



中原大學 雲端計算平台實務

11/18-作業報告

Administer containers in Azure

資訊碩一 11177035 林彥輝

授課教師：鍾武君 教授

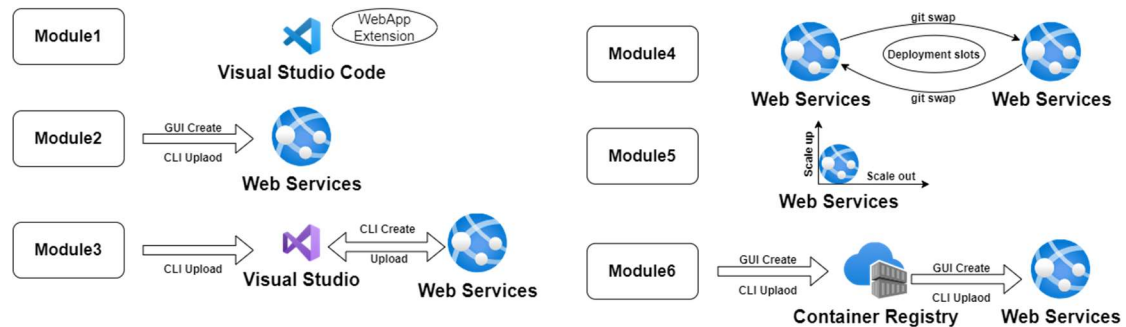
中華民國一一一年十一月

# 1. Model Intro

Deploy a website to Azure with Azure App Service

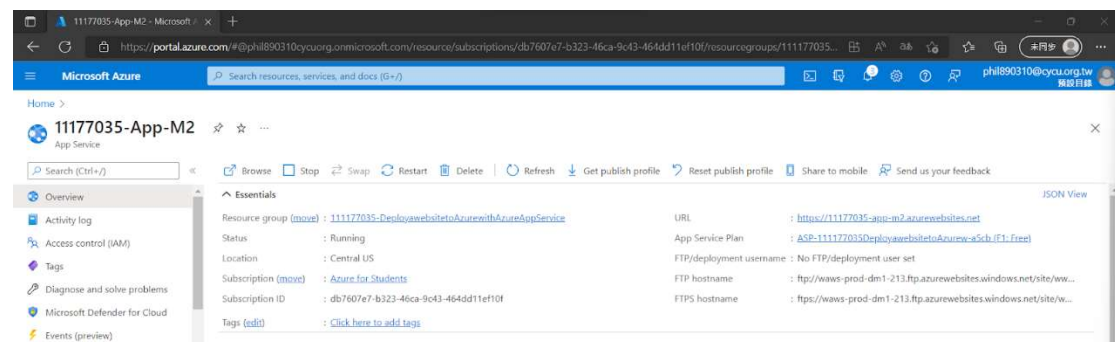
<https://learn.microsoft.com/en-us/training/paths/deploy-a-website-with-azure-app-service/>

## 2. Summary Homework Assignment

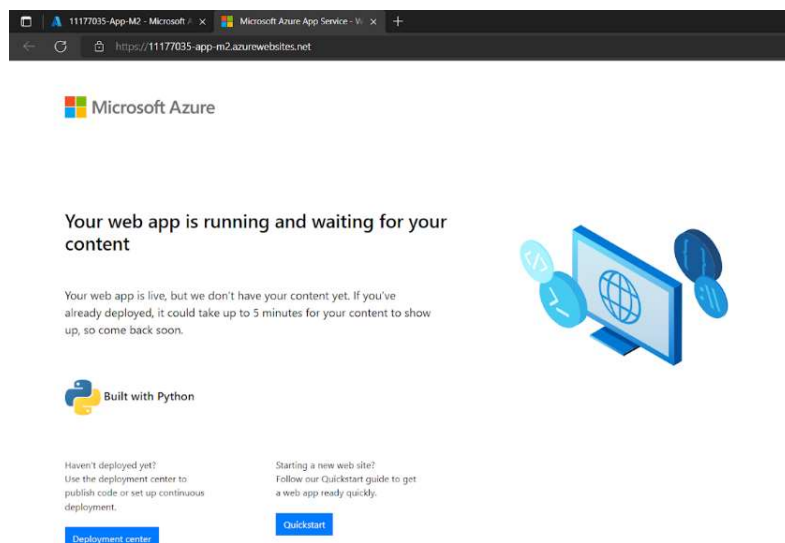


## Module 2: Host a web application with Azure App Service

1. Create a web app in the Azure portal



2. Test web app



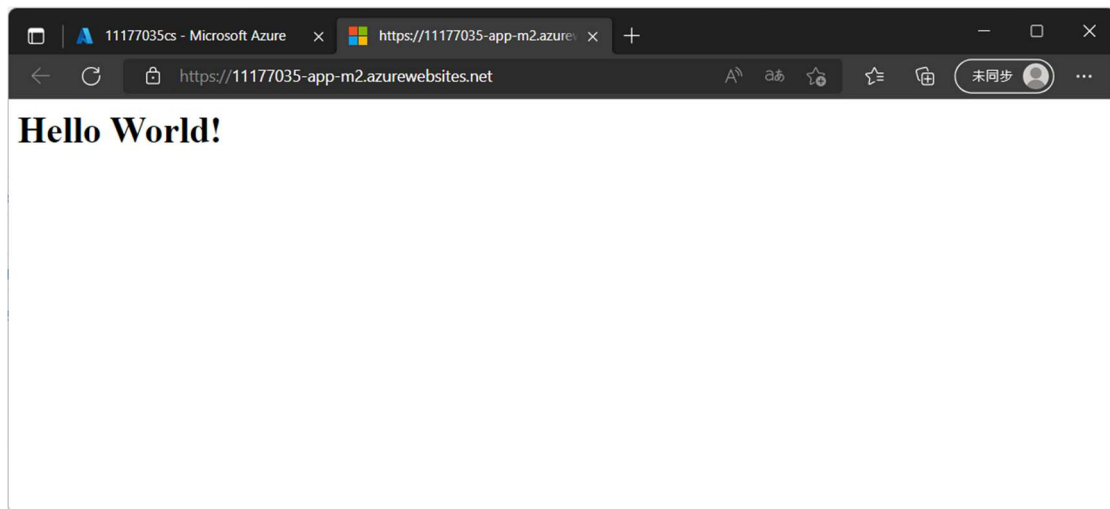
### 3. Create a new web project (local)

```
Bash
npm WARN A complete log of this run can be found in:
npm WARN /home/phil890310/.npm/_logs/2022-08-06T08_37_48_debug.log
phil890310@Azure:~/helloworld$ code .
phil890310@Azure:~/helloworld$ npm start

> helloworld@1.0.0 start /home/phil890310/helloworld
> node index.js

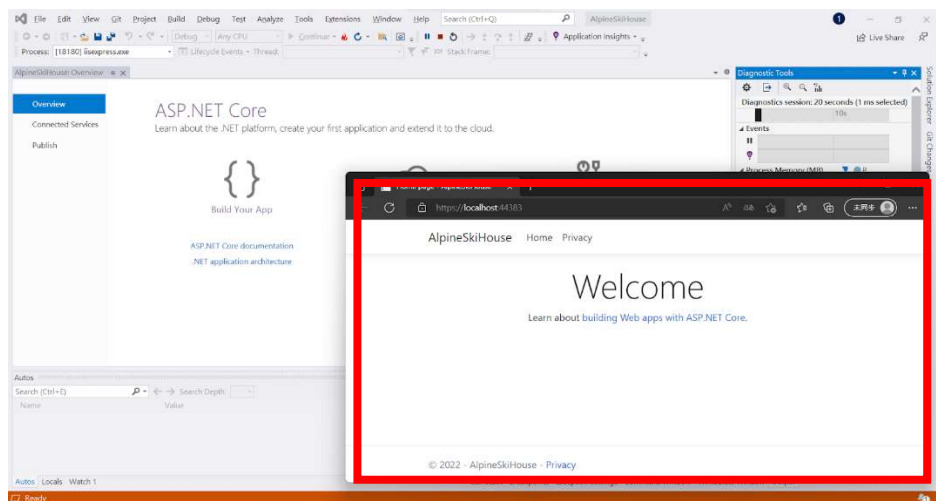
Server running at http://localhost:1337
```

### 4. Deploy to Azure web app

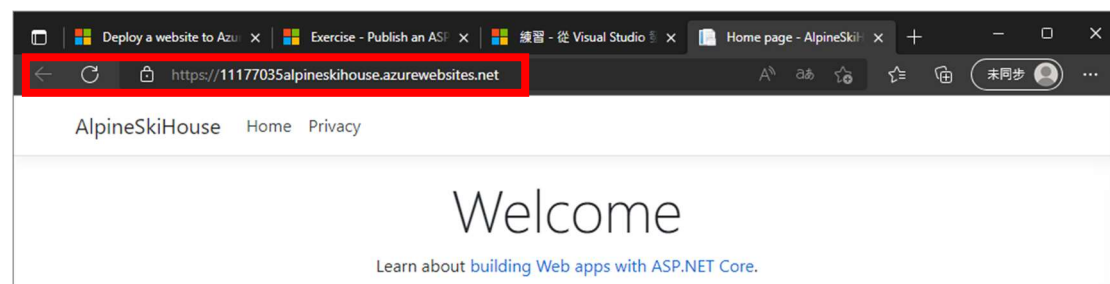
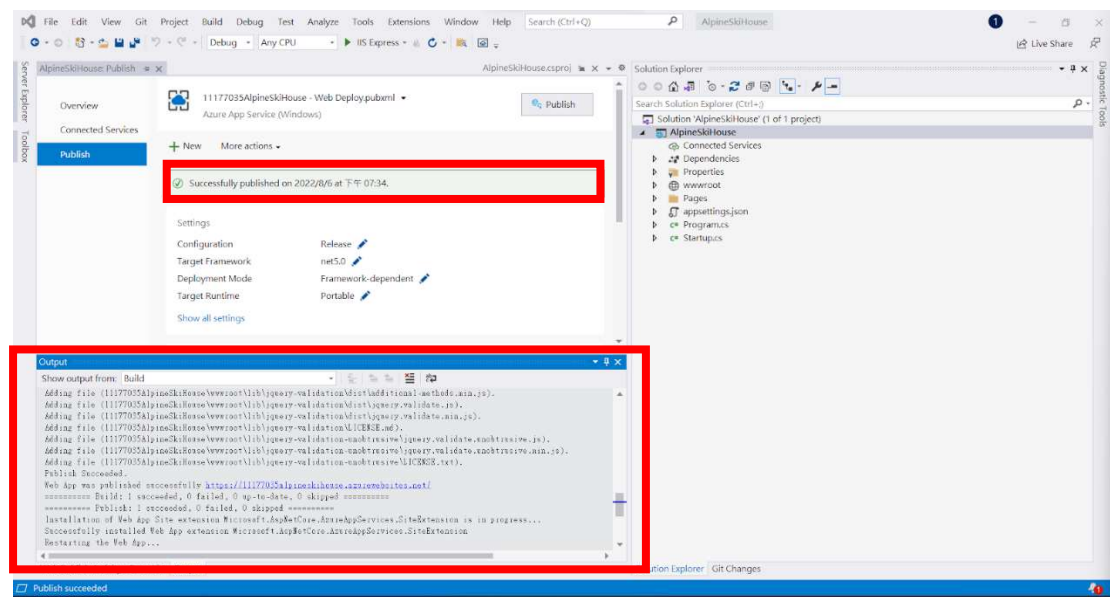


## Module 3: Publish a web app to Azure with Visual Studio

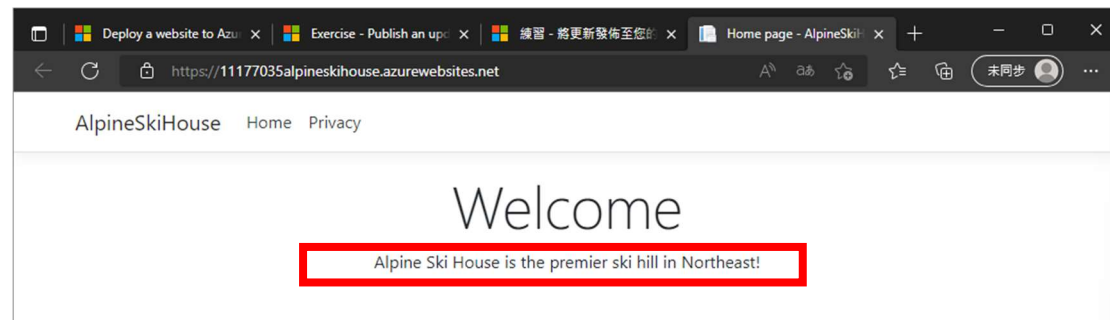
### 1. Create a ASP.NET Core App (use Visual Studio in local environment)



## 2. Publish an ASP.NET app from Visual Studio to Azure WebApp

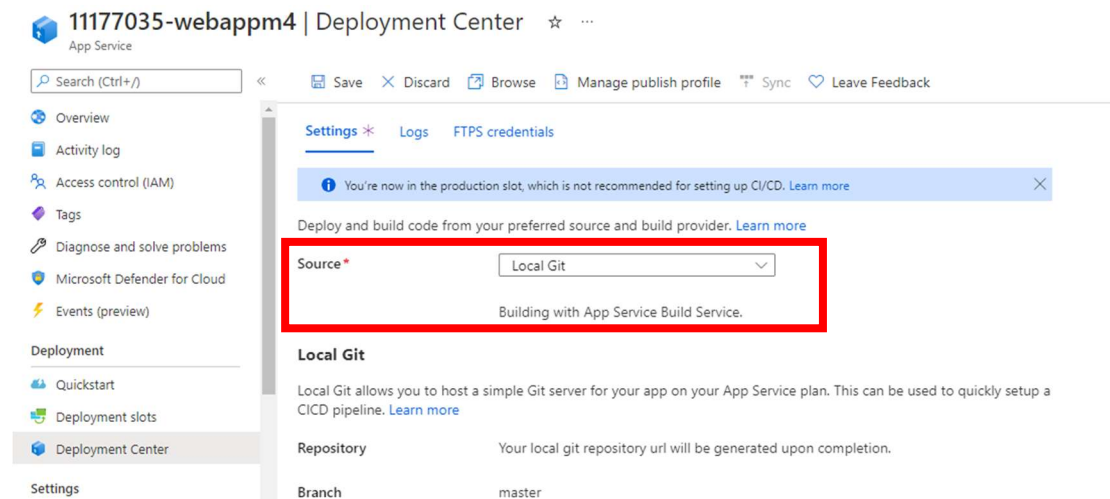


## 3. Publish an update to your site

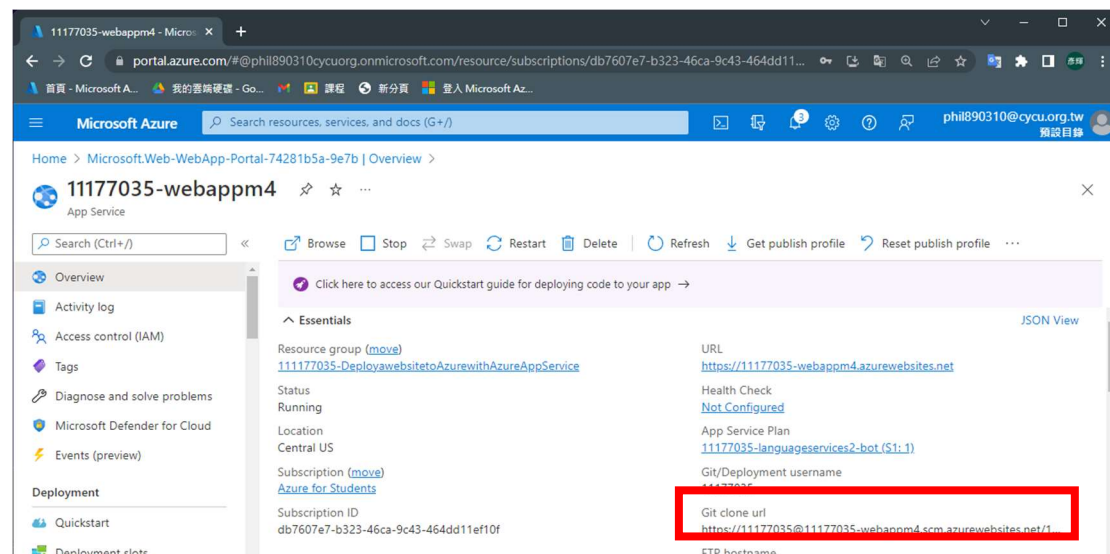


## Module 4: Stage a web app deployment for testing and rollback by using App Service deployment slots

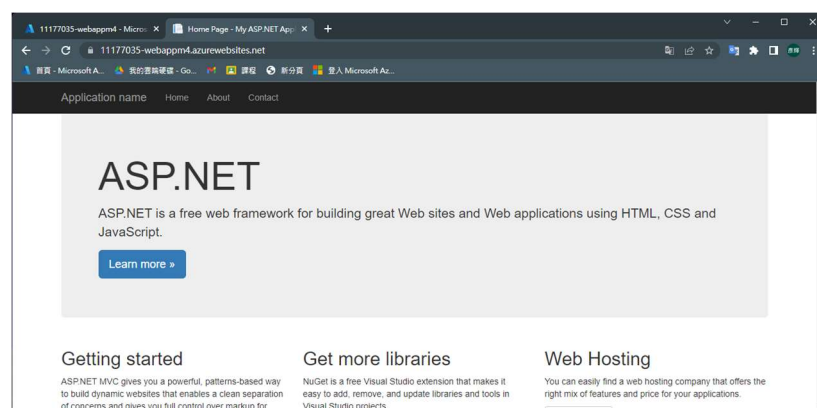
### 1. Create WebApp & Configure git deployment



### 2. Configure a git remote to deploy the app to production



### 3. Result



#### 4. Create a new staging slot

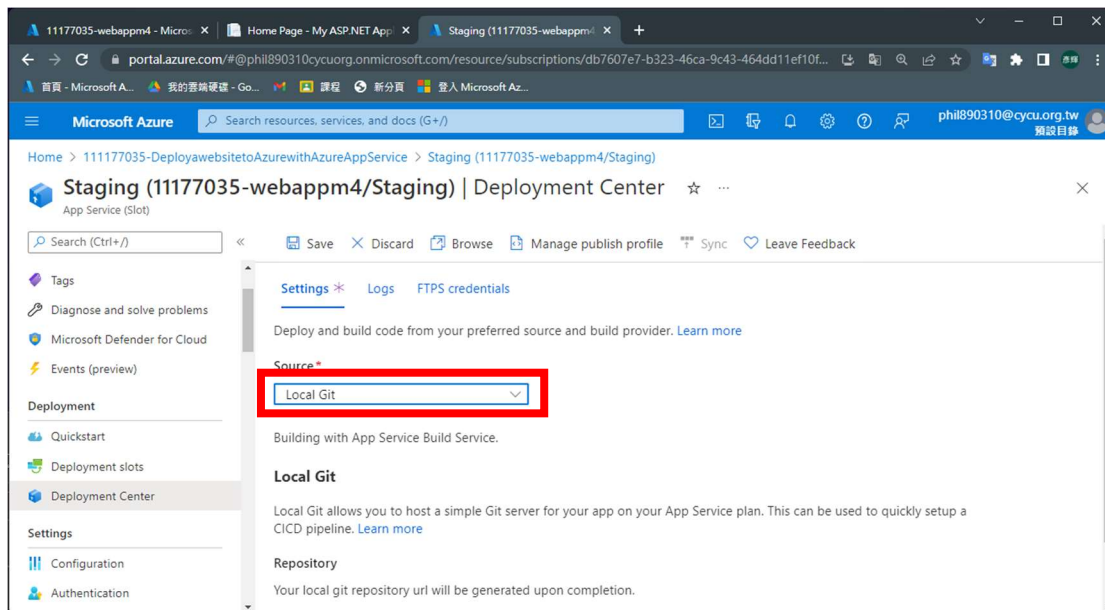
The screenshot shows the Microsoft Azure portal interface. On the left, the navigation pane is visible with the 'Deployment slots' option selected. The main content area displays the '11177035-webappm4 | Deployment slots' page. A red box highlights the 'Add Slot' button. To the right, the 'Add a slot' dialog box is open, showing the 'Name' field set to 'Staging' and the 'Clone settings from' dropdown set to 'Do not clone settings'. The dialog box also shows the '11177035-webappm4-Staging.azurewebsites.net' URL. Below the dialog box, there is a table of deployment slots.

NAME	STATUS
11177035-webappm4	Running
11177035-webappm4-PRODUCTION	Running

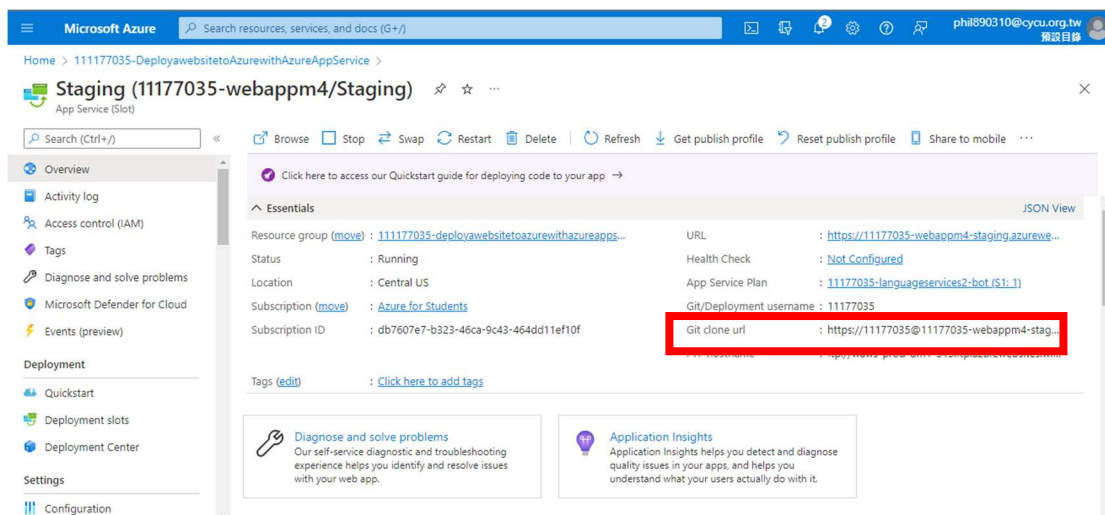
#### 5. Set up git deployment for the staging slot

The screenshot shows the Microsoft Azure portal interface. The main content area displays the '11177035-DeployawebstetoAzurewithAzureAppService' resource group. The 'Resources' tab is selected, and a table of resources is shown. A red box highlights the 'Staging (11177035-webappm4/Staging)' resource.

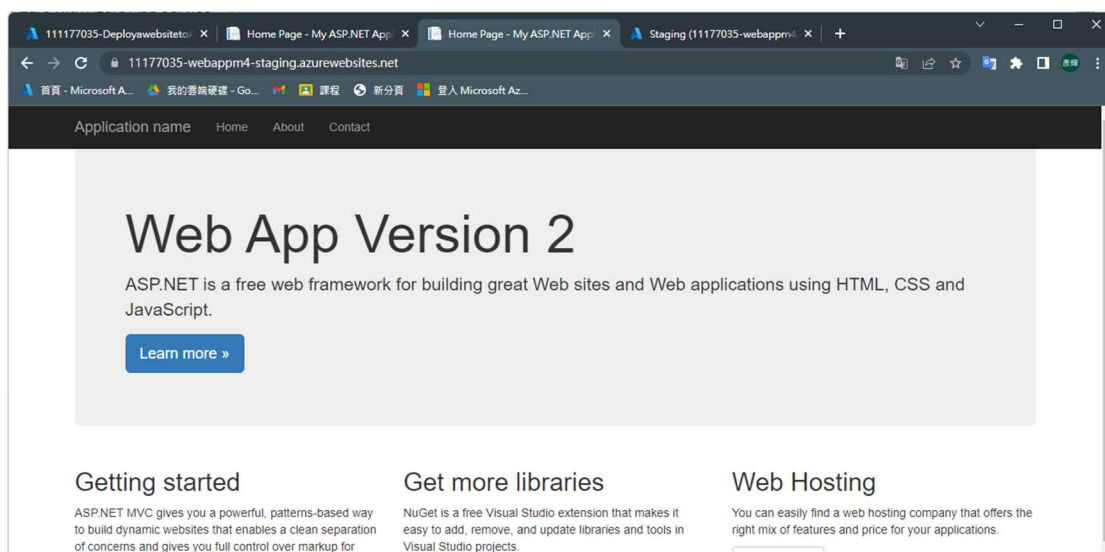
Name	Type	Location
11177035-webappm4	App Service	Central US
Staging (11177035-webappm4/Staging)	App Service (Slot)	Central US



## 6. Set up git to deploy the app to the staging slot



## 7. Browse the staging slot





## 8. Configure a slot setting

11177035-webappm4 | Configuration

Search (Ctrl+/) Refresh Save Discard Leave Feedback

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Microsoft Defender for Cloud Events (preview) Deployment Quickstart Deployment slots Deployment Center

Application settings General settings Default documents

Application settings

Application settings are encrypted at rest and transmitted over a secure connection at runtime. [Learn more](#)

+ New application setting Show values Advanced

Filter application settings

Name

Add/Edit application setting

Name ENVIRONMENT\_NAME

Value production

☒ Deployment slot setting

## 9. Swap the slots

11177035-webappm4 - Micro... Home Page - My ASP.NET App... Swap - Microsoft Azure... Home Page - My ASP.NET App... +

portal.azure.com/#@phil890310cycu.org.onmicrosoft.com/resource/subscriptions/db7607e7-b323-46ca-9c43-464dd11ef10f...

Microsoft Azure Search resources, services, and docs (G+/)

Home > 11177035-Deploya website to Azure with Azure App Service > Staging (11177035-webappm4/Staging)

Staging (11177035-webappm4/Staging) | Deployment slots

App Service (Slot)

Search (Ctrl+/) Add Slot Swap Logs Refresh

Deployment Slots

Deployment slots are live apps with their own hostnames. App content is deployed to the production slot.

NAME	STATUS
11177035-webappm4 PRODUCTION	Running
11177035-webappm4-Staging	Running

Swap

Source 11177035-webappm4-Staging

Target PRODUCTION 11177035-webappm4

☒ Perform swap with preview

Config Changes

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

SETTING	TYPE	OLD VALUE	NEW VALUE
PhpVersion	General	5.6	
APP_VERSION	AppSetting	2	1
WEBSITE_NODE_DEFAULTS	AppSetting	6.9.1	Not set

Swap Close

## 10. Version2 swap back to Version1 & swap again to new Version3

11177035-webappm4 - Micro... Home Page - My ASP.NET App... Swap - Microsoft Azure... Home Page - My ASP.NET App... +

11177035-webappm4.azurewebsites.net

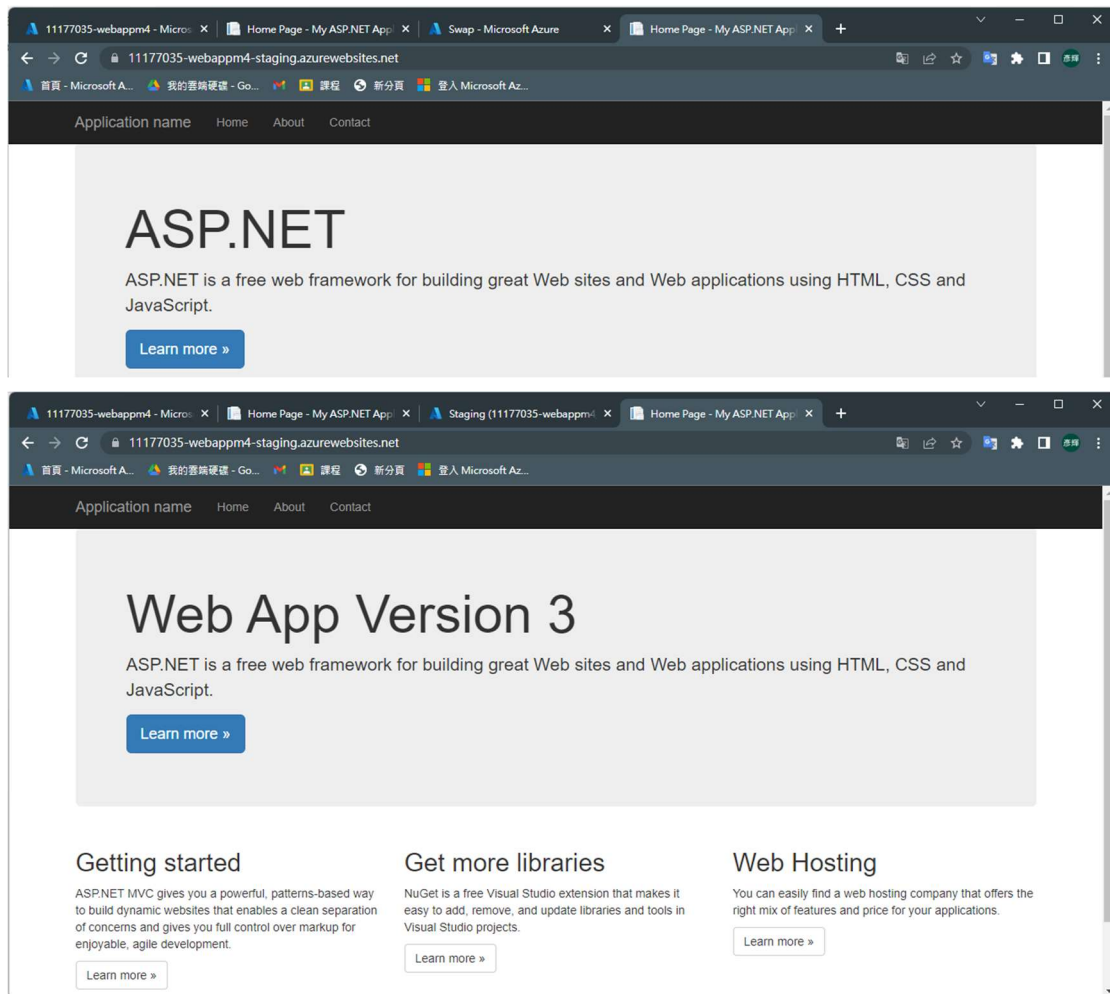
Application name Home About Contact

# Web App Version 2

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

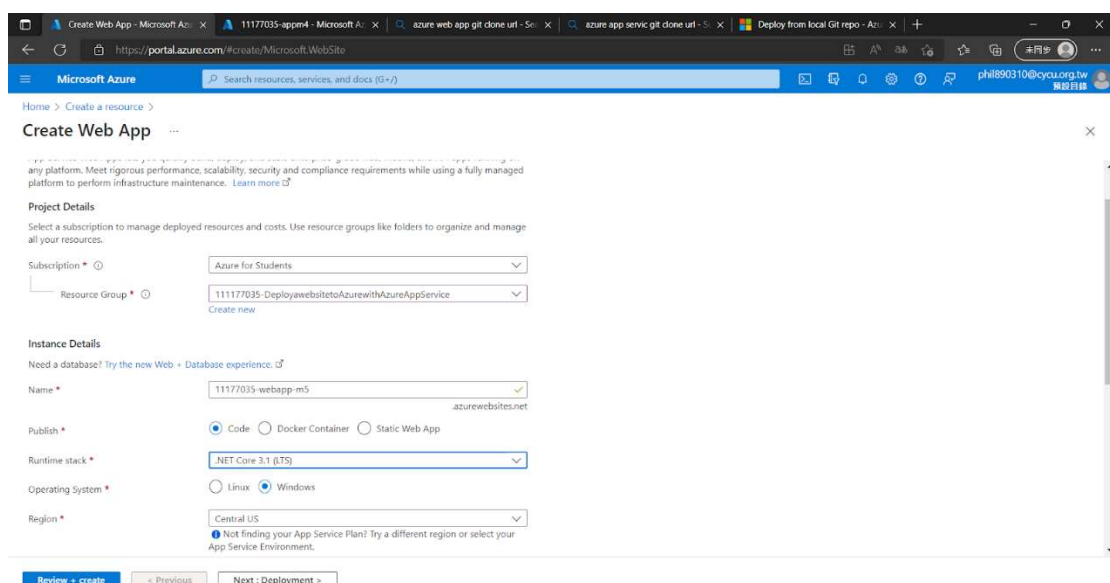
[Learn more »](#)



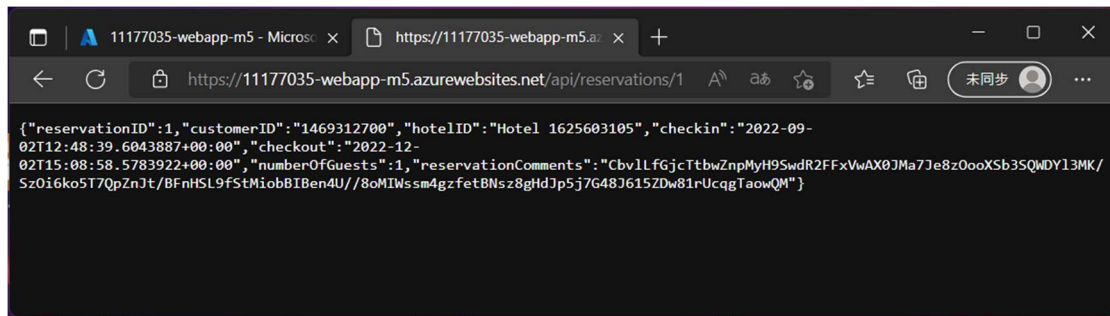


## Module 5: Scale an App Service web app to efficiently meet demand with App Service scale up and scale out

### 1. Create an Web App



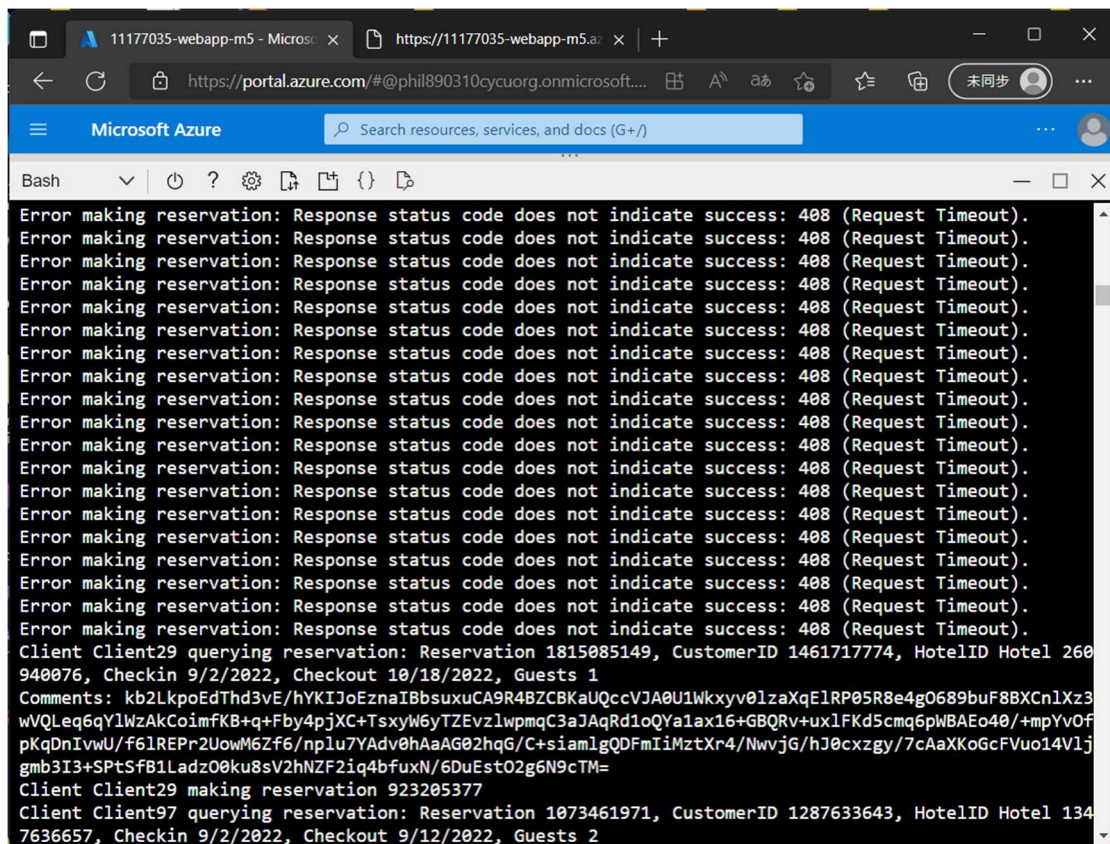
## 2. Deploy a demo project



A screenshot of a web browser window. The address bar shows the URL `https://11177035-webapp-m5.azurewebsites.net/api/reservations/1`. The page content displays a JSON object representing a reservation:

```
{ "reservationID": 1, "customerID": "1469312700", "hotelID": "Hotel 1625603105", "checkin": "2022-09-02T12:48:39.6043887+00:00", "checkout": "2022-12-02T15:08:58.5783922+00:00", "numberOfGuests": 1, "reservationComments": "CbvlLfGjcTtbwZnpMyH9SwdR2FFxVwAX0JMa7Je8zOooXSb3SQWdY13MK/Sz0i6ko5T7QpZnJt/BFnHSL9fStMiob8IBen4U//8oMIWssm4gzfetBNsz8ghDjp5j7G48J615ZDw81rUcqgTaowQM" }
```

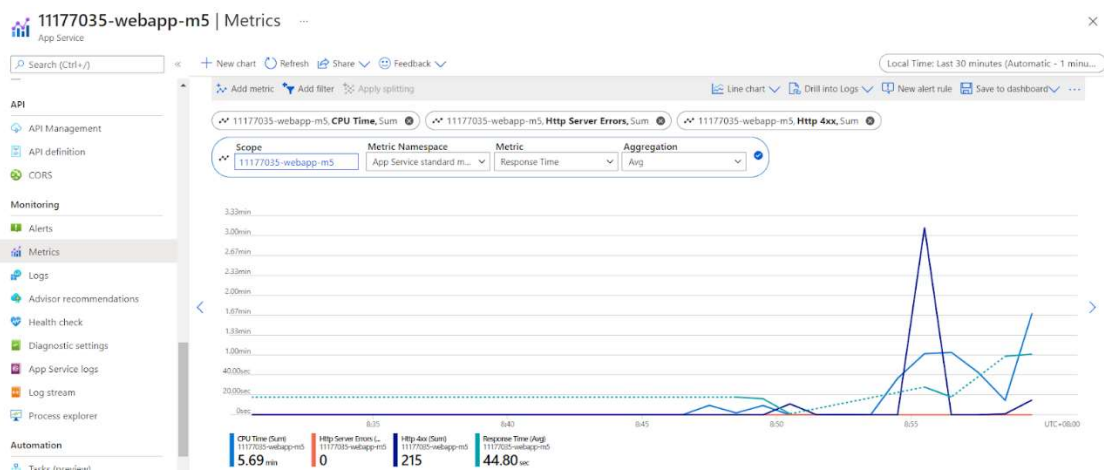
## 3. Run dotnet reservation script program (Make a large number of continuous reservation requests)



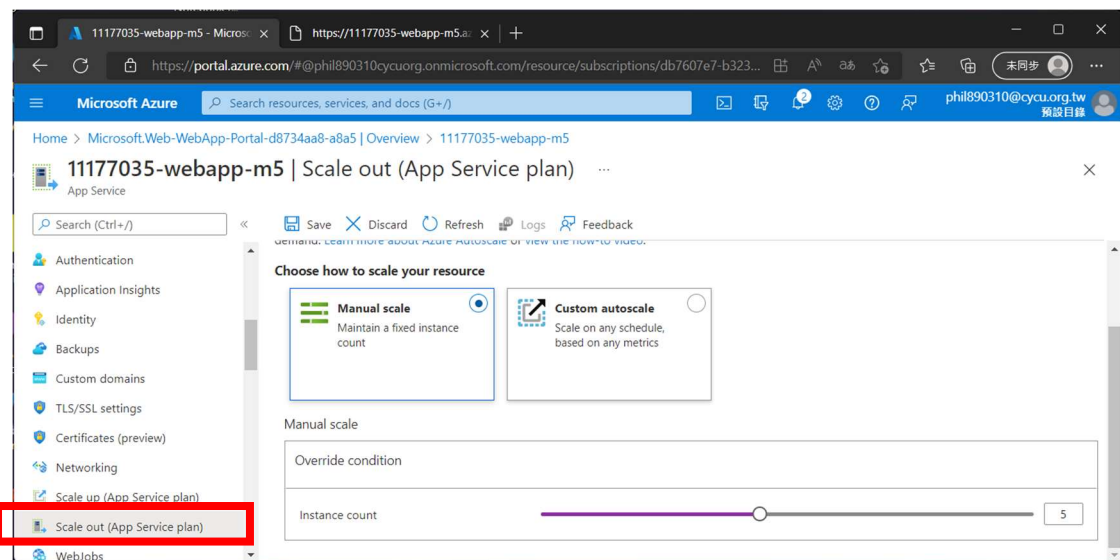
A screenshot of a terminal window running a script. The output shows a series of errors followed by reservation queries and creation attempts.

```
Bash
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Error making reservation: Response status code does not indicate success: 408 (Request Timeout).
Client Client29 querying reservation: Reservation 1815085149, CustomerID 1461717774, HotelID Hotel 260940076, Checkin 9/2/2022, Checkout 10/18/2022, Guests 1
Comments: kb2LkpoEdThd3vE/hYKIJoEznaIBbsuxuCA9R4BZCBKaUQccVJA0U1Wkxyv01zaXqElRP05R8e4g0689buF8BXCn1Xz3wVQLeq6qYlWzAkCoimfKB+q+Fby4pjXC+TsxYw6yTZEzVlwpmqC3aJAqRd1oQYalax16+GBQRv+ux1FKd5cmq6pWBAEo40/+mpYvOfpKqDnIvWU/f61REPr2UowM6Zf6/nplu7YAdv0hAaAG02hqG/C+siamlgQDFmIiMztXr4/NwvjG/hJ0cxzgy/7cAaXKoGcFVuo14Vljgmb3I3+SPTsfB1Ladz00ku8sV2hNZF2iq4bfuxN/6DuEst02g6N9cTM=
Client Client29 making reservation 923205377
Client Client97 querying reservation: Reservation 1073461971, CustomerID 1287633643, HotelID Hotel 1347636657, Checkin 9/2/2022, Checkout 9/12/2022, Guests 2
```

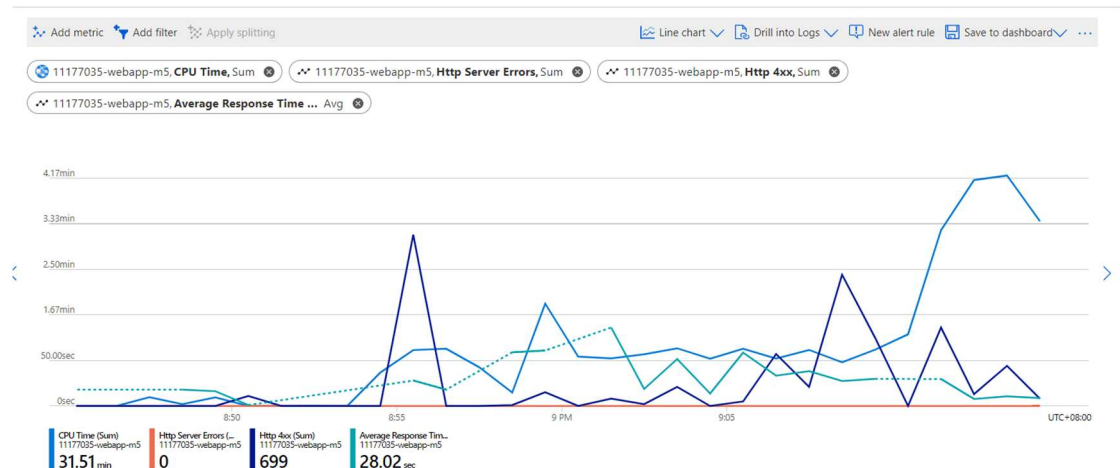
#### 4. Waiting 20 minutes, watch Webapp Monitor Metrics



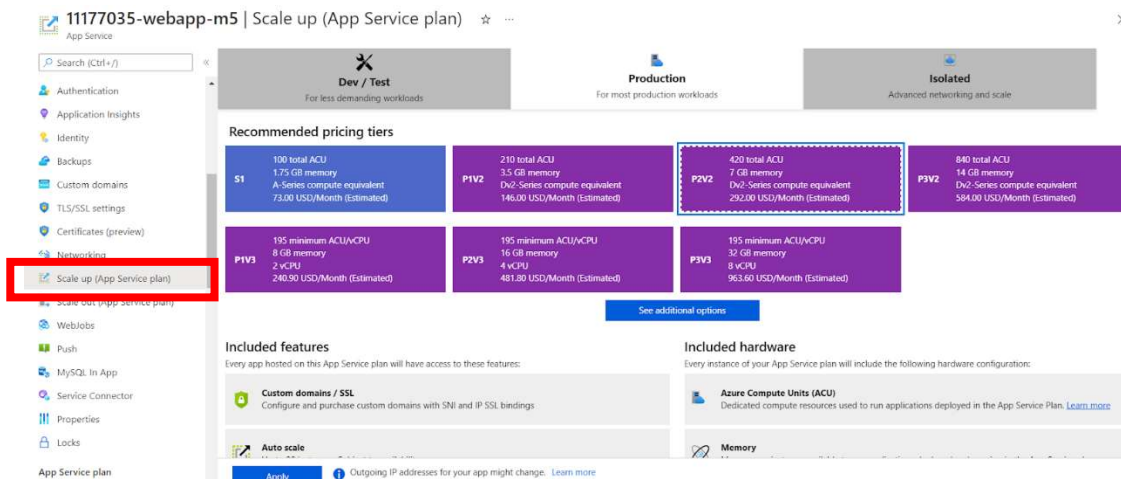
#### 5. Scale out Webapp



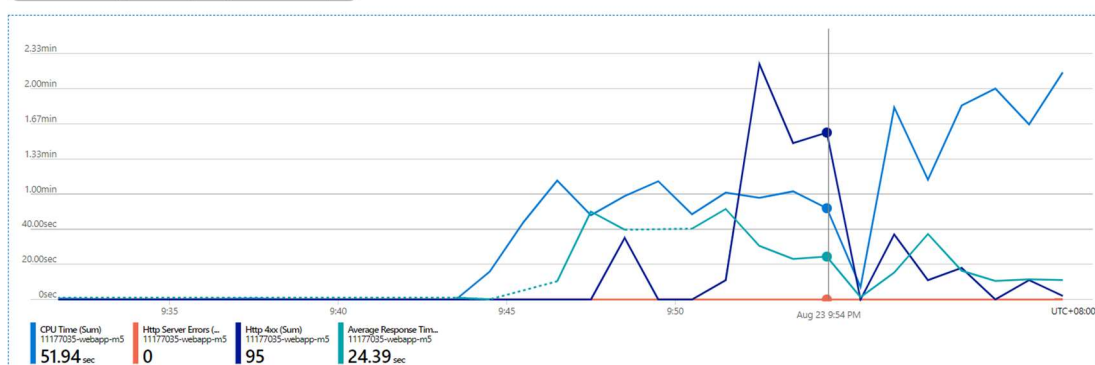
#### 6. Results



#### 7. Scale Up

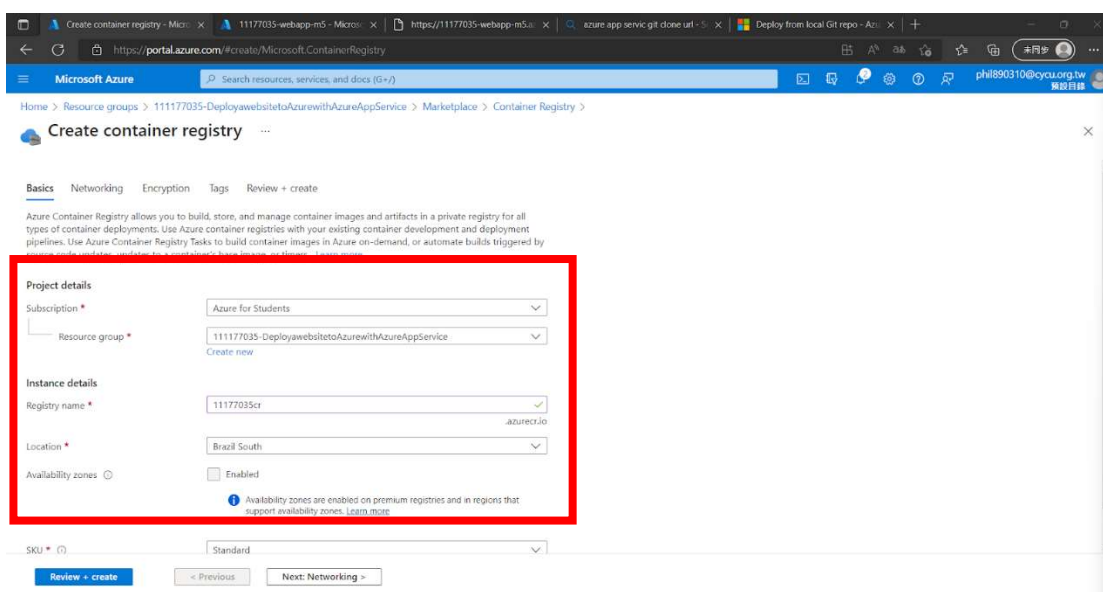


## 8. Results

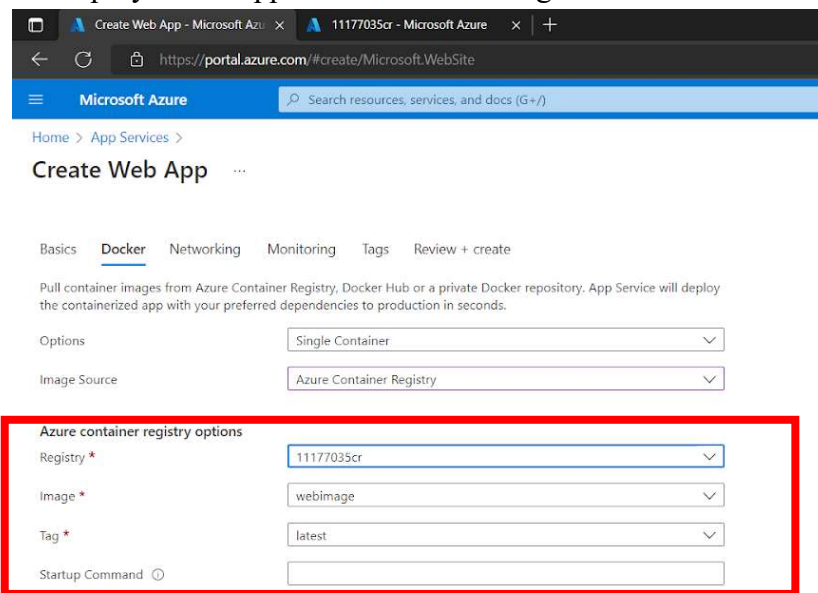


## Module 6: Deploy and run a containerized web app with Azure App Service

### 1. Build and store an image by using Azure Container Registry



## 2. Create and deploy a web app from a Docker image



The screenshot shows the 'Create Web App' page in the Microsoft Azure portal, specifically the 'Docker' tab. The page is titled 'Create Web App' and has a breadcrumb trail 'Home > App Services > Create Web App'. Below the title, there are tabs for 'Basics', 'Docker', 'Networking', 'Monitoring', 'Tags', and 'Review + create'. The 'Docker' tab is selected. The main content area contains instructions: 'Pull container images from Azure Container Registry, Docker Hub or a private Docker repository. App Service will deploy the containerized app with your preferred dependencies to production in seconds.' Below this, there are two dropdown menus: 'Options' set to 'Single Container' and 'Image Source' set to 'Azure Container Registry'. A red rectangular box highlights the 'Azure container registry options' section, which includes three dropdown menus: 'Registry' set to '11177035cr', 'Image' set to 'webimage', and 'Tag' set to 'latest'. There is also a 'Startup Command' field with a help icon.

Options: Single Container

Image Source: Azure Container Registry

**Azure container registry options**

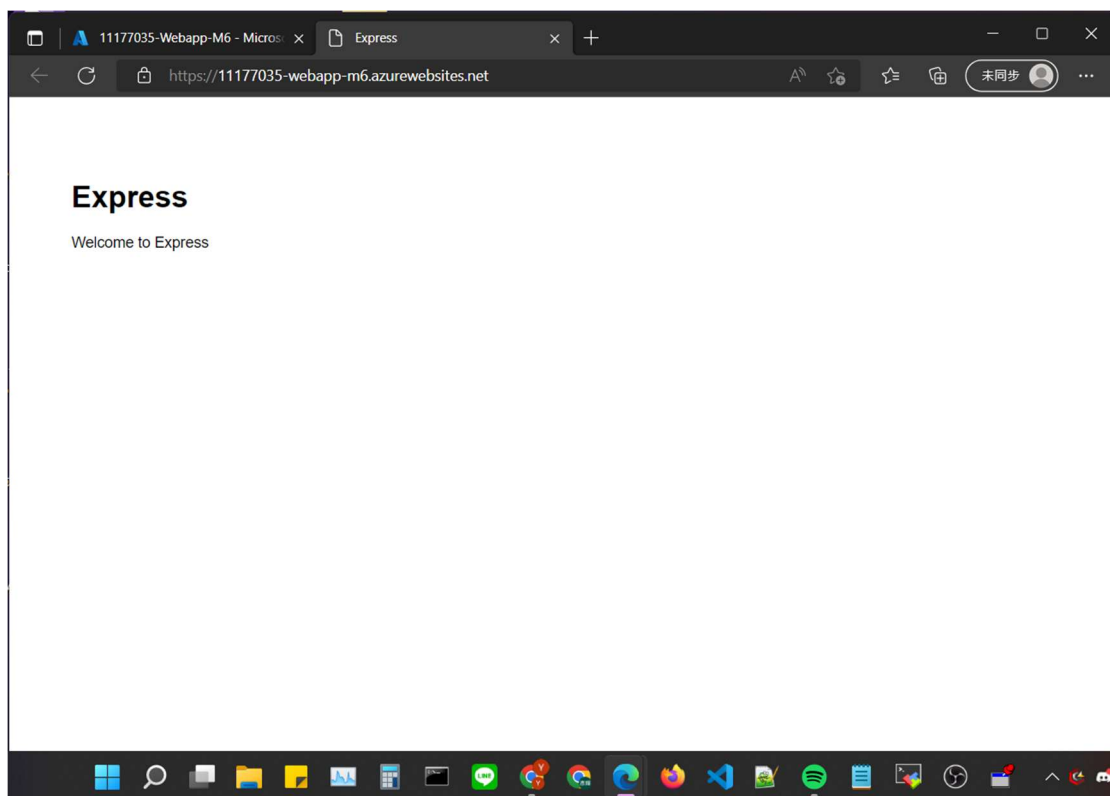
Registry \*: 11177035cr

Image \*: webimage

Tag \*: latest

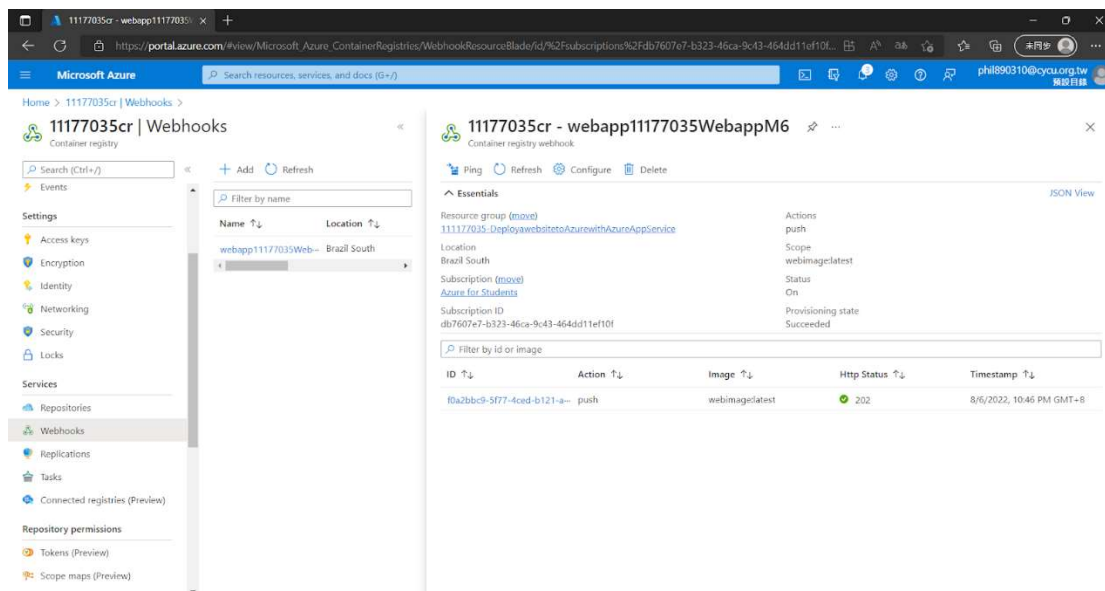
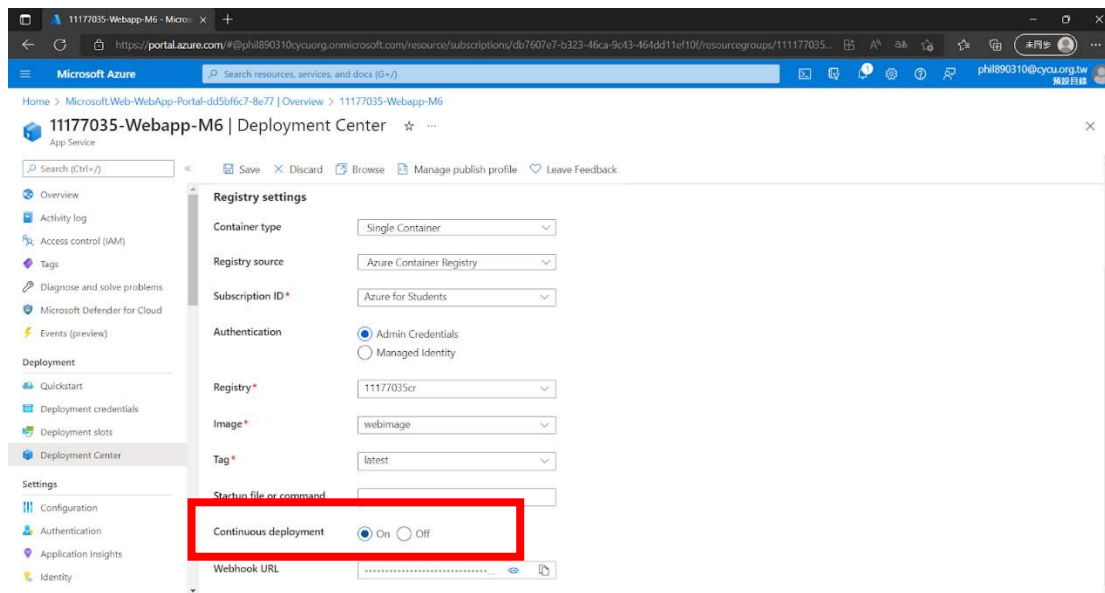
Startup Command ⓘ

## 3. Results

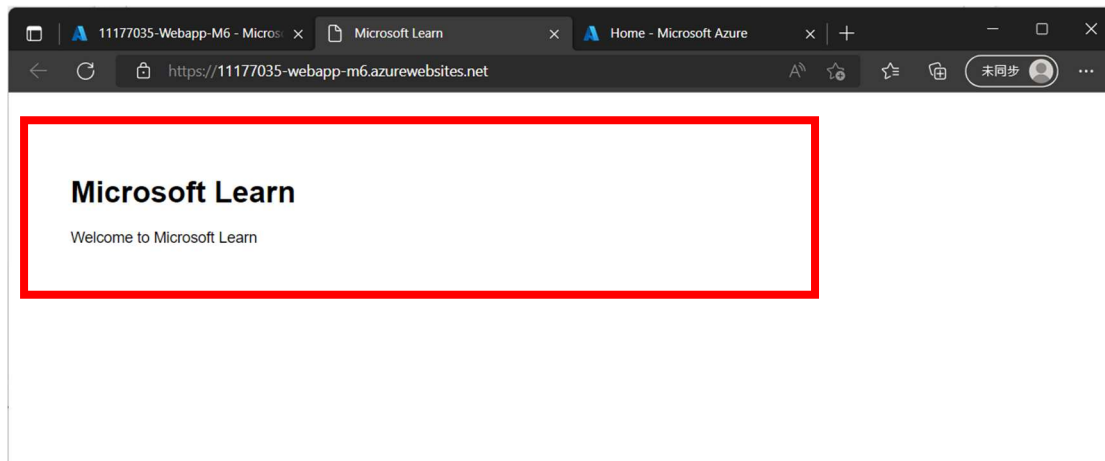




#### 4. Modify the image and redeploy the web app

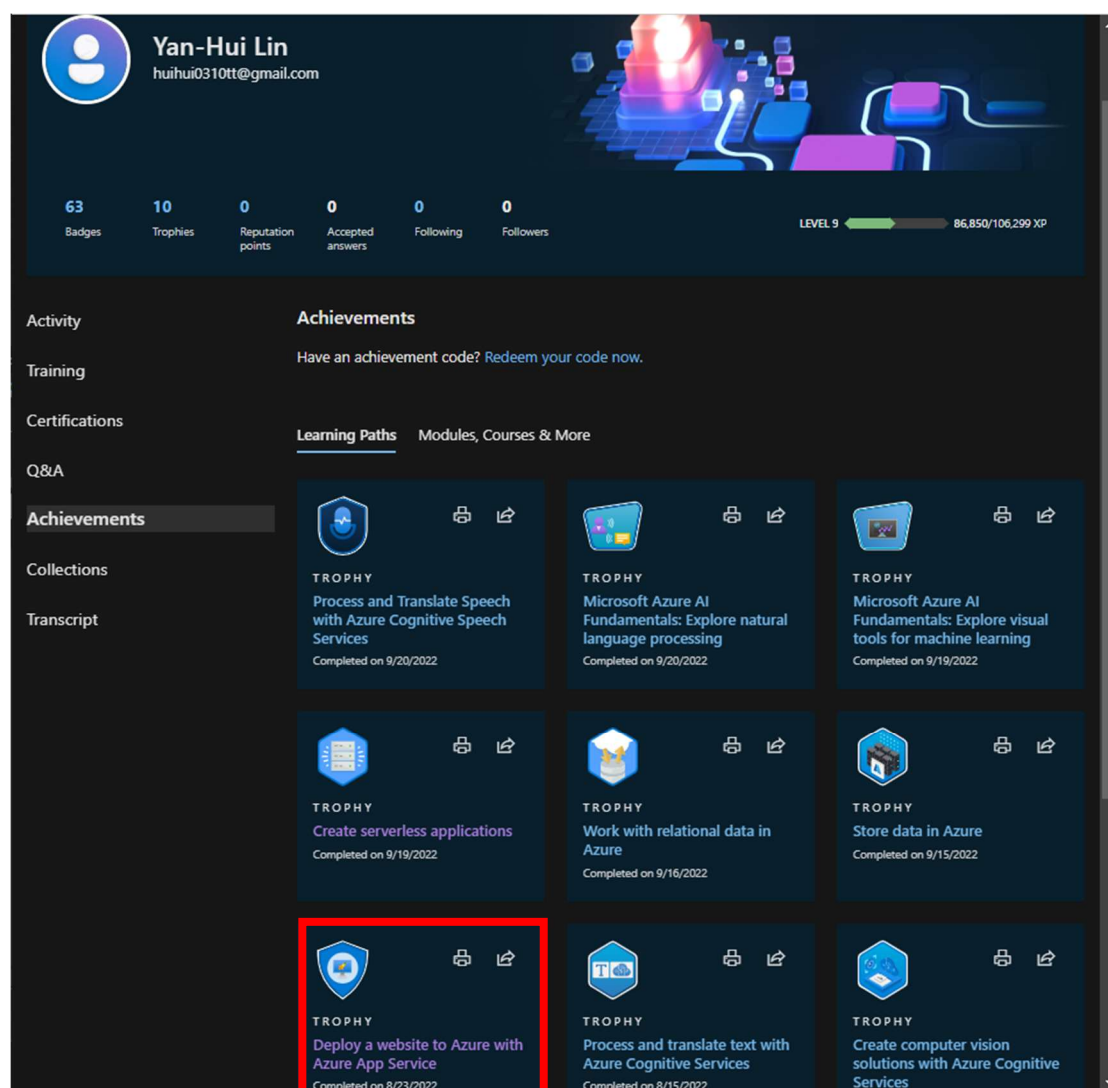


#### 5. Rebuild a image to ACR & Results





## Take screenshots of Badges and Trophies



## Learned from the Learning Path

Continuous integration & Continuous Deployment (CI/CD) 是產品開發的重要開發方式。在現在的網頁特性中，「不中斷服務」是一個基本的特性，由於網際網路的盛行，一個公開在網際網路的網頁會以 24/7 的方式供世界各地的人坐使用，尤其是購物平台。透過這個 Learning Path 讓我深刻的體悟到這個特性，搭配雲端可以自由的 Scale out & Scale up 將底層的運算設備擴充或升級，使得一個應用可以自由的使用雲端特性輕鬆地達到困難的技術要求。而 Web App 使用的 url 採用 https 協議，這使得一些安全性得到考量，因為一些套件會受限於 https 的功能，例如 getUserMedia() 開啟攝像頭的功能這使得一些前端開發可以更順利。

### 3. Problems

整體實作上無太大困難，唯獨過程中的 Web Portal 會有更新過慢的問題，但這個問題是罕見情形，過幾個小時仍舊恢復正常。

### FeedBack

相比起 Node.js，Python 框架也已經擁有一些成熟的 Framework，例如 Flask 與 Django。再其他的 Azure Learning Path Module 已經有針對這些 Framework 有了 Tutorial，希望往後可以再這個 Learning Path 中可以加入這些 Module。