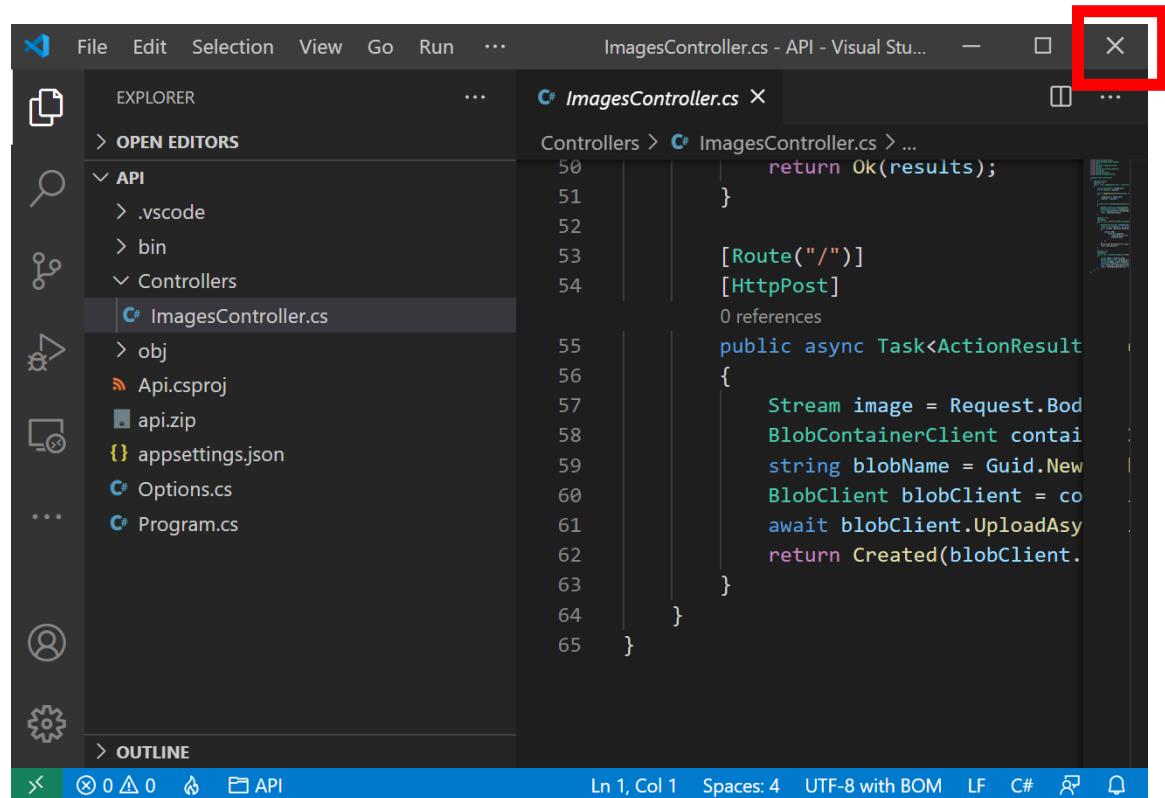
前の工程でアップロードした画像のURLが表示されることを確認します。

表示された画像のURLをコピーします。  
※ ["と"]の中の文字列をコピーします



コピーしたURLをWebブラウザで開きます。  
画像が表示されることを確認します。





File Edit Selection View Go Run ...

EXPLORER

> OPEN EDITORS

API

> .vscode

> bin

< Controllers

ImagesController.cs

obj

Api.csproj

api.zip

appsettings.json

Options.cs

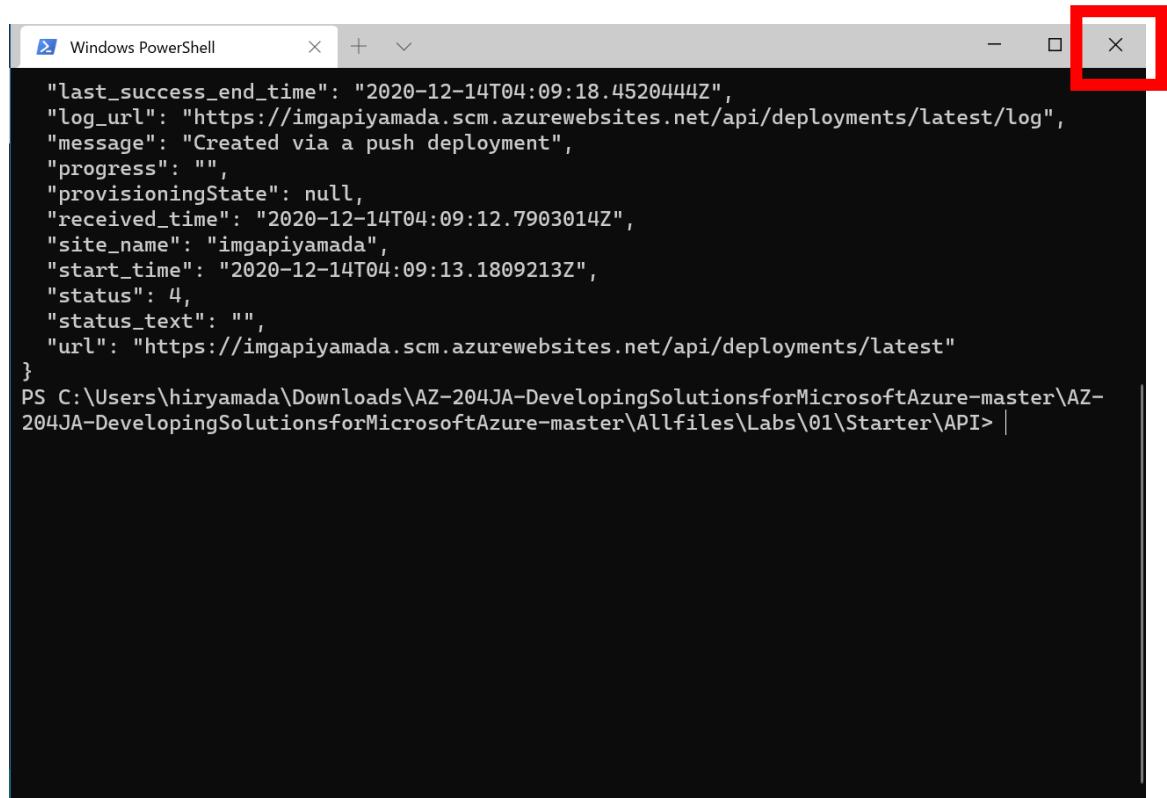
Program.cs

OUTLINE

Ln 1, Col 1 Spaces: 4 UTF-8 with BOM LF C# ⚡

C# ImagesController.cs - API - Visual Studio

```
50         return Ok(results);
51     }
52
53     [Route("/")]
54     [HttpPost]
55     public async Task<ActionResult>
56     {
57         Stream image = Request.Body;
58         BlobContainerClient containerClient =
59         string blobName = Guid.NewGuid().ToString();
60         BlobClient blobClient = containerClient.
61         await blobClient.UploadAsynchronously(image);
62         return Created(blobClient.
63             };
64     }
65 }
```



Windows PowerShell

```
"last_success_end_time": "2020-12-14T04:09:18.4520444Z",
"log_url": "https://imgapiyamada.scm.azurewebsites.net/api/deployments/latest/log",
"message": "Created via a push deployment",
"progress": "",
"provisioningState": null,
"received_time": "2020-12-14T04:09:12.7903014Z",
"site_name": "imgapiyamada",
"start_time": "2020-12-14T04:09:13.1809213Z",
"status": 4,
"status_text": "",
"url": "https://imgapiyamada.scm.azurewebsites.net/api/deployments/latest"
}
PS C:\Users\hiriyamada\Downloads\AZ-204JA-DevelopingSolutionsforMicrosoftAzure-master\AZ-204JA-DevelopingSolutionsforMicrosoftAzure-master\Allfiles\Labs\01\Starter\API> |
```

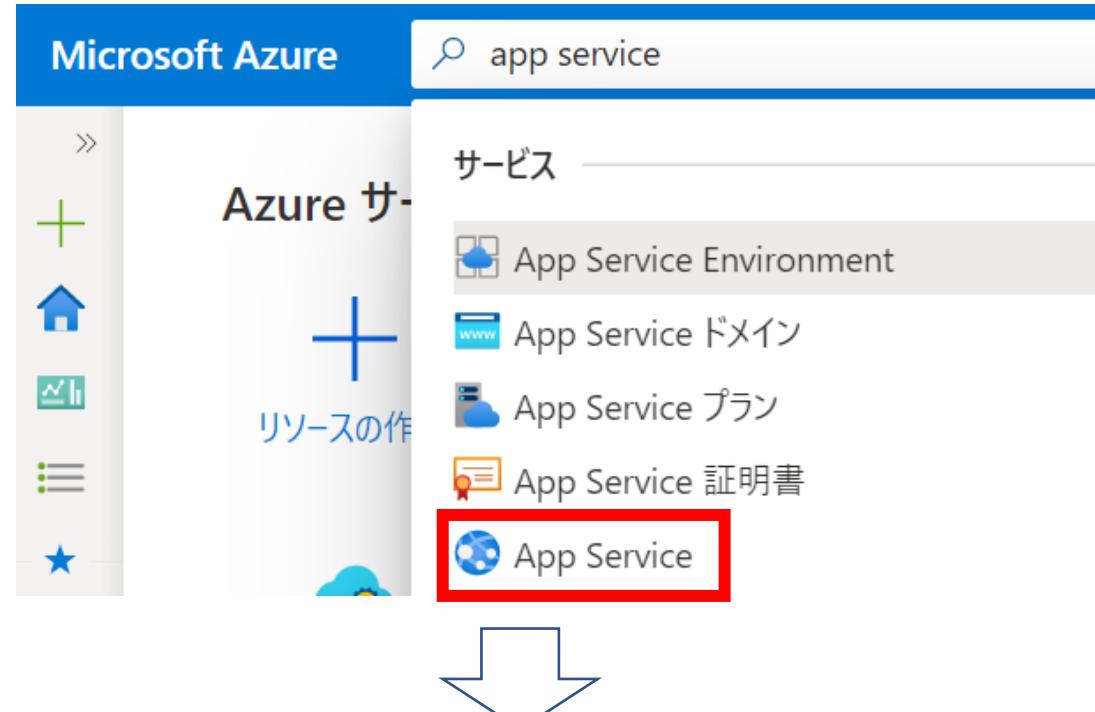
演習 2: Azure Web アプリを使用してフロントエンド Web アプリケーションを構築する

## タスク 1: Web アプリを作成する

1. Azure Portal で、新しい Web アプリを作成します。詳細は次の通りです:

- 既存のリソース グループ: **ManagedPlatform**
- Web アプリの名前: *imgweb[yourname]*\*\*
- 公開: **コード**
- ランタイム スタック: **.NET Core 3.1(LTS)**
- オペレーティング システム: **Windows**
- リージョン: **米国東部**
- 既存の App Services プラン: **ManagedPlan**
- Application Insights: **無効**

2. Azure が Web アプリの作成を完了するのを待ってから、ラボに進みます。アプリの作成時に通知が届きます。



Microsoft Azure

リソース、サービス、ドキュメントの検索 (G-)

ホーム >

## App Service

AZ204-2020-1214

+ 追加 ビューの管理 更新 CSV に

名前でフィルター... フィルターの追加

1 件中 1 ~ 1 件のレコードを表示しています。

リストビュー

This screenshot shows the 'App Service' blade in the Microsoft Azure portal. At the top, it displays the service name 'AZ204-2020-1214'. Below the service name, there is a toolbar with several buttons: '+ 追加' (highlighted with a red box), 'ビューの管理', '更新', and 'CSV に'. There are also two input fields: '名前でフィルター...' and 'フィルターの追加'. At the bottom, a message indicates '1 件中 1 ~ 1 件のレコードを表示しています.' and a dropdown menu shows 'リストビュー'.

## Web アプリの作成

基本    監視    タグ    確認および作成

App Service Web Apps を使用すると、任意のプラットフォームで実行するエンタープライズ レベルの Web アプリ、モバイル アプリ、API アプリを素早くビルド、デプロイ、スケーリングできます。フル マネージド プラットフォームを使用してインフラストラクチャ メンテナンスを実行するには、パフォーマンス、スケーラビリティ、セキュリティおよびコンプライアンスの要件を確実に満たしてください。 [詳細情報](#)

### プロジェクトの詳細

デプロイされているリソースとコストを管理するサブスクリプションを選択します。  
を整理し、管理します。

「リソース グループ」のプルダウンを  
クリックし、ManagedPlatformをクリックします。

サブスクリプション \* ⓘ

Azure Pass - AZ204-2020-1214



リソース グループ \* ⓘ

[既存のものを選択...]



### インスタンスの詳細

名前 \*

DefaultResourceGroup-WUS

ManagedPlatform



確認および作成

< 前へ

次: 監視 >

## インスタンスの詳細

名前 \*

imgapiyamada2



.azurewebsites.net

公開 \*



ランタイム スタック \*

.NET Core 3.1 (LTS)



オペレーティング システム \*



地域 \*

East US



App Service プランが見つからない場合、別のリージョンをお試しください。

## App Service プラン

App Service プランの価格レベルによって、アプリに関連する場所、機能、コスト、コンピューティング リソースが決定されます。[詳細情報](#)

Windows プラン (East US) \*

ManagedPlatform (S1)



[新規作成](#)

SKU とサイズ \*

**Standard S1**

ACU 合計 100, 1.75 GB メモリ

[確認および作成](#)

< 前へ

次: 監視 >

## Web アプリの作成

基本    監視    タグ    確認および作成

Application Insights は、アプリケーションについて詳細な監視を行うためのコードなしのアタッチです。 [詳細情報](#)

### Application Insights

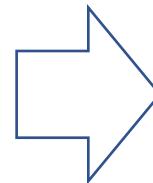
Application Insights を有効にする \*

いいえ  はい

**確認および作成**

< 前へ

次: タグ >



## Web アプリの作成

基本    監視    タグ    **確認および作成**

### 概要



#### Web アプリ

Microsoft 提供

### 詳細

サブスクリプション	7b05aae1-3804-41bc-b616-1636a8711a81
リソース グループ	ManagedPlatform
名前	imgapiyamada2
公開	コード
ランタイム スタック	.NET Core 3.1 (LTS)

### App Service プラン

名前	ManagedPlatform
オペレーティング システム	Windows
地域	East US
SKU	Standard

**作成**

< 前へ

次へ >

Automation のテンプレートをダウンロードする

## タスク 2: Web アプリを構成する

1. 前のタスクで作成した *imgweb[yourname]\*\** Web アプリにアクセスします。
2. 左側のメニューから、「**構成**」を選択します。
3. 次の詳細を使用して、新しい**アプリケーション設定**を作成します。
  - 名前: **ApiUrl**
  - 値: 以前にこのタブでコピーした Web アプリの URL
  - デプロイ スロットの設定: **非選択**
4. 変更内容を**アプリケーション設定**に保存します。

**注:** `https://`などのプロトコルが、このアプリケーション設定の「**値**」テキストボックスにコピーする URL に含まれていることを確認します。

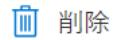
ホーム >



## Microsoft.Web-WebApp-Portal-3a949c61-95f5 | 概要

デプロイ

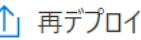
検索 (Ctrl+ /)



削除



キャンセル



再デプロイ



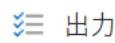
最新の情報に更新



概要



入力



出力



テンプレート



フィードバックをお待ちしています。→



デプロイが完了しました



デプロイ名: Microsoft.Web-WebApp-Portal-3a949c61-95f5 開始時

相関 ID

サブスクリプション: Azure Pass - AZ204-2020-1214

リソース グループ: ManagedPlatform

▽ 展開の詳細 (ダウンロード)

ヘ 次の手順

リソースに移動

# imgapiyamada2 | 構成

App Service

X

検索 (Ctrl+ /)

最新の情報に更新 保存 破棄

- 概要
- アクティビティログ
- アクセス制御 (IAM)
- タグ
- 問題の診断と解決
- セキュリティ
- イベント (プレビュー)

## デプロイメント

- クイック スタート
- デプロイスロット
- デプロイセンター
- デプロイセンター (プレビュー)

## 設定

構成

認証/承認

Application Insights

アプリケーション設定 全般設定 既定のドキュメント パスのマッピング

## アプリケーション設定

アプリケーション設定は保存時に暗号化され、暗号化されたチャネルで送信されます。以下のコントロールを使用して、お使いのブラウザーにプレーンテキストで表示するよう選択することができます。アプリケーション設定は、実行時にアプリケーションでアクセスするための環境変数として公開されます。[詳細情報](#)

+ 新しいアプリケーション設定

値を表示する

高度な編集

アプリケーション設定のフィルター処理

名前	値	ソース
ANCM_ADDITIONAL_ERROR_PAGE_LINK	非表示の値です。値を表示するには、 <a href="#">ク アプリ</a>	

## 接続文字列

接続文字列は保存時に暗号化され、暗号化されたチャネルで送信されます。

+ 新しい接続文字列

値を表示する

高度な編集

接続文字列のフィルター処理



## アプリケーション設定の追加/編集

名前

StorageConnectionString

値

{==;EndpointSuffix=core.windows.net}

デプロイ スロットの設定

チェックが付いていれば、外します。

OK

キャンセル

# imgwebyamada | 構成

App Service

X

検索 (Ctrl+ /)

最新の情報に更新

保存

破棄

セキュリティ

イベント (プレビュー)

## デプロイメント

クイック スタート

デプロイ スロット

デプロイ センター

デプロイ センター (プレビュー)

## 設定

構成

認証/承認

Application Insights

ID

バックアップ

カスタム ドメイン

TLS/SSL の設定

アプリケーション設定 \* 全般設定 既定のドキュメント パスのマッピング

## アプリケーション設定

アプリケーション設定は保存時に暗号化され、暗号化されたチャネルで送信されます。以下のコントロールを使用して、お使いのブラウザーにプレーンテキストで表示するよう選択することができます。アプリケーション設定は、実行時にアプリケーションでアクセスするための環境変数として公開されます。 [詳細情報](#)

+ 新しいアプリケーション設定 [値を表示する](#) [高度な編集](#)

▼ アプリケーション設定のフィルター処理

名前	値	ソース
ANCM_ADDITIONAL_ERROR_PAGE_LINK	<a href="#">非表示の値です。値を表示するには、ク アプリ</a>	
StorageConnectionString	<a href="#">非表示の値です。値を表示するには、ク アプリ</a>	

## 接続文字列

接続文字列は保存時に暗号化され、暗号化されたチャネルで送信されます。

## 変更の保存

×

アプリケーション設定と接続文字列を変更すると、アプリケーションが再起動します。  
続行しますか?

続行

キャンセル

# imgwebyamada | プロパティ

App Service

 検索 (Ctrl+ /)



状態

Running



認証/承認



Application Insights



ID



バックアップ



カスタム ドメイン



TLS/SSL の設定



ネットワーク



スケール アップ (App Service のプ...



スケール アウト (App Service のプ...



Web ジョブ



プッシュ



MySQL In App



プロパティ



ロック

App Service プラン

App Service プラン

URL

imgwebyamada.azurewebsites.net

仮想 IP アドレス

20.49.104.16

URLをコピーして、  
メモ帳などに貼り付けておきます

モード

Standard

送信 IP アドレス

20.62.212.169,20.62.212.170,20.62.212.185,20.62.212.198,20.62.212.214,20.62.212.222,20....



追加の送信 IP アドレス



20.62.212.169,20.62.212.170,20.62.212.185,20.62.212.198,20.62.212.214,20.62.212.222,20....



FTP/デプロイ ユーザー

FTP/デプロイ ユーザー セットがありません



## タスク 3: ASP.NET Web アプリケーションを Web アプリにデプロイする

ステップ1~2

1. Visual Studio Code を使用して、Allfiles\Labs\01\Starter\Web フォルダーにある Web アプリケーションを開きます。
2. Pages\Index.cshtml.cs ファイルを開き、各メソッドのコードを確認します。

タスク2と同様の操作を行います。  
タスク2の手順を参照しながら実施してください。

## タスク 3: ASP.NET Web アプリケーションを Web アプリにデプロイする

ステップ3~6

3. Windows ターミナル アプリケーションを開き、Azure の資格情報を使用して Azure CLI にログインします。

```
az login
```

4. ManagedPlatform リソース グループ内のすべてのアプリを一覧表示します。

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

5. プレフィックス imgweb\* を持つアプリを見つけます。

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

6. プレフィックス imgweb\* を持つ単一のアプリの名前を表示します。

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

タスク2と同様の操作を行います。  
タスク2の手順を参照しながら実施してください。

## タスク 3: ASP.NET Web アプリケーションを Web アプリにデプロイする

ステップ7~9

7. 現在のディレクトリを、課題ファイルを含む Allfiles\Labs\01\Starter\Web ディレクトリに変更します。

```
cd <downloaded-files-dir>\Allfiles\Labs\01\Starter\Web\
```

8. この実習ラボで前に作成した Web アプリ に web.zip ファイルを展開します。

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

注: <name-of-your-web-app> プレイスホルダーを、この実習ラボで前に作成した Web アプリの名前に置き換えます。このアプリ名は、以前のステップで最近クエリしました。

9. この実習ラボで前に作成した *imgweb[yourname]\*\** Web アプリにアクセスします。お使いのブラウザで *imgweb[yourname]\*\** ウェブアプリを開きます。

タスク2と同様の操作を行います。  
タスク2の手順を参照しながら実施してください。

10. Contoso フォト ギャラリー Web ページから 「新しい画像をアップロードする」 セクションを見つけ、  
Allfiles\Labs\01\Starter\Images ラボ マシンの フォルダーに記載された**bahnmi.jpg** に という名前のファイルをアップロードします。

| 注: 「アップロード」 ボタンをクリックして、イメージを Azure にアップロードしてください。

11. ギャラリー イメージの一覧が新しいイメージで更新されたことを確認します。

| 注: まれに、新しいイメージを表示するためにブラウザー ウィンドウを更新する必要があります。

12. 現在実行中の Visual Studio Code アプリケーションと Windows Terminal アプリケーションを閉じます。

## エクササイズ 3: サブスクリプションのクリーンアップ

### タスク 1: Azure Cloud Shell を開く

1. Azure portal の上部で、**Cloud Shell** アイコンを選択して新しいシェルインスタンスを開きます。
2. Cloud Shell がまだ構成されていない場合は、既定の設定を使用して Bash のシェルを構成します。
3. **Cloud Shell** コマンドプロンプトのポータルの下部にある次のコマンドを入力し、Enter キーを押してサブスクリプション内のすべてのリソースグループを一覧表示します。

```
az group list
```

4. 次のコマンドを入力し、Enter キーを押して、リソースグループを削除する可能性のあるコマンドの一覧を表示します。

```
az group delete --help
```

### タスク 2: リソース グループを削除する

1. 次のコマンドを入力し、Enter キーを押して **ManagedPlatform** リソース グループを削除します。

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

2. ポータルの Cloud Shell ペインを閉じます。

### タスク 3: アクティブなアプリケーションを閉じる

- 現在実行中の Microsoft Edge アプリケーションを閉じます。