

Deploy machine Learning Model on Azure Cloud

In Azure There are two ways to Deploy the Model

1. Deploy Machine Learning model on Microsoft Azure **Docker Container Instance**
2. Deploying Model as **Web Application** using Azure **App services**.

Note: Azure App Service is a fully managed “**Platform as a Service**” (**PaaS**) used for building web apps, mobile back ends and RESTful APIs.

Step 1: Build and **train** the model.

Step 2: Create an **API of the model**. (Here we have put it in a flask API).

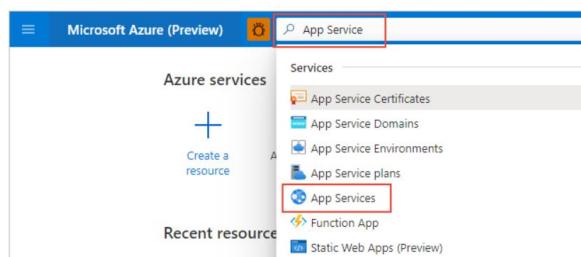
Step 3: Create the **requirements.txt** file containing all the required libraries.

Step 4: Check in Above the code in the **GitHub**

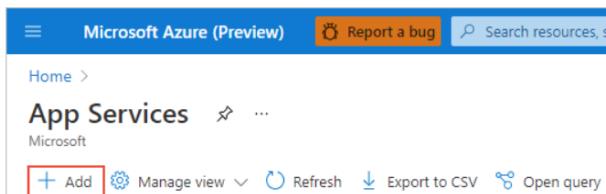
Here we are trying to deploy model as web application using azure App Service

Step 5: Setup the Azure App Service web app

1. Open the Azure portal at <https://portal.azure.com> and sign in if needed.
2. In the search bar at the top of the Azure portal, enter "App Service", then select App Services.



3. On the App Services page, select "+Add":



- On the **Create Web App** page, do the following actions

Basics [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 *	C&L Cross Service Content Team Testing
Resource Group *	(New) AppService-PythonQuickstart
Create new	

Instance Details

Name *	msdocs-pythonqs-portal
Publish *	<input checked="" type="radio"/> Code <input type="radio"/> Docker Container
Runtime stack *	Python 3.8
Operating System *	<input checked="" type="radio"/> Linux <input type="radio"/> Windows
Region *	Central US

App Service Plan

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app.

[Review + create](#) [Next : Monitoring >](#)

- At the bottom of the page, select **Review + Create**, review the details, then select **Create**.
- When provisioning is complete, select **Go to resource** to navigate to the new App Service page. Your web app at this point contains only a default page, so the next step deploys sample code.

Step 6: Deploy the sample code

- On the web app page on the Azure portal, select Deployment Center:

msdocs-pythonqs-portal

App Service

Search (Ctrl+ /)

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Security
- Events (preview)

Deployment

- Quickstart
- Deployment slots
- Deployment Center**
- Deployment Center (Classic)

Settings

Browse Stop

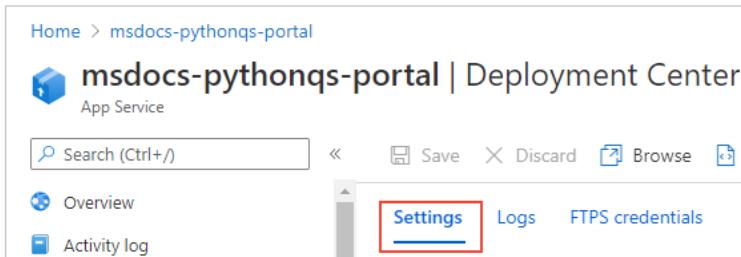
Essentials

- Resource group ([change](#))
- Status
- Location
- Subscription ([change](#))
- Subscription ID
- Tags ([change](#))

Diagnose and s
Our self-service diagr
helps you identify and

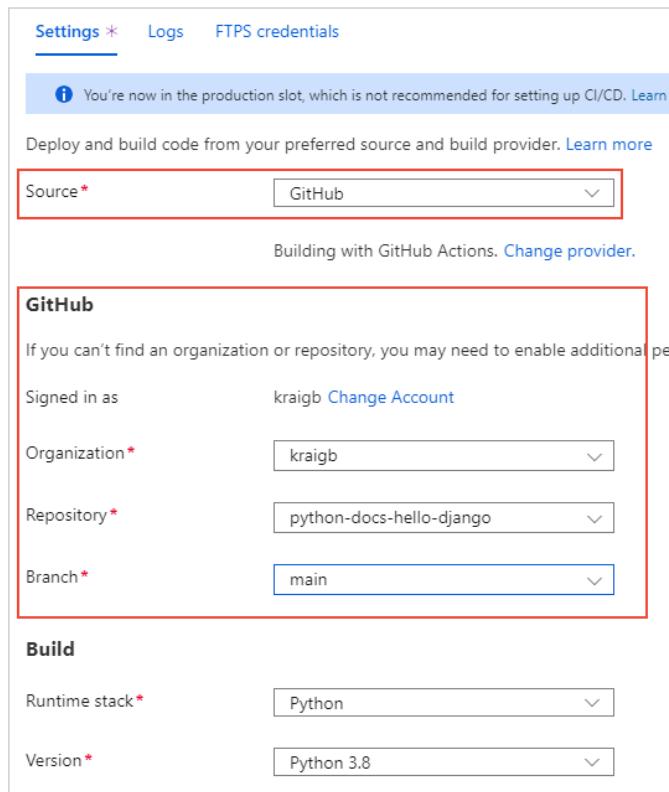
Http 5xx

2. On the Deployment Center page, select the Settings tab if it's not already open:



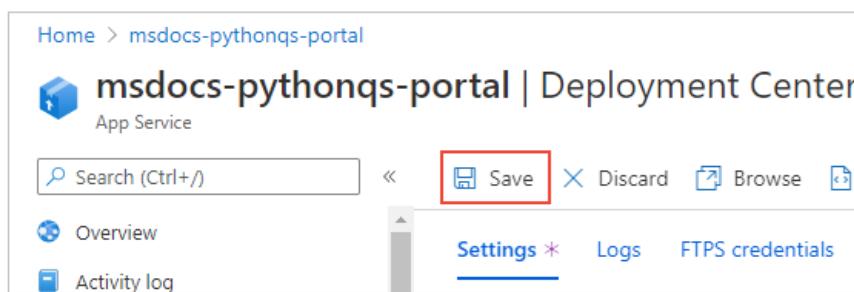
The screenshot shows the Microsoft Azure Deployment Center interface. At the top, there is a breadcrumb navigation "Home > msdocs-pythonqs-portal". Below it is the title "msdocs-pythonqs-portal | Deployment Center" with an "App Service" icon. A search bar with placeholder "Search (Ctrl+ /)" is followed by "Save", "Discard", "Browse", and a refresh icon. Below the search bar are four tabs: "Overview", "Settings" (which is highlighted with a red box), "Logs", and "FTPS credentials". Underneath these tabs, there are two more: "Activity log" and "GitHub Actions".

3. Under Source, select GitHub, then on the GitHub form that appears, do the following actions



The screenshot shows the "Settings" tab in the Deployment Center. A message at the top says "You're now in the production slot, which is not recommended for setting up CI/CD. Learn more". Below it, a note says "Deploy and build code from your preferred source and build provider. [Learn more](#)". A dropdown menu for "Source*" is set to "GitHub", which is highlighted with a red box. The "GitHub" section is also highlighted with a large red box. It contains fields for "Signed in as" (set to "kraigb"), "Organization*" (set to "kraigb"), "Repository*" (set to "python-docs-hello-django"), and "Branch*" (set to "main"). The "Build" section below has "Runtime stack*" set to "Python" and "Version*" set to "Python 3.8".

4. At the top of the page, select Save to apply the settings.



The screenshot shows the Microsoft Azure Deployment Center settings page again. The "Save" button at the top is highlighted with a red box. The rest of the interface is identical to the previous screenshot, showing the "Settings" tab selected and the GitHub configuration details.

- Select the Logs tab to view the status of the deployment. It takes a few minutes to build and deploy the sample and additional logs appear during the process.

The screenshot shows the Azure Deployment Center interface for the app service 'msdocs-pythonqs-portal'. The 'Logs' tab is selected. A single log entry is displayed in a table:

Time	Commit ID	Commit Author	Status	Message
Monday, March 15, 2021(1) 12:03:35 PM -07:00	3e7d069	N/A	Success (Active)	{"type": "deployment"}

Step 7: Browse to the app

- Once deployment is complete, select Overview on the left-hand menu to return to the main page for the web app.
- Select the URL that contains address of the web app

The screenshot shows the Azure App Service Overview page for 'msdocs-pythonqs-portal'. The 'Overview' section displays the app's URL:

URL: <https://msdocs-pythonqs-portal.azurewebsites.net>

- Verify that the output of the app is "Hello, World!"

The screenshot shows a browser window displaying the output of the deployed application. The URL in the address bar is:

<https://msdocs-pythonqs-portal.azurewebsites.net>

The page content is:

Hello, World!