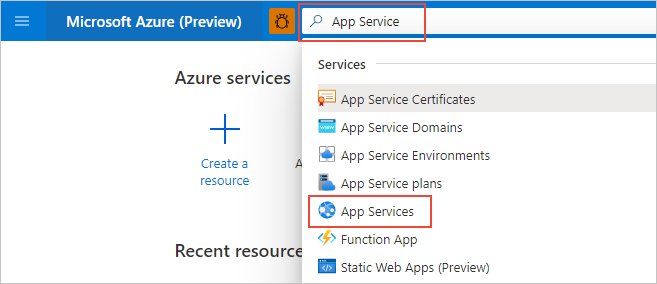
**Setup Patient Browser web application in local machine**

**Steps** 🡪

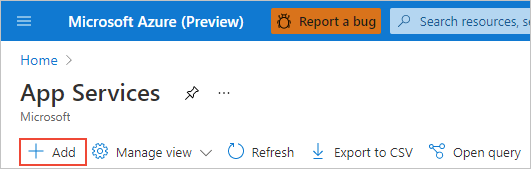
1. Install Python 3.9+ in your machine.
2. Install PyCharm IDE for development and execution.
3. Download the project code from [link](https://bitbucket.org/ssmeltz/interop-platform/src/f6b0b5e28db1d8e7695e803139bce18dcadeaef9/fhir-patient-browser/?at=feature%2FInterop-module)
4. Go to project path e.g.- D:\Projects\HPP\FHIR\Workspace\interop-platform\fhir-patient-browser
5. Download the required python libraries by executing command - pip freeze > requirements.txt
6. Configure the Azure FHIR server credentials in file (D:\Projects\HPP\FHIR\Workspace\fhir-patient-browser\app) ‘credentials.json’
7. Start the server by executing command **python manage.py runserver @** **\fhir-patient-browser**

**Deploy Application in azure**

1. Open the Azure portal at [https://portal.azure.com](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**.

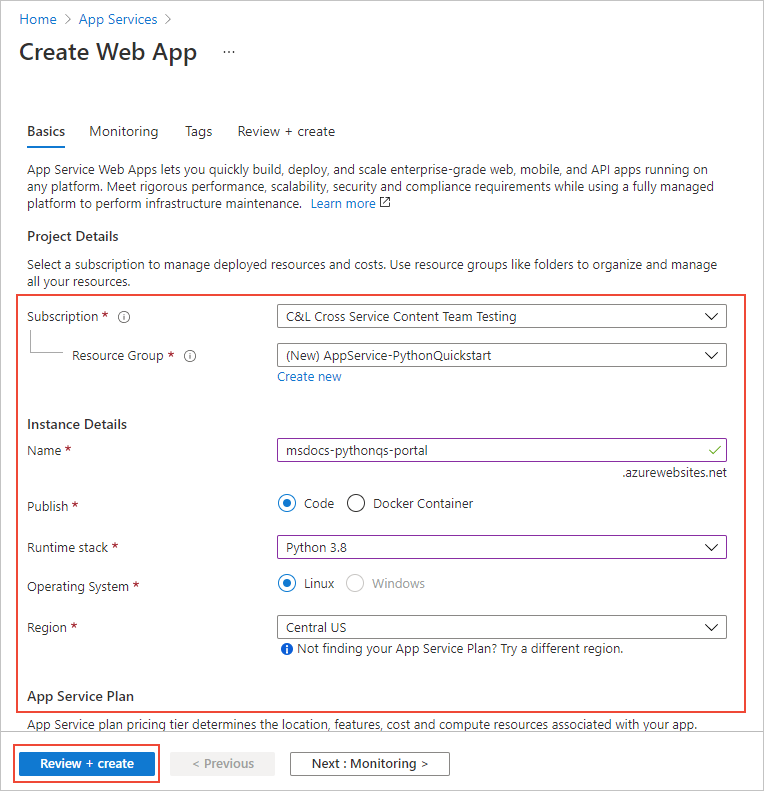


1. On the **App Services** page, select "**+Add**:



1. On the **Create Web App** page, do the following actions:

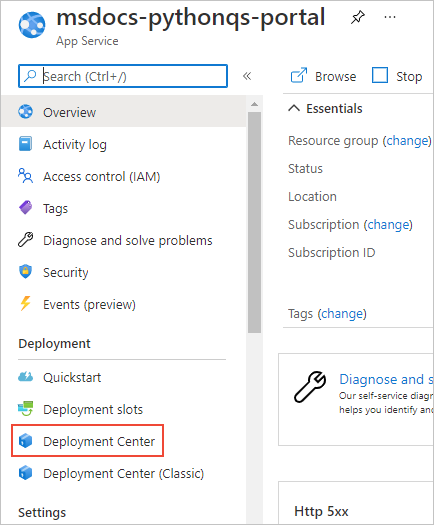
| **TABLE 1** | |
| --- | --- |
| **Field** | **Action** |
| Subscription | Select the Azure subscription you want to use. |
| Resource Group | Select **Create New** below the drop-down. In the popup, enter "AppService-PythonQuickstart" and select "**OK**. |
| Name | Enter a name that's unique across all of Azure, typically using a combination of your personal or company names, such as contoso-testapp-123. |
| Publish | Select **Code**. |
| Runtime stack | Select **Python 3.8**. |
| Operating System | Select **Linux** (Python is supported only on Linux). |
| Region | Select a region near you. |
| Linux Plan | Select an exiting App Service Plan or use **Create new** to create a new one. We recommend using the **Basic B1** plan. |



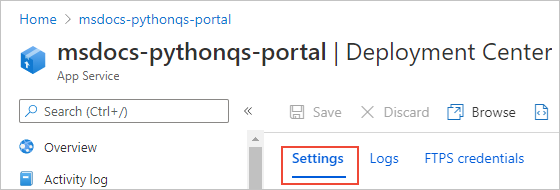
1. At the bottom of the page, select **Review + Create**, review the details, then select **Create**.
2. When provisioning is complete, select **Go to resource** to navigate to the new App Service page. Your web app at this point contains only a default page, so the next step deploys sample code.

**Deploy the sample code**

1. On the web app page on the Azure portal, select **Deployment Center**:

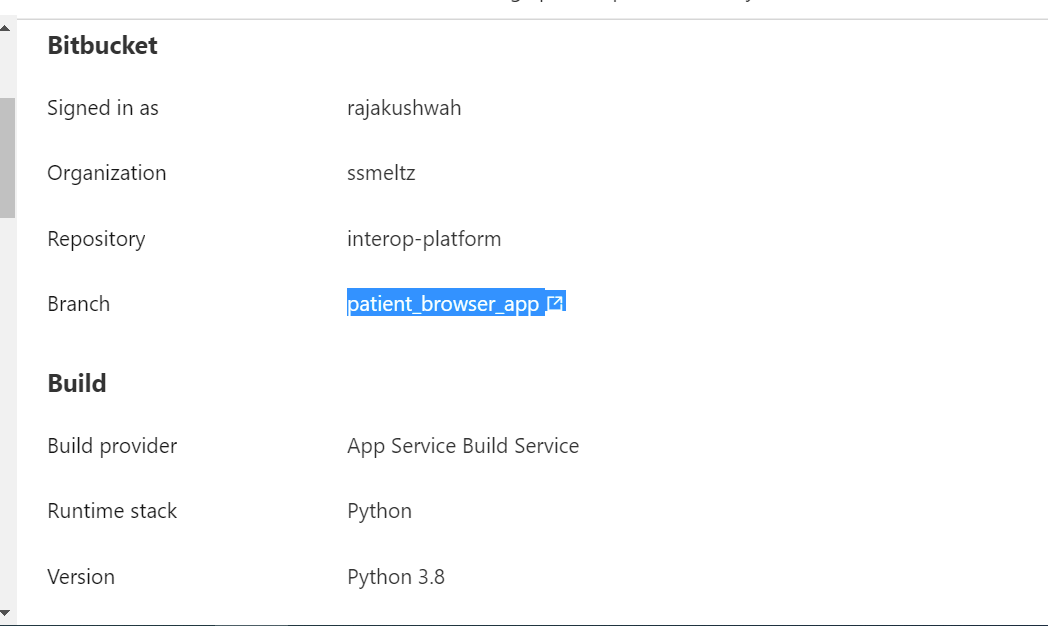


1. On the **Deployment Center** page, select the **Settings** tab if it's not already open:

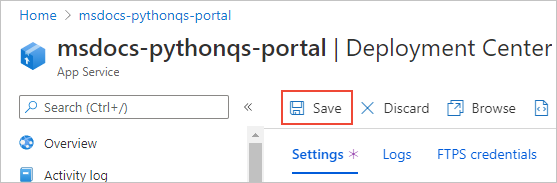


1. Under **Source**, select **Bitbuket**, then on the **Bitbucket** form that appears, do the following actions:

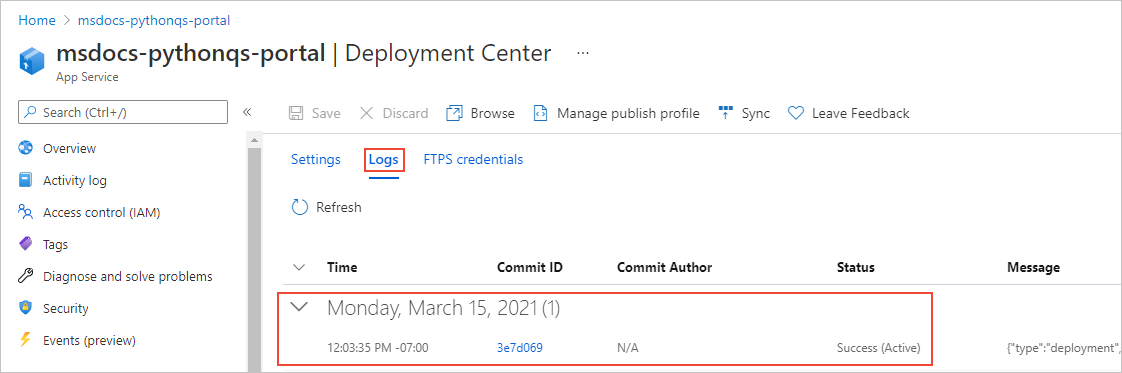
| **TABLE 2** | |
| --- | --- |
| **Field** | **Action** |
| Signed in as | If you're not signed into **Bitbuket** already, sign in now or select \**Change Account* if needed. |
| Organization | Select your **Bitbuket** organization, if needed. |
| Repository | Select the name of the sample repository as **interop-platform** |
| Branch | Select **patient\_browser\_app**. |



1. Select the runtime stack, version and save the changes

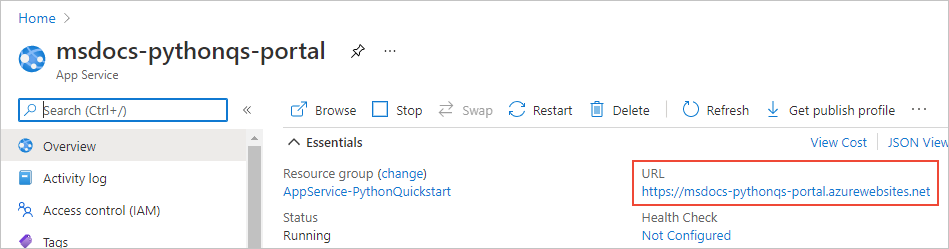


1. Select the **Logs** tab to view the status of the deployment. It takes a few minutes to build and deploy the sample and additional logs appear during the process. Upon completion, the logs should reflect a Status of **Success (Active)**:



## Browse to the app

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



1. Verify that the output of the app is “ZS Patient Browser":

