Hosting a Flask Application on Azure Web App Services Using Windows PowerShell

This guide provides step-by-step instructions to host a Flask application on Azure Web App Services using Windows PowerShell. We will create a resource group, an app service plan, and a web app in Azure, and then deploy the Flask application to the web app.

# Prerequisites

1. An Azure Account: If you don't have an Azure account, you can create one at https://azure.microsoft.com/en-us/free/.  
2. Azure CLI: Download and install the Azure CLI from https://docs.microsoft.com/en-us/cli/azure/install-azure-cli.

# Steps

## 1. Log in to Azure CLI

Open PowerShell and log in to your Azure account using the following command:

az login

A browser window will open asking you to sign in to your Azure account. Follow the prompts to log in.

## 2. Create a Resource Group

Create a resource group in a New Zealand region (or a nearby region such as Australia East) using the following command:

az group create --name FlaskResourceGroup --location "australiaeast"

Replace `FlaskResourceGroup` with your desired resource group name.

## 3. Create an App Service Plan

Create an app service plan using the Free tier with the following command:

az appservice plan create --name FlaskAppServicePlan --resource-group FlaskResourceGroup --sku F1

Replace `FlaskAppServicePlan` with your desired app service plan name.

## 4. Create a Web App

Create a web app with the following command:

az webapp create --resource-group FlaskResourceGroup --plan FlaskAppServicePlan --name YourFlaskAppName --runtime "PYTHON|3.8"

Replace `YourFlaskAppName` with your desired web app name.

## 5. Prepare Your Project for Deployment

Ensure your project directory has the following structure:  
```  
your\_project\_directory/  
├── app.py  
├── requirements.txt  
├── runtime.txt  
├── Procfile  
└── templates/  
 └── index.html  
  
- `Procfile`:  
  
web: python app.py  
  
- `requirements.txt`: Generate this file by running the following command in your project directory:

pip freeze > requirements.txt

- `runtime.txt`:  
python-3.8.12

## 6. Initialize a Git Repository

Initialize a Git repository in your project directory:

cd your\_project\_directory

git init

Add and commit your changes:

git add .

git commit -m "Initial commit"

## 7. Deploy Your App to Azure

Set the Azure remote for your local repository and push your code to Azure:

az webapp deployment source config-local-git --name YourFlaskAppName --resource-group FlaskResourceGroup

The command above will provide you with the Git remote URL. Add this remote URL to your local repository:

git remote add azure <your-git-remote-url>

git push azure master

## 8. Open Your App in the Browser

After the deployment is complete, you can open your app with the following command:

az webapp browse --name YourFlaskAppName --resource-group FlaskResourceGroup

# Troubleshooting

If you encounter issues during deployment, you can check the logs with the following command:

az webapp log tail --name YourFlaskAppName --resource-group FlaskResourceGroup

Replace `YourFlaskAppName` with the name of your web app.