



Week 5: Cloud and API deployment

Name: Darshan Dhanani

Batch code: LISUM32

Submission date: 28 Apr 2024

Submitted to:

Snapshots of Cloud deployment

Step 1: Crafting API

1. Retrieve Wine Dataset - /api/dataset

- **Method:** GET
- **Description:** Access the dataset containing wine class information utilized for model training.

2. Predict Wine Class - /api/predict-wine-class

- **Method:** POST
- **Description:** Submit data to predict the wine class using the trained model.

```
@app.route("/api/dataset", methods=['GET'])
def dataset_api():
    datas = pd.read_csv('dataset/wine_data.csv').iloc[:, 1:]
    data_dict = datas.to_dict(orient='records')
    return jsonify(data_dict)

@app.route("/api/predict-wine-class", methods=['POST'])
def predict_api():
    # convert form value into array
    features = [[(x) for x in request.form.values()]]

    # make predication with multiple model
    predicted_data = pred.process_data(features)

    response = [{'model': item[0], 'prediction class': float(item[1])} for item in predicted_data]
    return jsonify(response)
```

Step 2: Initiating API Deployment:

A dark-themed terminal window with three colored window control buttons (red, yellow, green) in the top-left corner. The terminal displays a single line of Python code: `app.run(host="0.0.0.0", port=8080)`. The code is color-coded: `app` is purple, `.run` is blue, `(` is blue, `host` is blue, `=` is blue, `"0.0.0.0"` is blue, `,` is blue, `port` is blue, `=` is blue, `8080` is red, and `)` is blue.

```
app.run(host="0.0.0.0", port=8080)
```

Step 3: Sign in to Azure Portal Dashboard

- Access the Azure Portal at portal.azure.com.
 - Enter credentials to sign in.
-

Step 4: Create Resource Group

- Navigate to the left-hand menu and select "Resource groups".
 - Click on "Create resource group".
 - Enter the name for the resource group.(In this case name is "ML_Projects")
 - Choose desired subscription. (Student Pack)
 - Select the preferred region for deployment.
 - Click "Review + create" and then "Create" to finalize.
-

Step 5: Create Web App

- From the Azure Portal dashboard, click on "Create a resource".
- Search for "Web App" and select it from the list of available options.
- Click "Create" to start the creation process.

Week 5: Cloud and API x Home - Canva x Create Web App - Mic x Week 5 - Flyer (A4) x ChatGPT x Carbon | Create and sh x

portal.azure.com/#create/Microsoft.WebSite

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

Home > Create a resource >

Create Web App

Basics Database Deployment Networking Monitoring Tags Review + create

App Service Web Apps lets you quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. Meet rigorous performance, scalability, security and compliance requirements while using a fully managed platform to perform infrastructure maintenance. [Learn more](#)

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure for Students

Resource Group * (New) Resource group [Create new](#)

Instance Details

Name * Web App name .azurewebsites.net

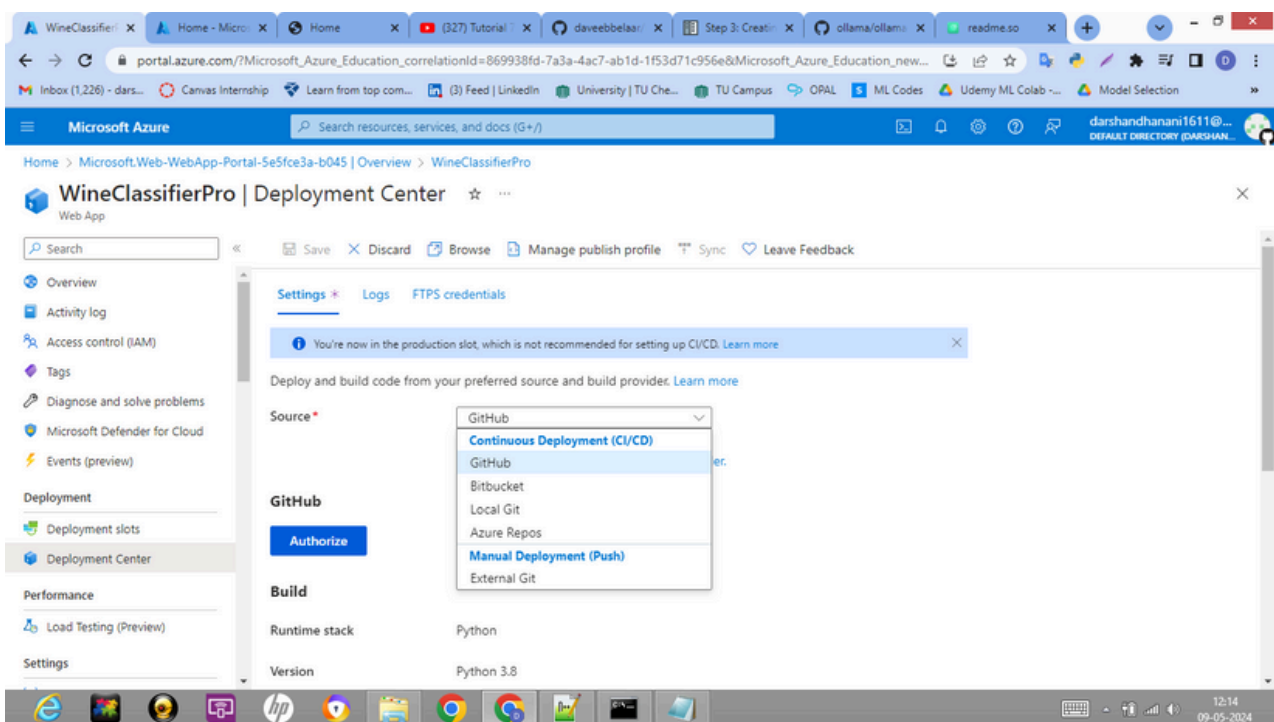
Publish * ☒ Code ☐ Container ☐ Static Web App

[Review + create](#) < Previous Next : Database >

11:34 12-05-2024

Step 6: Connect Source Code App

- After creating the web app, navigate to the "Deployment Center" tab.
- Click on "Deployment Center" from the left-hand menu.
- Choose your preferred source control option; (in this case "GitHub")
- Follow the prompts to authorize Azure to access GitHub repositories.
- Select the repository containing source code.
- Configure the deployment settings as required, such as branch to deploy from.
- Review the settings and click "Finish" or "Deploy" to connect your source code with the web app.



Step 7: Set Environment Variables

- From the Azure Portal dashboard, navigate to your web app.
- In the left-hand menu, select "Configuration" under the Settings section.
- Click on "New application setting" to add a new environment variable.
- Set the following environment variables:
- **WEBSITES_PORT**: Value set to **8080**.
- **WEBSITE_HTTPLOGGING_RETENTION_DAYS**: Value set to **15**.
- Save your changes to apply the environment variables to your web app.

The screenshot shows the Azure Portal interface for a web app named 'wine-classifier-pro'. The left-hand menu is expanded to the 'Settings' section, and 'Environment variables' is selected. The main pane displays the 'App settings' tab, which includes a search bar, '+ Add application setting', 'Refresh', 'Hide values', 'Advanced edit', and 'Fetch latest values' buttons. Below these is a table of environment variables:

Name	Value	Deployment slot setting	Source	Delete
SCM_DO_BUILD_DURING_DEPLOYMENT	1		App Service	
WEBSITE_HTTPLOGGING_RETENTION_DAYS	15		App Service	
WEBSITES_PORT	8080		App Service	

At the bottom of the table are 'Apply' and 'Discard' buttons. A 'Send us your feedback' link is also present. The browser's taskbar at the bottom shows the time as 12:05 on 12-05-2024.

Step 8: Successful Deployment

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo and a search bar. The main content area displays the 'Overview' page for a deployment named 'Microsoft.Web-WebApp-Portal-5e5fce3a-b045'. The status is 'Your deployment is complete'. The deployment details show the start time as 5/9/2024, 12:08:41 PM, and the correlation ID as ea0997ab-988b-4ea8-bc51-6a32a28524b. The deployment is associated with the 'Azure for Students' subscription and the 'ML_Projects' resource group. The 'Next steps' section includes links to 'Manage deployments for your app' and 'Protect your app with authentication'. The right sidebar contains links to 'Cost Management', 'Microsoft Defender for Cloud', 'Free Microsoft tutorials', and 'Work with an expert'. The bottom taskbar shows various application icons and the system clock indicating 12:10 on 09-05-2024.

The screenshot shows the Microsoft Azure portal interface for a web application named 'wine-classifier-pro'. The top navigation bar includes the Microsoft Azure logo and a search bar. The main content area displays the 'Overview' page for the web app. The status is 'Running'. The deployment details show the location as 'East US', the subscription as 'Azure for Students', and the subscription ID as d91324f0-186e-4bd1-9623-0c69e4566604. The default domain is 'wine-classifier-pro.azurewebsites.net'. The app service plan is 'ASP-MLProjects-bb92 (B1: 1)'. The operating system is 'Linux'. The health check is 'Not Configured'. The GitHub project is 'https://github.com/darshan-1611-dev/wine-class-ere...'. The 'Properties' section shows the name 'wine-classifier-pro', the publishing model 'Code', and the runtime stack 'Python - 3.8'. The right sidebar contains links to 'JSON View', 'Browse', 'Stop', 'Swap', 'Restart', 'Delete', 'Refresh', 'Download publish profile', 'Reset publish profile', and 'Share to mobile'. The bottom taskbar shows various application icons and the system clock indicating 12:13 on 12-05-2024.