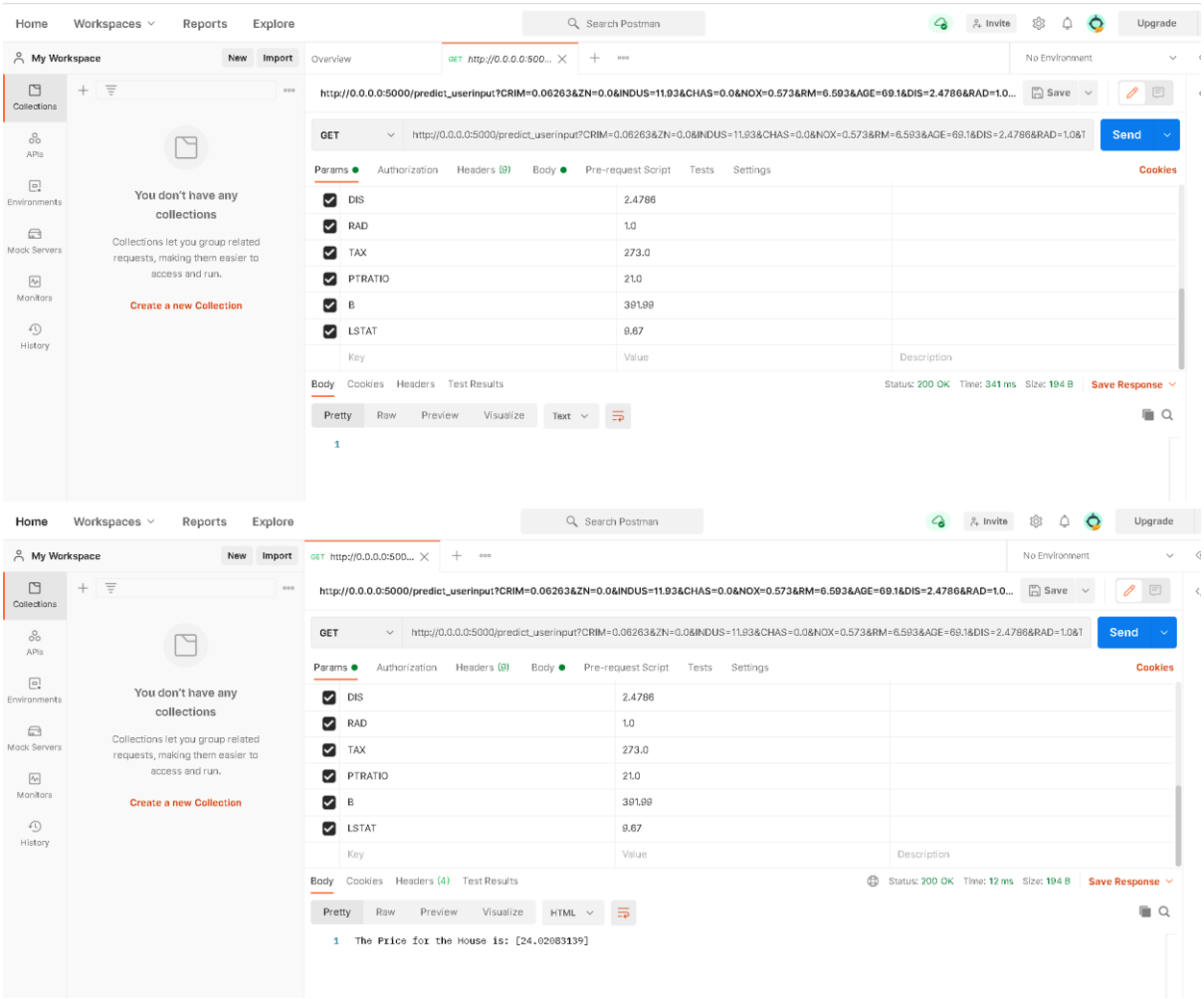
Building ML model, Creating an API for the model using Flask, Dockerizing Flask Web App, and Deploying on Azure Cloud

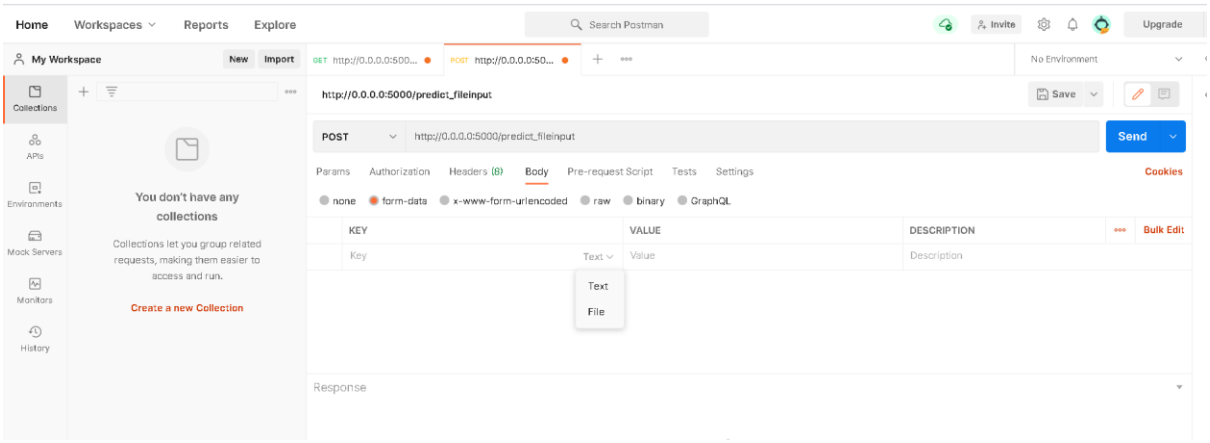
- Priyanka Naikade

Steps:

1. Trained and developed a machine learning model (Multiple Linear Regression Model) for predicting the price of a house. The model is saved as a pickle file.
2. Created a web application using Flask (a lightweight web framework for python that can be used to build and host an API). The web application deserializes the 'model.pkl' file and loads the model for making predictions on the input data.
3. Tested the API using POSTMAN.

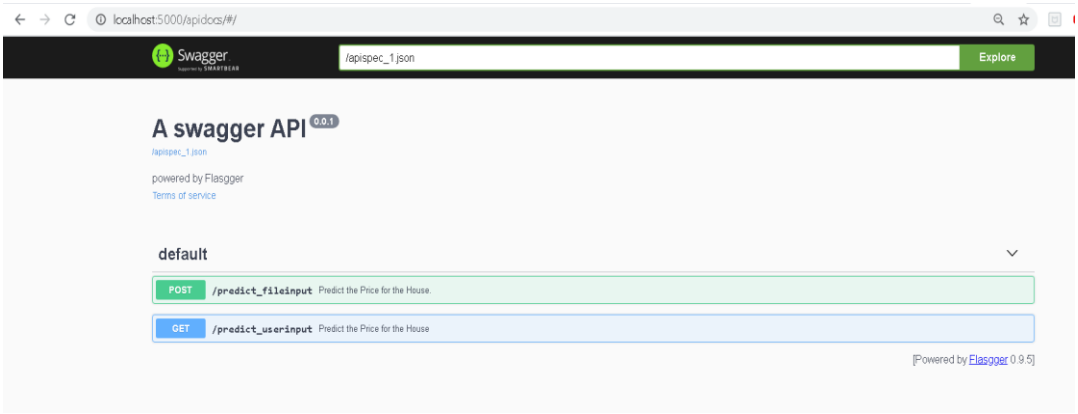


Testing for file input:
Select 'POST' request, Under Body ->form-data, Select key as 'File' from the drop-down.

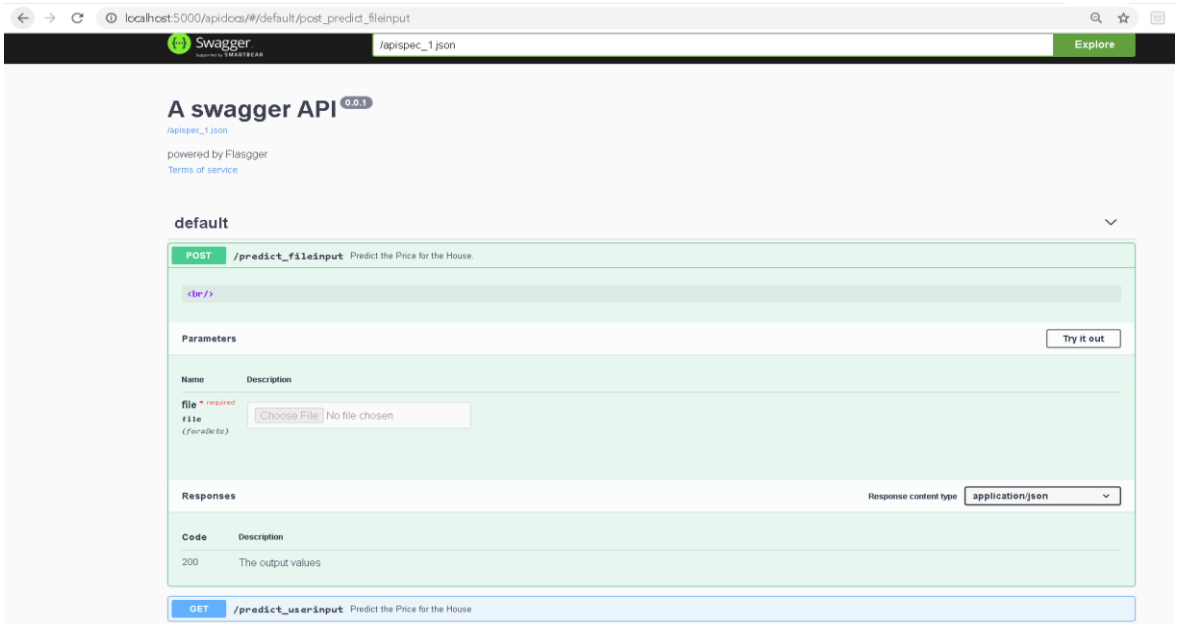


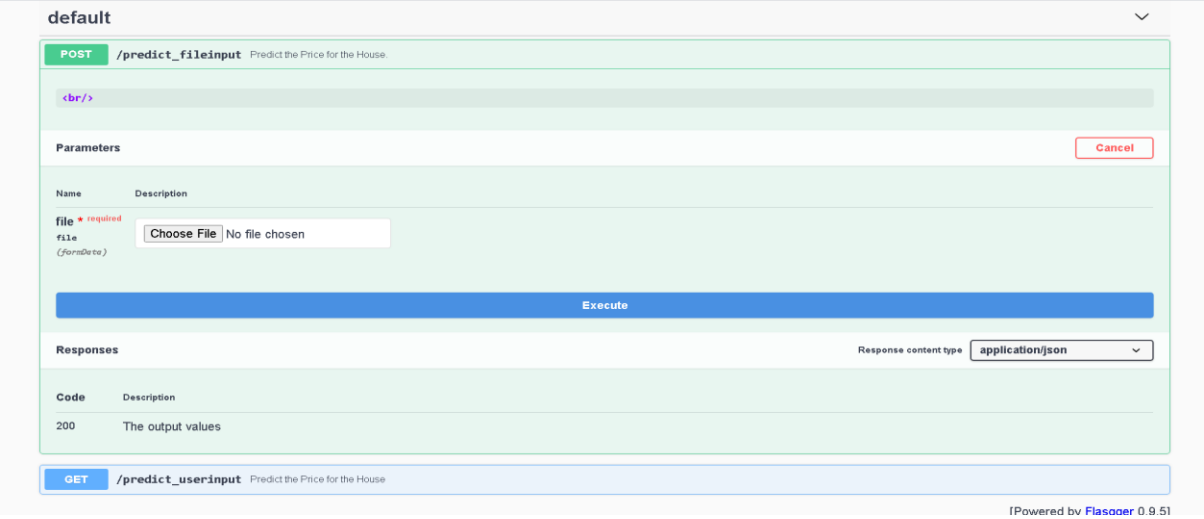
4. Created a front-end UI for the Web application using Flasgger's Swagger module. Model Predictions were made for single user input (a single record of input features) and file input.

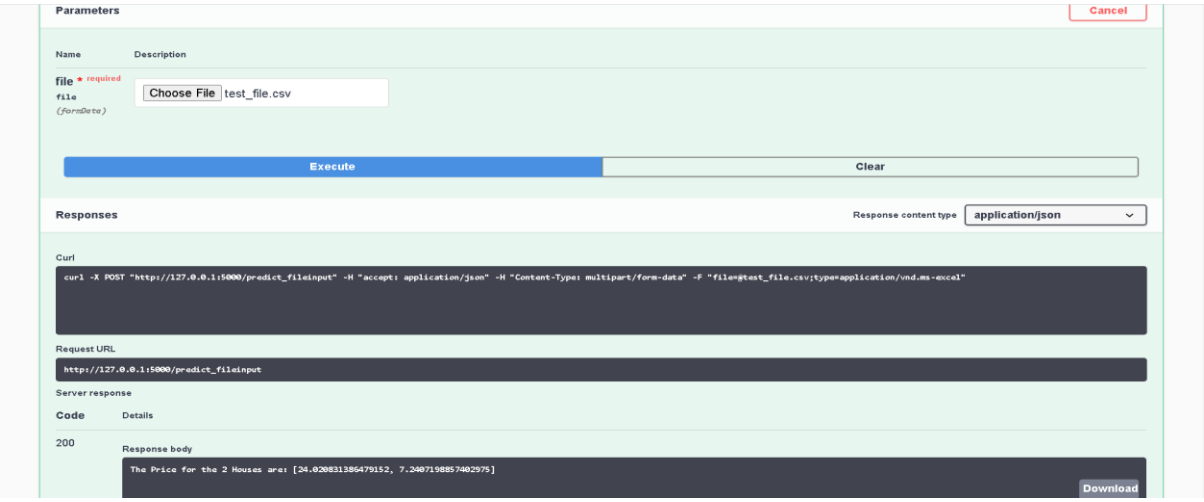
<u>Testing Web application via a local server (http://localhost:5000/apidocs)</u>



Prediction for input file data: Click on 'Try it out' button to enable the 'Choose file' option.

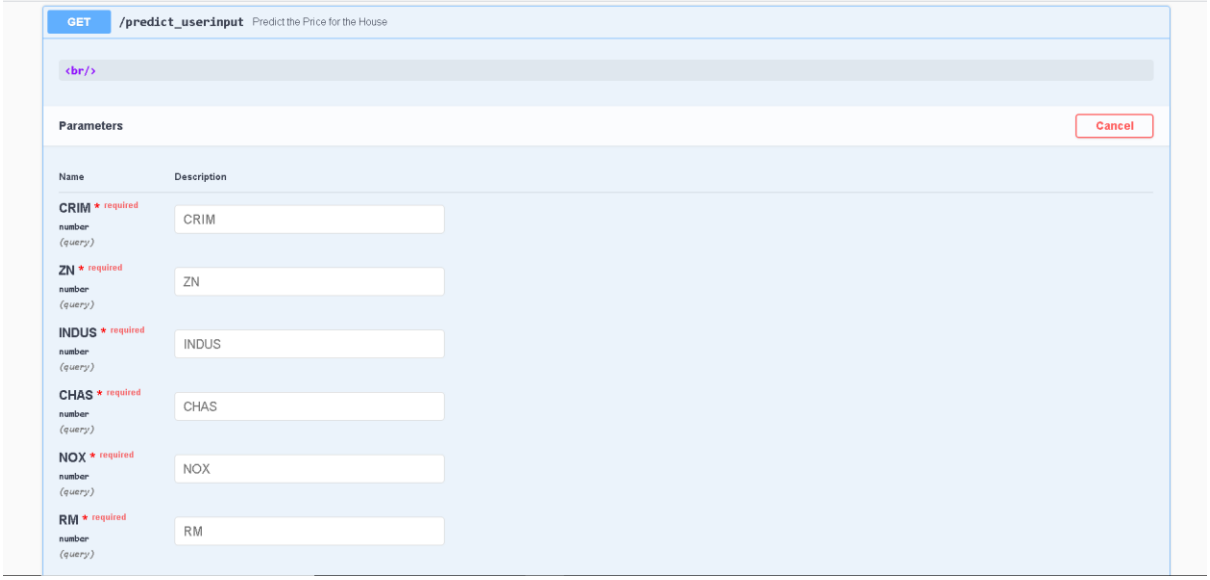


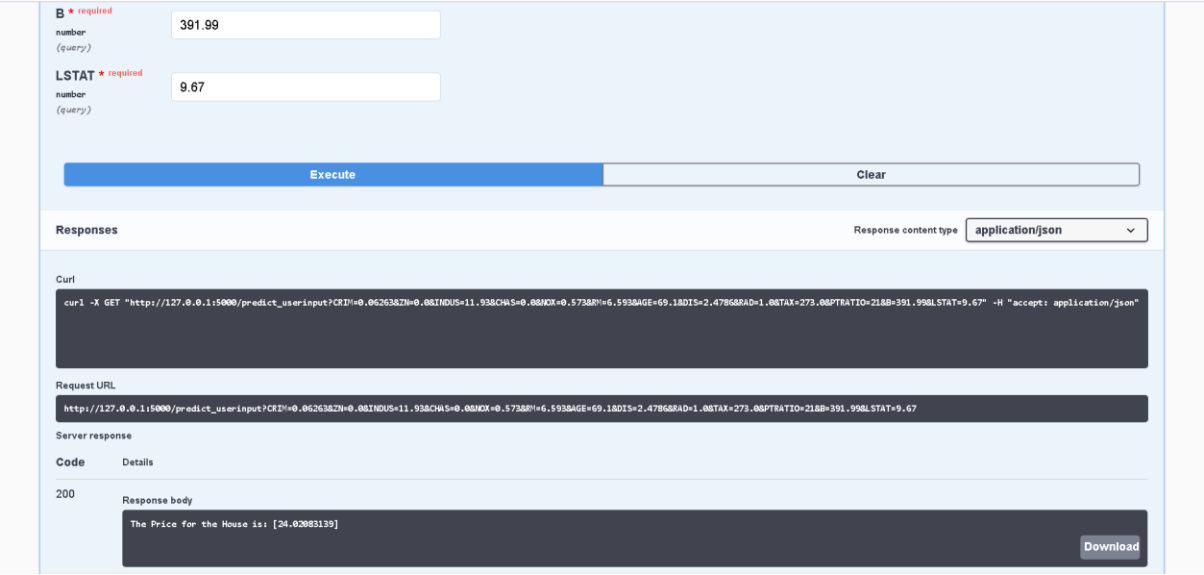


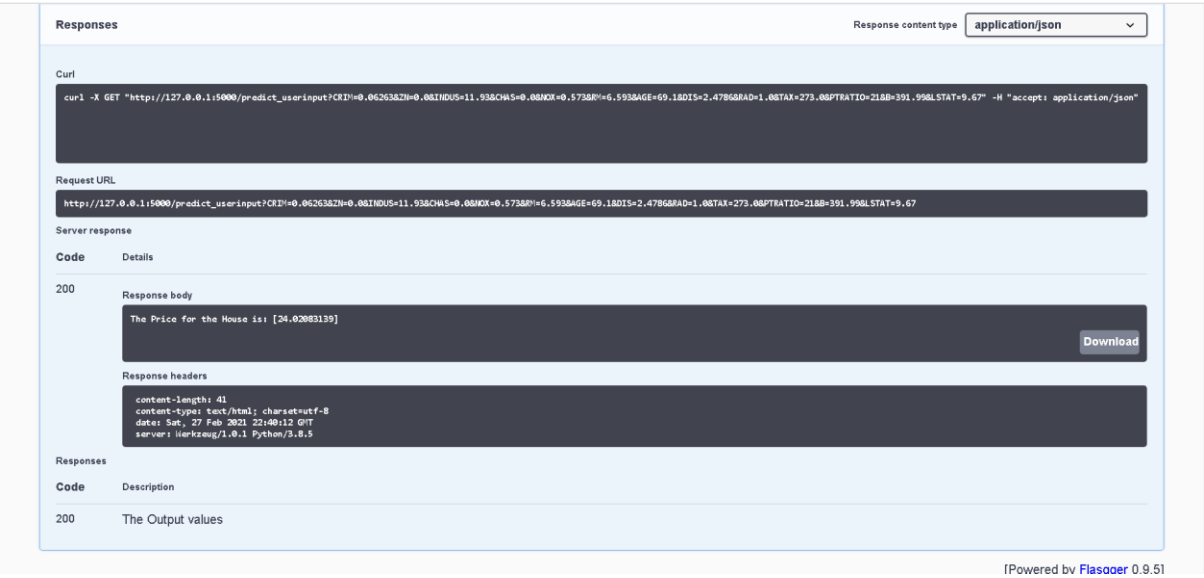


Prediction for a single record-

Enter Input Value:
CRIM=0.06263&ZN=0.0&INDUS=11.93&CHAS=0.0&NOX=0.573&RM=6.593&AGE=69.1&DIS=2.4786&RAD=1.0&TAX=273.0&PTRATIO=21.0&B=391.99&LSTAT=9.67





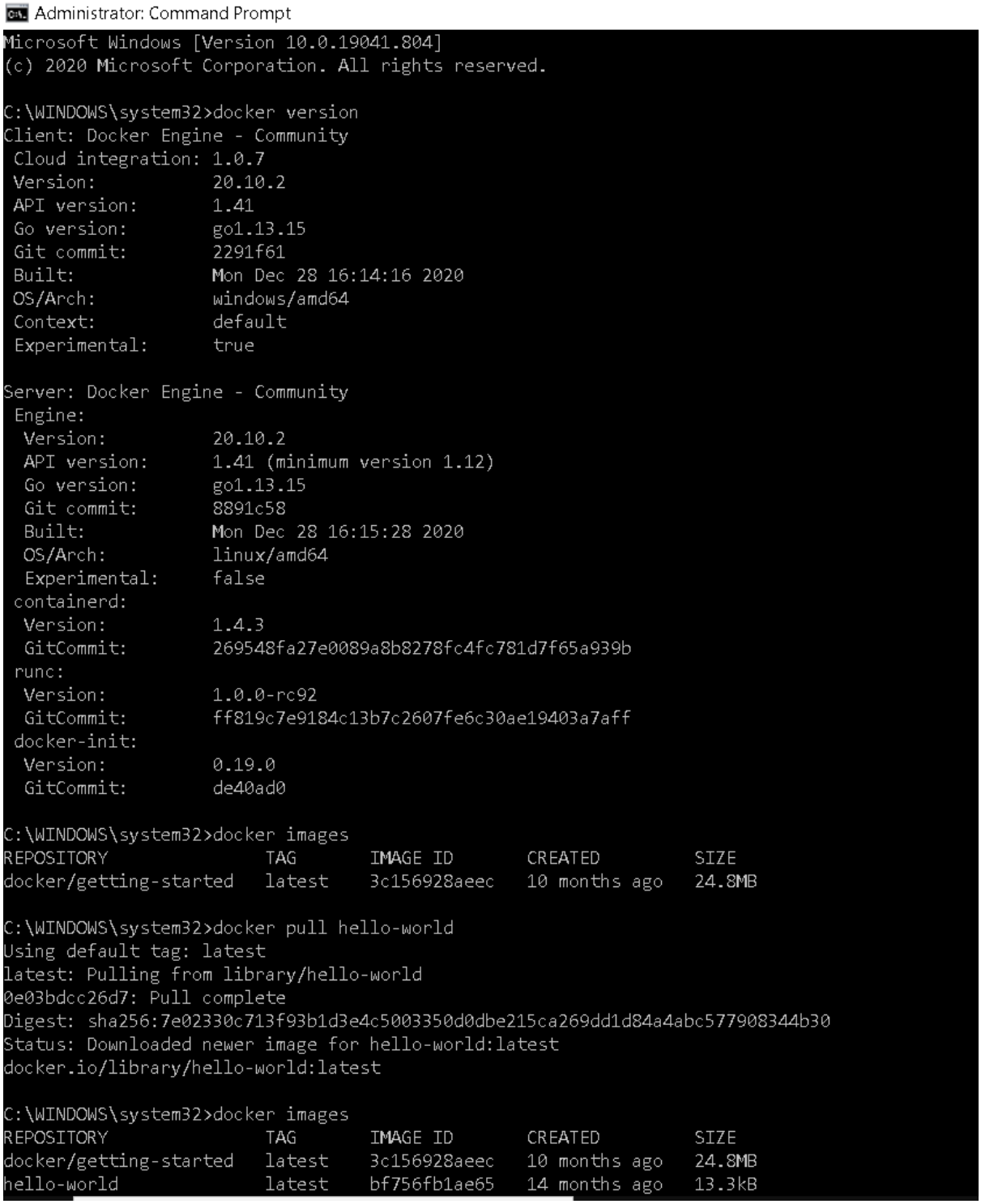


5. **Dockerized the web application**, i.e., deployed the web app into a Docker container.
   5.1. Installed the Docker Desktop app.
   5.2. Created a Docker file in the same directory where the model file, environment requirements file (modules with specific versions that are required for running the model) exists. Docker file basically consists of a set of instructions/commands.
   5.3. Built the docker image (an image is created when the docker file is executed by the docker engine during runtime). Docker image is an executable package built from the docker file using the 'docker build' command.
   5.4. Docker Container (a live instance of docker image) is created using the 'docker run' command. Run the docker container locally from the docker image and test the application before publishing it on the cloud.

Screenshots for Step 5 Process:

Install Docker Desktop for Windows and check if the docker is installed and working properly or not.

- By checking the version, list of images, pulling one sample image and running to check if it's working or not.



Administrator: Command Prompt

```
C:\WINDOWS\system32>docker images
REPOSITORY               TAG       IMAGE ID       CREATED         SIZE
docker/getting-started   latest    3c156928aeec   10 months ago   24.8MB

C:\WINDOWS\system32>docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:7e02330c713f93b1d3e4c5003350d0dbe215ca269dd1d84a4abc577908344b30
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest

C:\WINDOWS\system32>docker images
REPOSITORY               TAG       IMAGE ID       CREATED         SIZE
docker/getting-started   latest    3c156928aeec   10 months ago   24.8MB
hello-world              latest    bf756fb1ae65   14 months ago   13.3kB

C:\WINDOWS\system32>docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Go to the current project directory (where the code file, docker file, and requirements file exist) in the local system.

```
(base) C:\Users\Priyanka Naikade>docker images
REPOSITORY               TAG       IMAGE ID       CREATED         SIZE
docker/getting-started   latest    3c156928aeec   10 months ago   24.8MB
hello-world              latest    bf756fb1ae65   14 months ago   13.3kB

(base) C:\Users\Priyanka Naikade>cd C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>dir
 Volume in drive C is Windows
 Volume Serial Number is E897-3D83

 Directory of C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction

28-02-2021  01:38    <DIR>          .
28-02-2021  01:38    <DIR>          ..
27-02-2021  17:00    <DIR>          .ipynb_checkpoints
28-02-2021  00:48             3,317 app.py
27-02-2021  21:18         3,775,328 Boston_House_Price_Prediction.ipynb
28-02-2021  01:38               644 Dockerfile
27-02-2021  21:18               644 model.pkl
27-02-2021  19:58               114 requirements.txt
26-02-2021  04:27               191 test_file.csv
               6 File(s)      3,780,238 bytes
               3 Dir(s)  779,109,097,472 bytes free

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>
```

Build the docker image

>> docker build -t dock-hprice-app .

```
(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker build -t dock-hprice-app .
[+] Building 88.9s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                                                 2.2s
 => => transferring dockerfile: 683B                                                                                 0.3s
 => [internal] load .dockerignore                                                                                    2.5s
 => => transferring context: 2B                                                                                      0.1s
 => [internal] load metadata for docker.io/library/python:3.8.5-slim                                                 3.7s
 => [auth] library/python:pull token for registry-1.docker.io                                                        0.0s
 => [internal] load build context                                                                                    0.8s
 => => transferring context: 4.32kB                                                                                  0.3s
 => [1/4] FROM docker.io/library/python:3.8.5-slim@sha256:502cd057453744145010eceb5a4af1e4f04ebed54f6e1e8d23d29e  21.8s
 => => resolve docker.io/library/python:3.8.5-slim@sha256:502cd057453744145010eceb5a4af1e4f04ebed54f6e1e8d23d29eb  0.0s
 => => sha256:502cd057453744145010eceb5a4af1e4f04ebed54f6e1e8d23d29ebe2afdbe6d 1.86kB / 1.86kB                       0.0s
 => => sha256:7eae34b43d562e5dcd08c0c637e14f0f66a21209fe2591660ff37cfb7e5fa9bb 1.37kB / 1.37kB                       0.0s
 => => sha256:6cf621cb132714e451d5f6d8f7567b890f379db9f85e53cbb4862ab52f658d45 7.69kB / 7.69kB                       0.0s
 => => sha256:d121f8d1c4128ebc1e95e5bfad90a0189b84eadbbb2fbaad20cbb26d20b2c8a2 27.09MB / 27.09MB                     4.2s
 => => sha256:776716b0aa87537801fc2287d4c3796c233bfe1ce6568038494d101ed2e3f291 10.42MB / 10.42MB                     3.5s
 => => sha256:ca572574cc82ba635292c0aba70b719de39faffe7a40a802bb0f8c22663b51ac 2.75MB / 2.75MB                       2.8s
 => => sha256:c37461631ba3a30c811c3427aa9e32b65c94551d6d8d2fbd26bc39bc54c98219 233B / 233B                           4.3s
 => => extracting sha256:d121f8d1c4128ebc1e95e5bfad90a0189b84eadbbb2fbaad20cbb26d20b2c8a2                            1.9s
 => => sha256:bdb458dd27ea53bf3830e3e158cff3c11d3580fa6bf7b9691022da4e963ab4c3 2.41MB / 2.41MB                       5.9s
 => => extracting sha256:ca572574cc82ba635292c0aba70b719de39faffe7a40a802bb0f8c22663b51ac                            0.3s
 => => extracting sha256:776716b0aa87537801fc2287d4c3796c233bfe1ce6568038494d101ed2e3f291                            0.6s
 => => extracting sha256:c37461631ba3a30c811c3427aa9e32b65c94551d6d8d2fbd26bc39bc54c98219                            0.0s
 => => extracting sha256:bdb458dd27ea53bf3830e3e158cff3c11d3580fa6bf7b9691022da4e963ab4c3                            0.3s
 => [2/4] WORKDIR /pricepredapp                                                                                      4.6s
 => [3/4] COPY . /pricepredapp                                                                                       1.5s
 => [4/4] RUN pip3 install --no-cache-dir -r requirements.txt                                                       50.7s
 => exporting to image                                                                                               4.3s
 => => exporting layers                                                                                              3.8s
 => => writing image sha256:d20ece7d74df1c30a707d957474f60a2fb9b32edd0cf6bbaaa4d740cf1a29e59                         0.1s
 => => naming to docker.io/library/dock-hprice-app                                                                   0.1s

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>
```

Check the list of docker images

```
(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker images
REPOSITORY               TAG       IMAGE ID       CREATED         SIZE
dock-hprice-app          latest    d20ece7d74df   5 minutes ago   416MB
docker/getting-started   latest    3c156928aeec   10 months ago   24.8MB
hello-world              latest    bf756fb1ae65   14 months ago   13.3kB

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>
```

No container running as of now

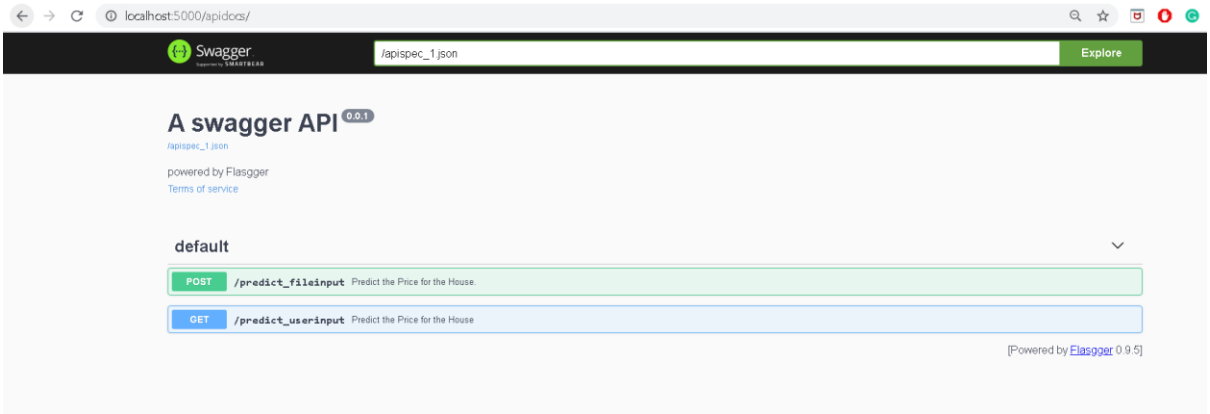
```
(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>
```

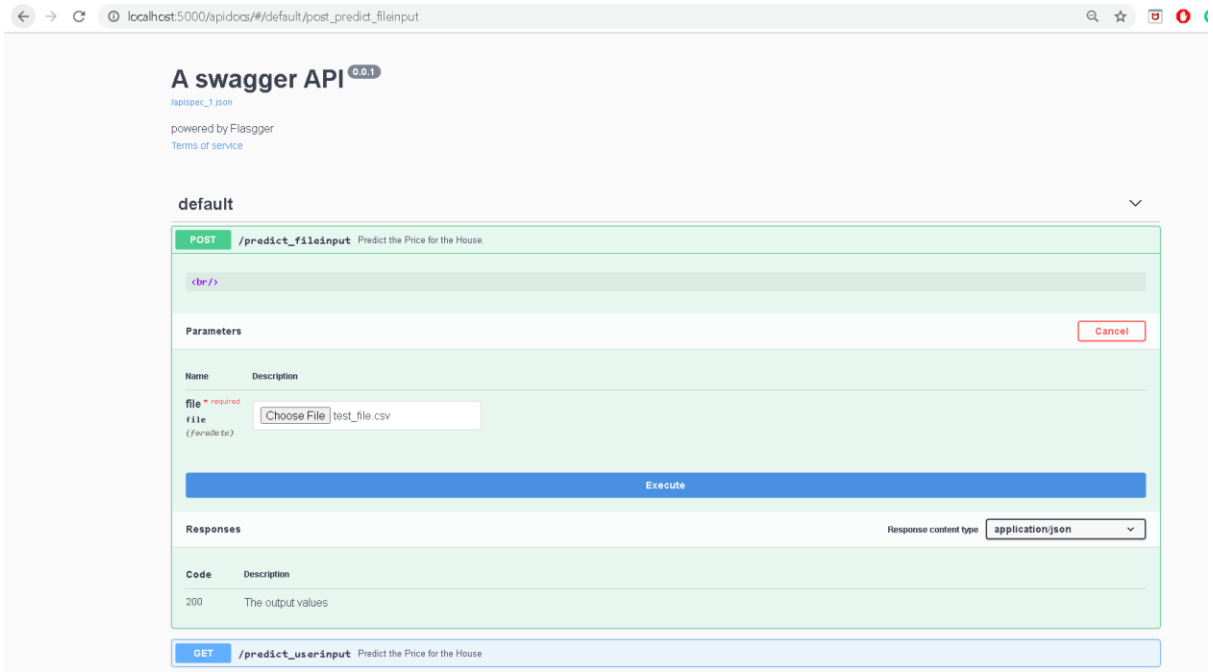
Run the docker image 'dock-hprice-app'

>> docker run -p 5000:5000 dock-hprice-app

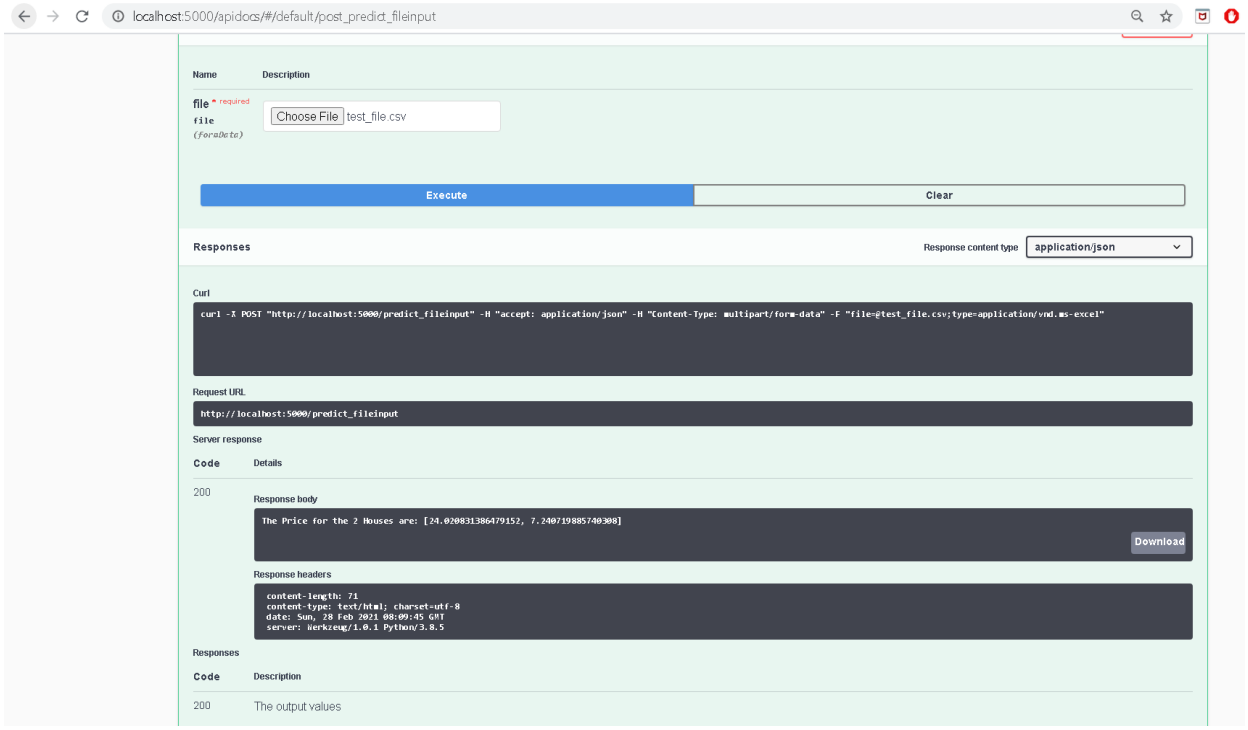
```
(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker run -p 5000:5000 dock-hprice-app
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
172.17.0.1 - - [28/Feb/2021 08:06:11] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [28/Feb/2021 08:06:19] "GET /apidocs/ HTTP/1.1" 200 -
172.17.0.1 - - [28/Feb/2021 08:06:19] "GET /apispec_1.json HTTP/1.1" 200 -

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker ps
CONTAINER ID   IMAGE            COMMAND           CREATED          STATUS          PORTS                    NAMES
e0d07fafa220   dock-hprice-app   "python app.py"   38 seconds ago   Up 35 seconds   0.0.0.0:5000->5000/tcp   angry_agnes
i

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>
```

Given the container is running, we will test the web app on ***localhost:5000/apidocs/***




localhost:5000/apidocs/#/default/post_predict_fileinput
A swagger API 0.0.1
/apispec_1.json
powered by Flasgger
Terms of service
default
POST /predict_fileinput Predict the Price for the House.

Parameters
Cancel
Name
Description
file * required
file
(formData)
Choose File test_file.csv
Execute
Responses
Response content type application/json
Code
Description
200
The output values
GET /predict_userinput Predict the Price for the House


localhost:5000/apidocs/#/default/post_predict_fileinput
Name
Description
file * required
file
(formData)
Choose File test_file.csv
Execute
Clear
Responses
Response content type application/json
Curl
curl -X POST "http://localhost:5000/predict_fileinput" -H "accept: application/json" -H "Content-Type: multipart/form-data" -F "file=@test_file.csv;type=application/vnd.ms-excel"
Request URL
http://localhost:5000/predict_fileinput
Server response
Code
Details
200
Response body
The Price for the 2 Houses are: [24.020831386479152, 7.240719885740308]
Download
Response headers
content-length: 71
content-type: text/html; charset=utf-8
date: Sun, 28 Feb 2021 08:09:45 GMT
server: Werkzeug/1.0.1 Python/3.8.5
Responses
Code
Description
200
The output values

```
(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker ps
CONTAINER ID   IMAGE            COMMAND           CREATED          STATUS          PORTS                    NAMES
e0d07fafa220   dock-hprice-app   "python app.py"   38 seconds ago   Up 35 seconds   0.0.0.0:5000->5000/tcp   angry_agnes
i

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker ps
CONTAINER ID   IMAGE            COMMAND           CREATED         STATUS         PORTS                    NAMES
e0d07fafa220   dock-hprice-app   "python app.py"   5 minutes ago   Up 5 minutes   0.0.0.0:5000->5000/tcp   angry_agnesi

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>
```

We can now stop the docker container

```
(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker ps
CONTAINER ID   IMAGE            COMMAND           CREATED         STATUS         PORTS                    NAMES
e0d07fafa220   dock-hprice-app   "python app.py"   5 minutes ago   Up 5 minutes   0.0.0.0:5000->5000/tcp   angry_agnesi

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker stop e0d07fafa220
e0d07fafa220

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>
```

Created another image after changing requirements.txt

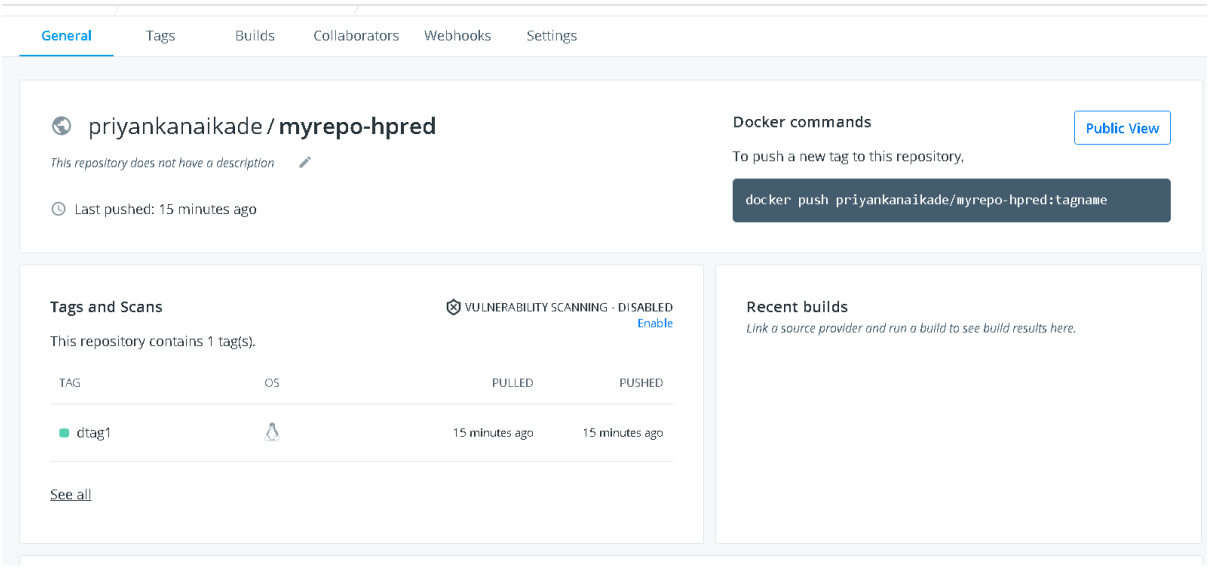
```
(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker images
REPOSITORY               TAG       IMAGE ID       CREATED         SIZE
dock2-price-app          latest    e09495fed141   7 minutes ago   441MB
dock-hprice-app          latest    3971acf0a999   16 hours ago    416MB
docker/getting-started   latest    3c156928aeec   10 months ago   24.8MB
hello-world              latest    bf756fb1ae65   14 months ago   13.3kB

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker ps
CONTAINER ID   IMAGE            COMMAND           CREATED              STATUS          PORTS                    NAMES
04b0255557ea   dock2-price-app   "python app.py"   About a minute ago   Up 25 seconds   0.0.0.0:5000->5000/tcp   tender_
wilbur

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker ps
CONTAINER ID   IMAGE            COMMAND           CREATED         STATUS         PORTS                    NAMES
04b0255557ea   dock2-price-app   "python app.py"   3 minutes ago   Up 2 minutes   0.0.0.0:5000->5000/tcp   tender_wilbur

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker stop 04b0255557ea
04b0255557ea

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker images
REPOSITORY               TAG       IMAGE ID       CREATED          SIZE
dock2-price-app          latest    e09495fed141   10 minutes ago   441MB
dock-hprice-app          latest    3971acf0a999   16 hours ago     416MB
docker/getting-started   latest    3c156928aeec   10 months ago    24.8MB
hello-world              latest    bf756fb1ae65   14 months ago    13.3kB

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker tag
```

6. **Publishing the docker image to docker hub repository which could be accessed from anywhere**
   6.1. Created a public repository in docker hub.
   6.2. Tagged the image to the repository

```
(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker tag dock2-price-app priyankanaikade/myrepo-hpred:dtag1

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>
```

   6.3. Pushed the docker image to the repository

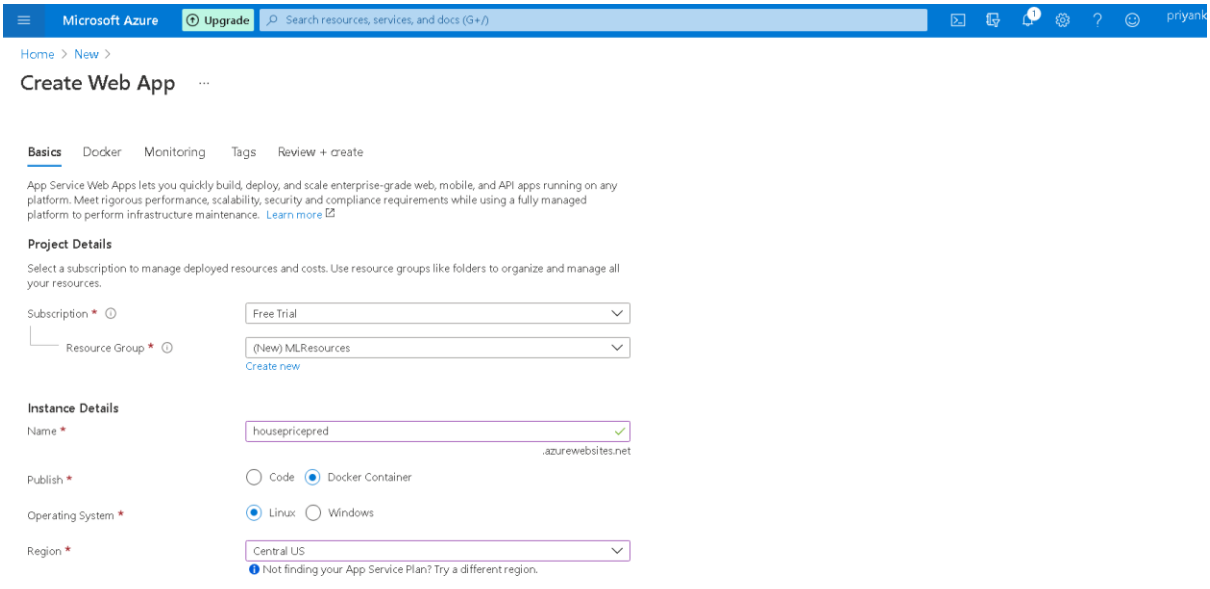
```
(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>docker push priyankanaikade/myrepo-hpred:dtag1
The push refers to repository [docker.io/priyankanaikade/myrepo-hpred]
434366b74fe1: Pushed
afba96b65a6a: Pushed
230a236c642b: Pushed
bb38c3f3fcc5: Mounted from library/python
9f72a1f4c21f: Mounted from library/python
6280b41d048d: Mounted from library/python
bd8e6688d36c: Mounted from library/python
07cab4339852: Mounted from library/python
dtag1: digest: sha256:0374091fac016cde8c8b21b1da4a2a7ed34fca67082bb33f885e7fa1c2ff0a52 size: 1997

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>
```

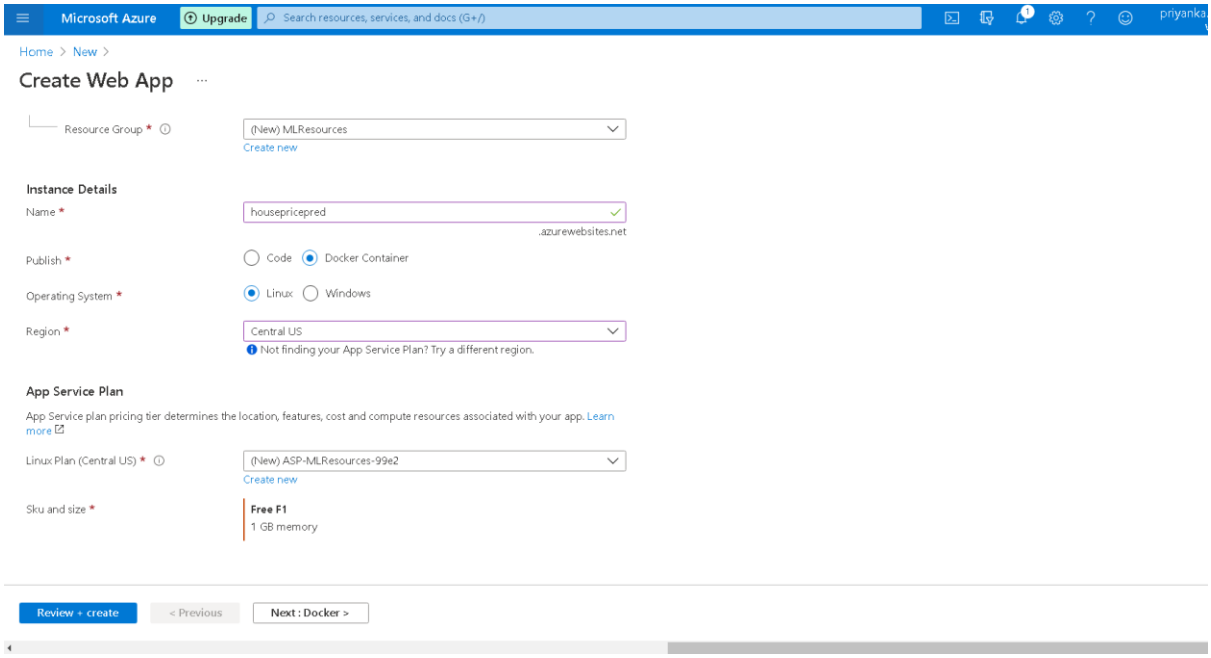


7. Hosting the docker container on Azure

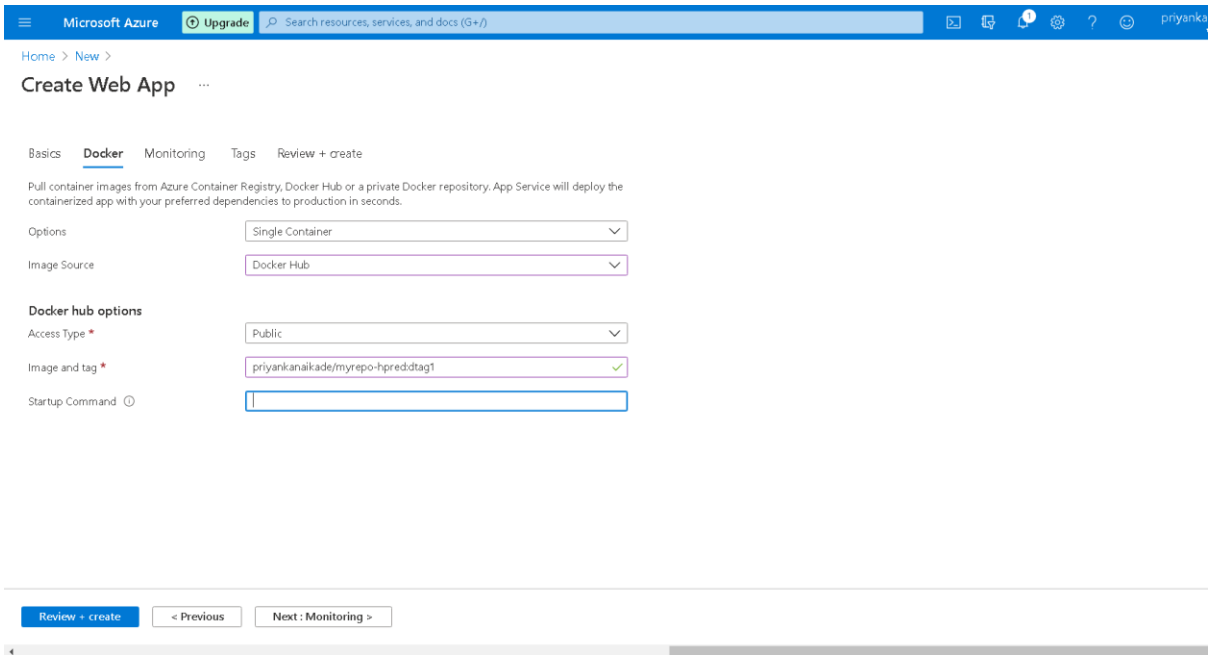
7.1. Open Azure Portal -> Select ‘Create a Resource’ -> Web App

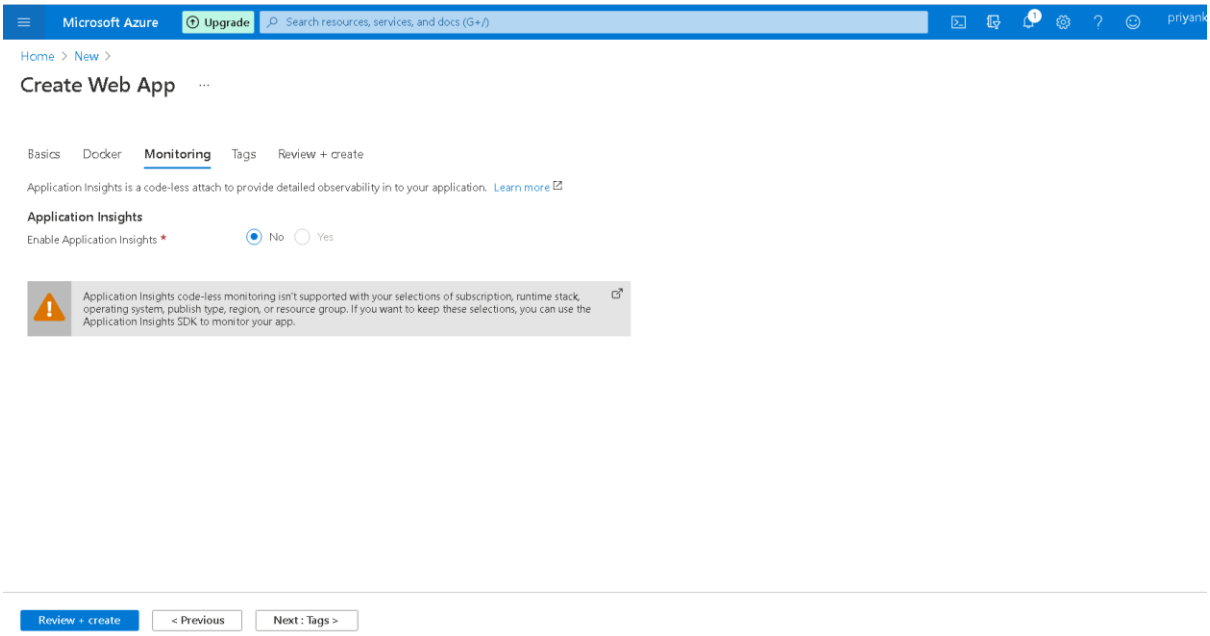
- Resource group -> Existing or create a new one if required.
- Publish -> Docker Container. Since we already created the docker container.



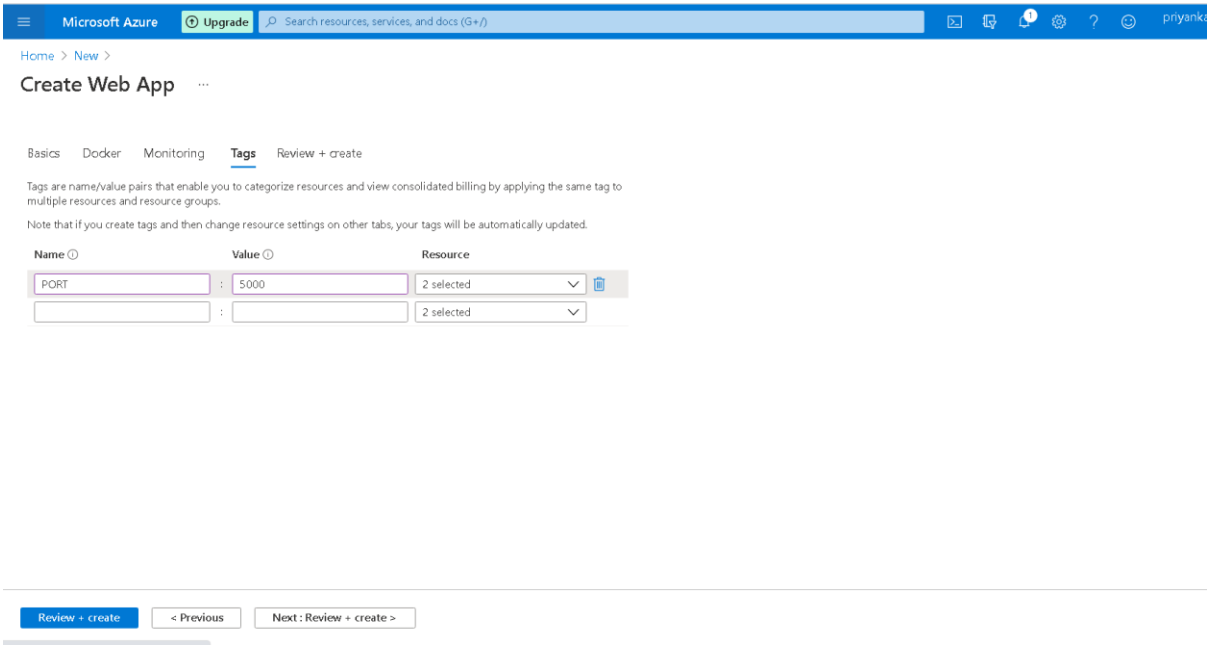


- Select Image source as ‘Docker Hub’ and give its details (Docker Image with Tag).

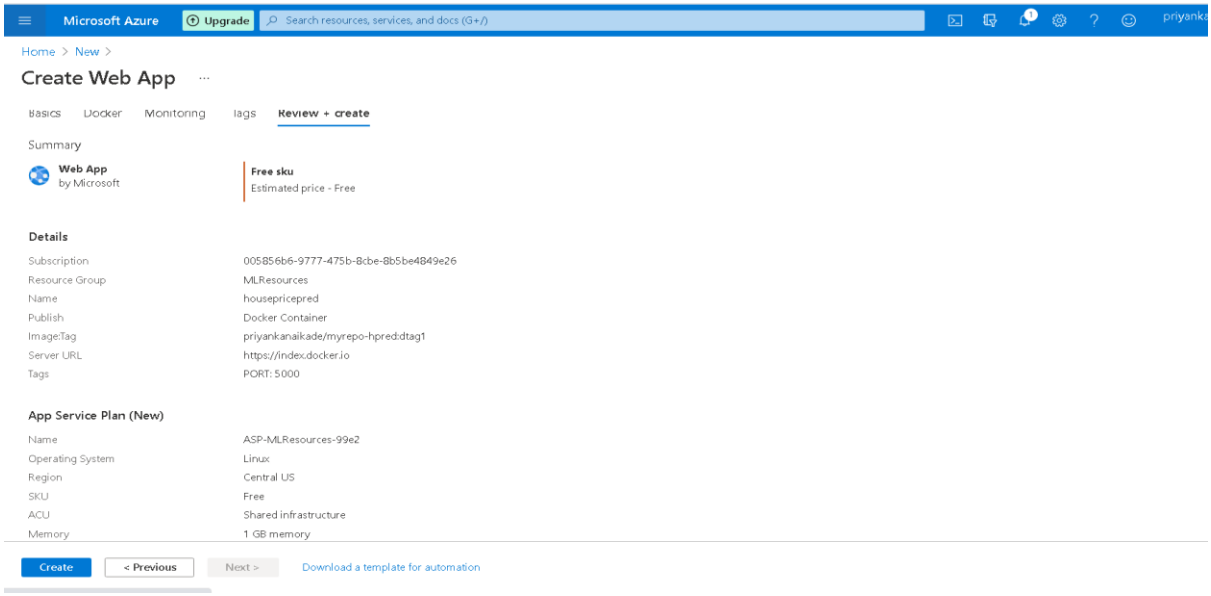




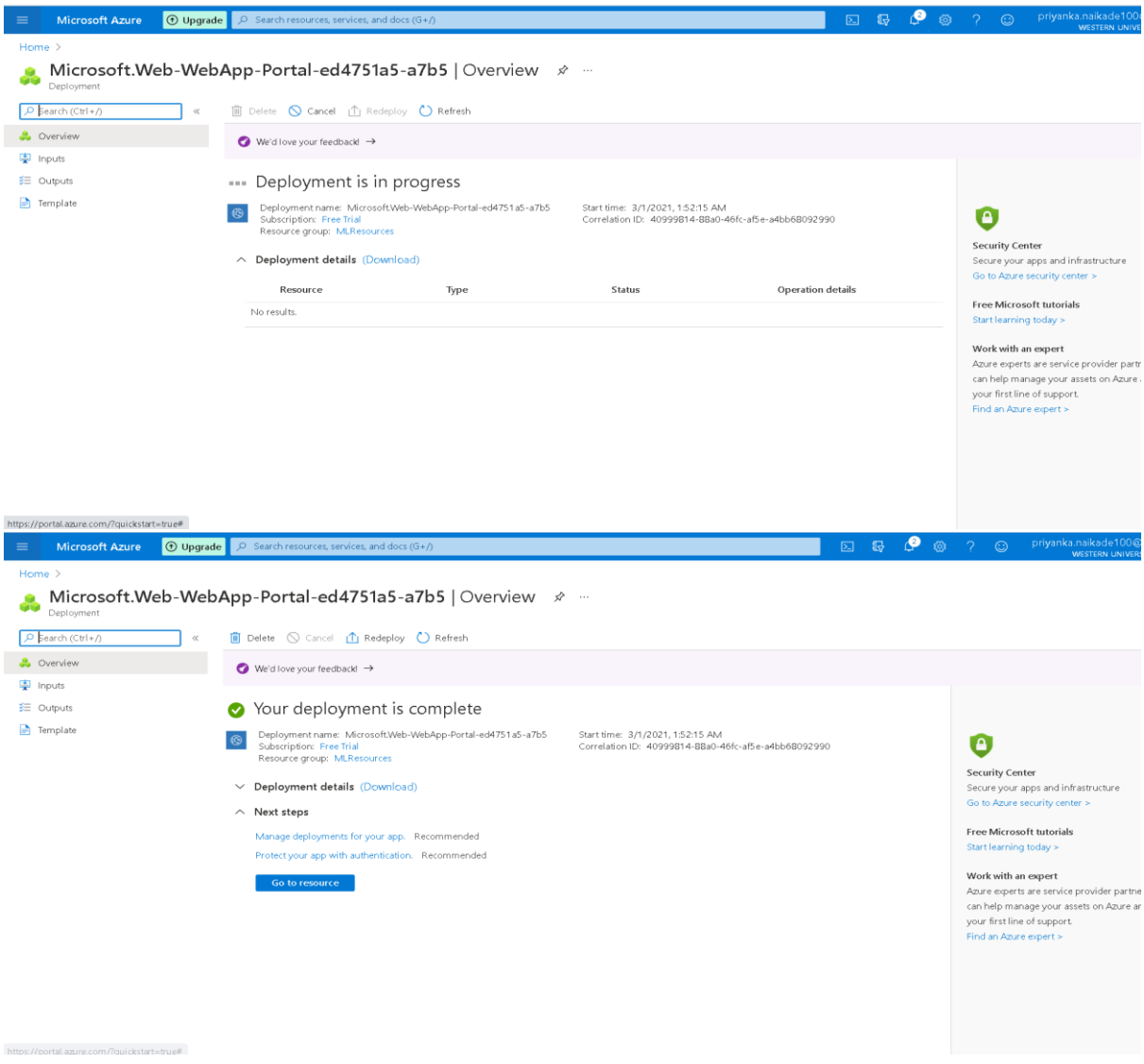
- Under Tags: Add the PORT field and specify the port number as 5000.

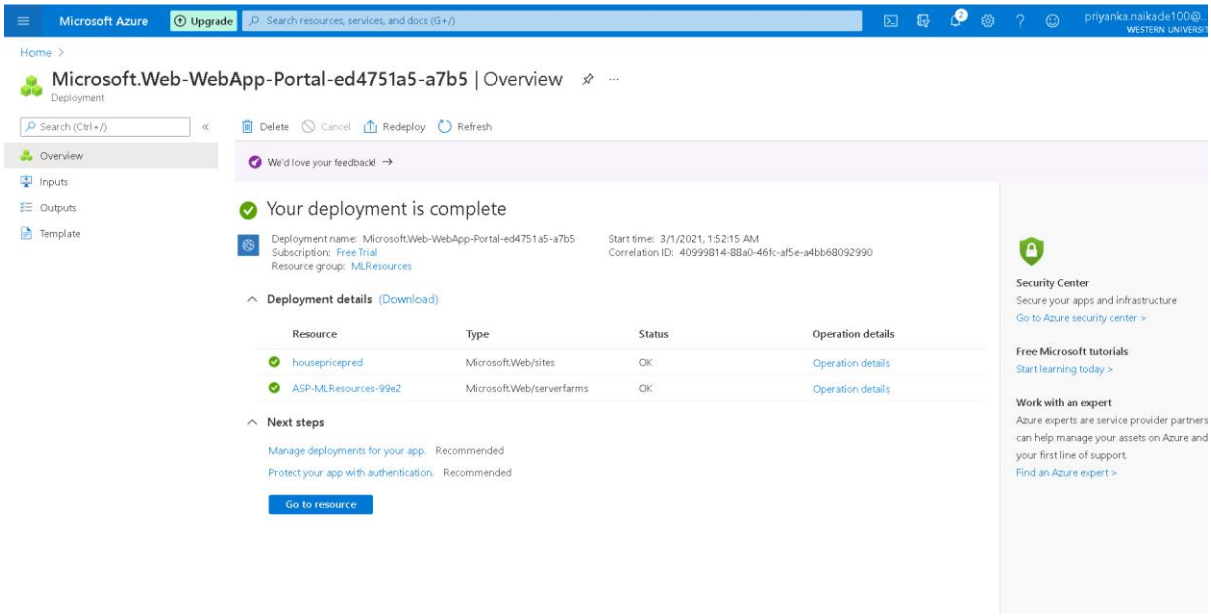


- Click ‘Review + Create’

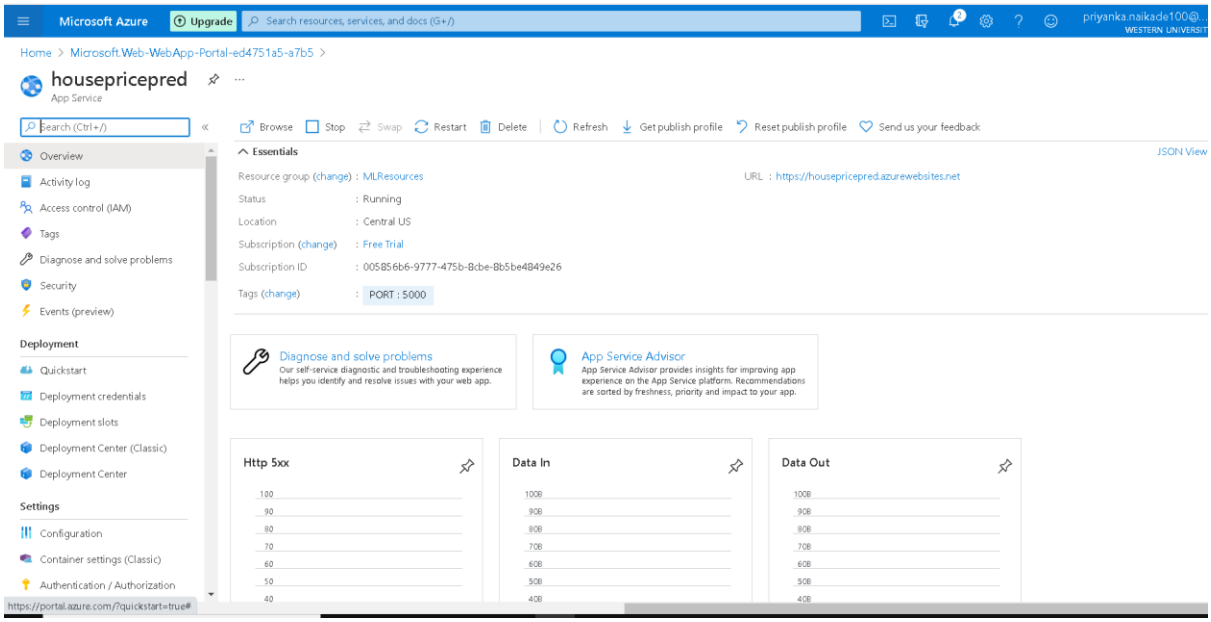


- Click on Create.





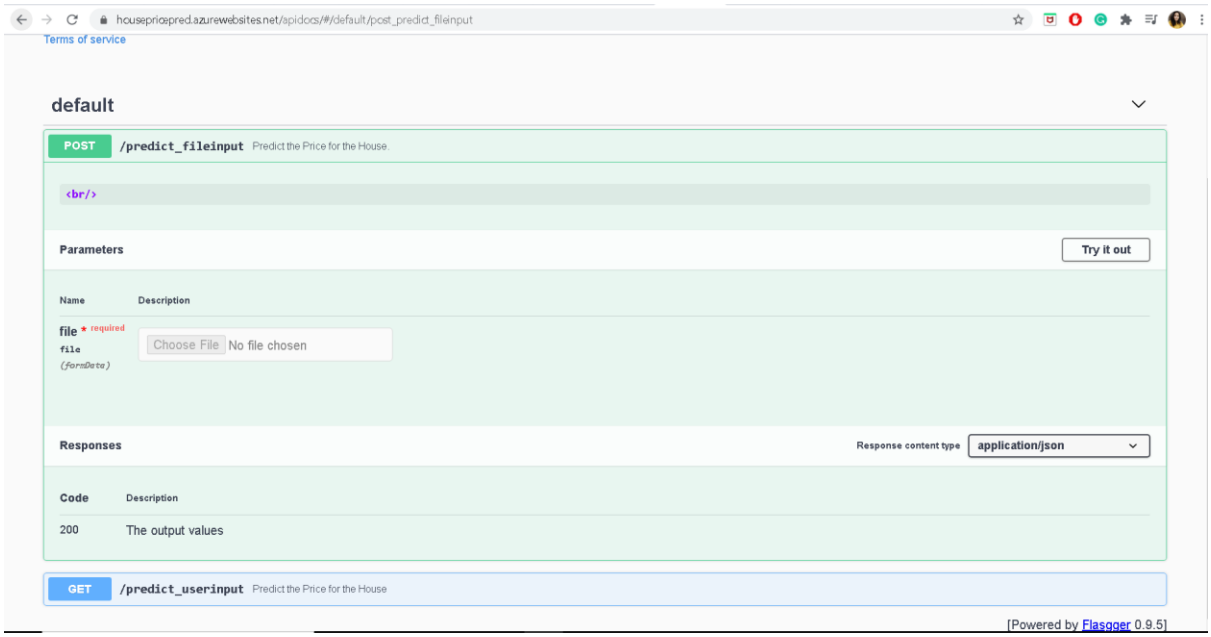
- Click on 'Go to resource'

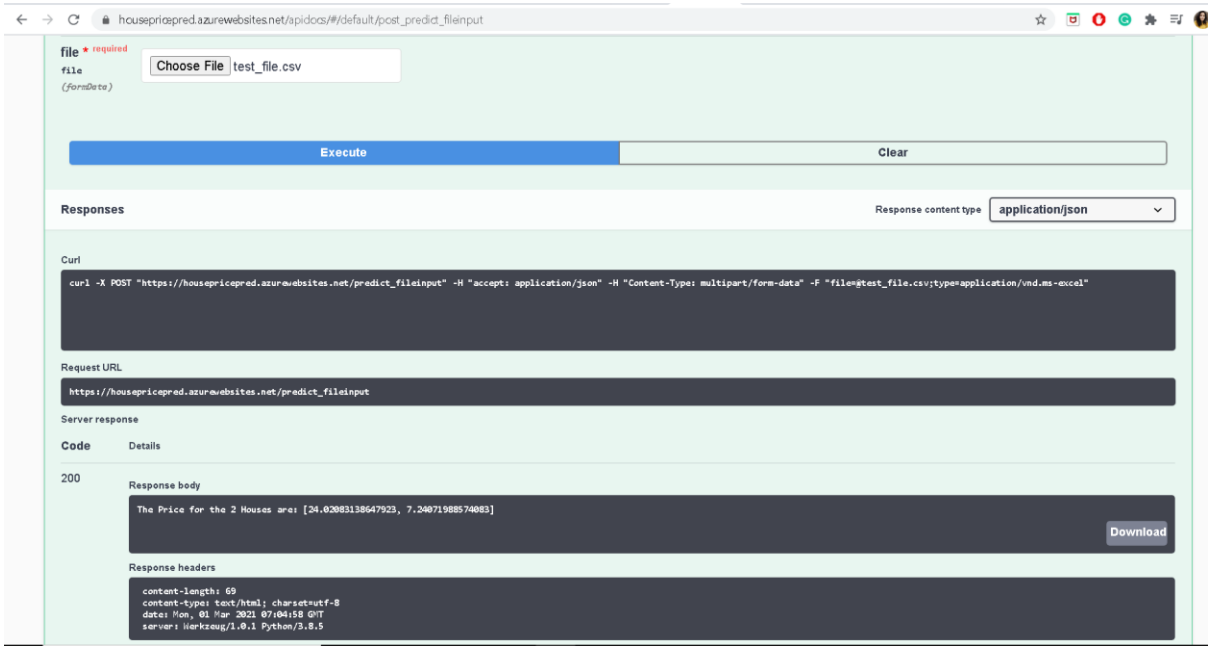


We can see that the status is running and now we can test the app using the URL mentioned.

URL opens: 'Welcome to home page'
To open the app add '/apidocs/' at the end of the URL.

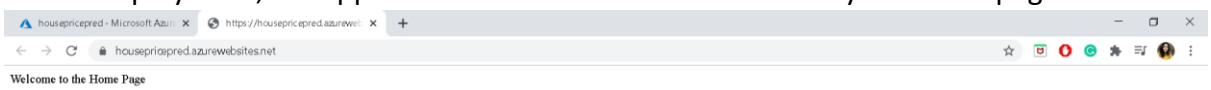
https://housepricepred.azurewebsites.net/apidocs/



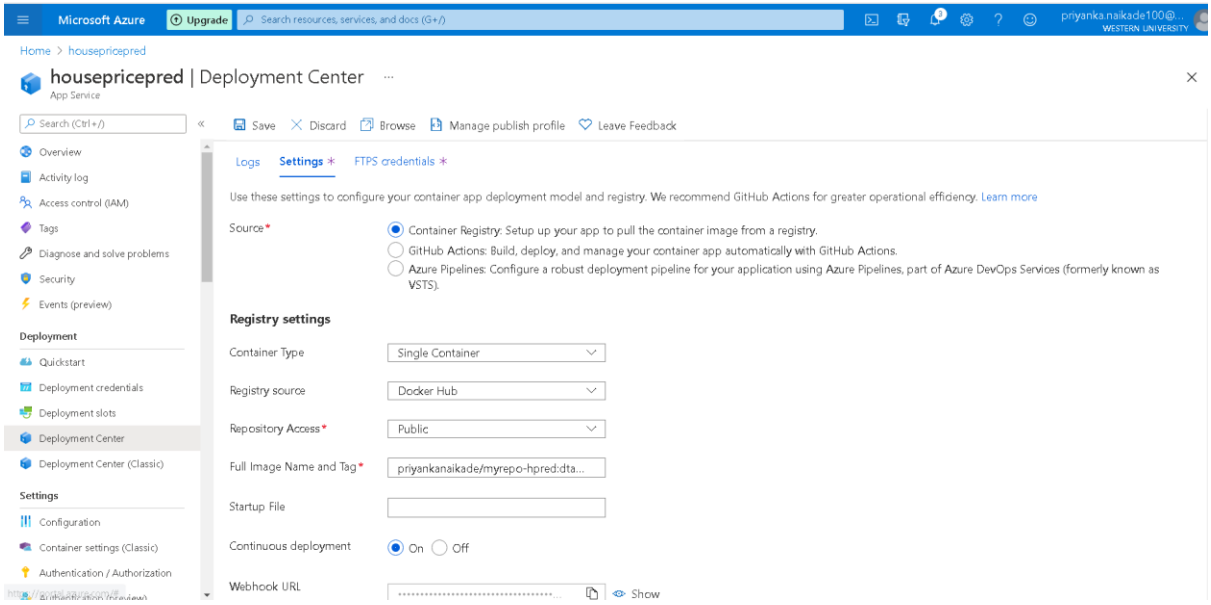


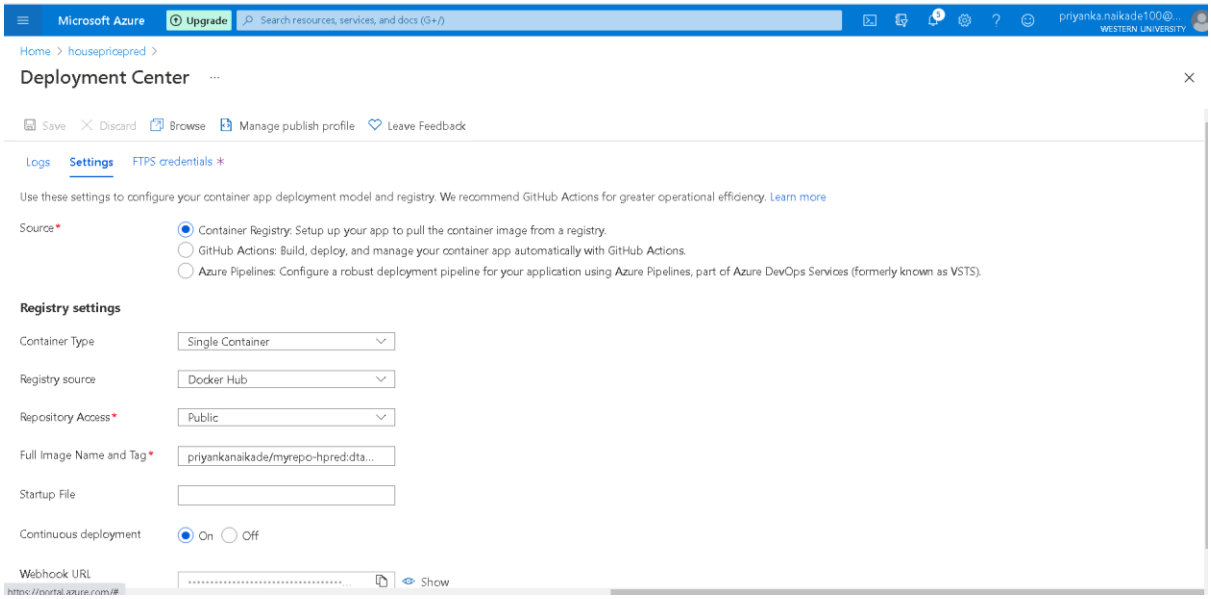
8. Continuous Integration and Continuous Deployment (CI/CD)

Before deployment, the application looks like this. We will modify this homepage.



8.1 **Azure Web Service Settings:** To ensure continuous integration and deployment, Go to Deployment Center -> turn ‘On’ the Continuous Deployment (CD) option. Go to Container settings -> set CD = ‘On’





8.2 Code Modification: Modified the home page using a HTML file and added a button that will redirect Swagger webpage. Updates made to app.py and a new file homepage.html under templates folder is added to the project directory.

app.py | Dockerfile | requirements.txt | homepage.html

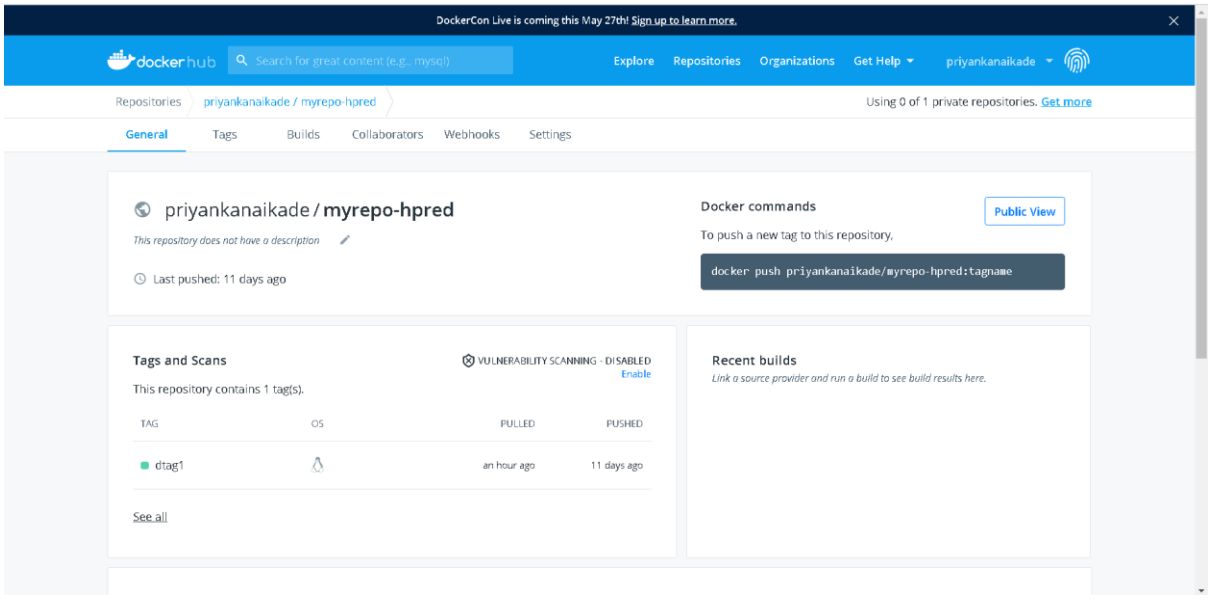
```
<!DOCTYPE html>

<html>
    <head>
    <meta charset = "utf-8">
    <title> House Price Prediction </title>
    </head>

    <body>
    <h1> Boston House Price Prediction </h1>
    <button id = "btn" > Click here for Prediction </button>
    <script>
            var btn = document.getElementById("btn");
            btn.onclick = function() {
            window.open("https://housepricepred.azurewebsites.net/apidocs/")};
    </script>

    </body>

</html>
```

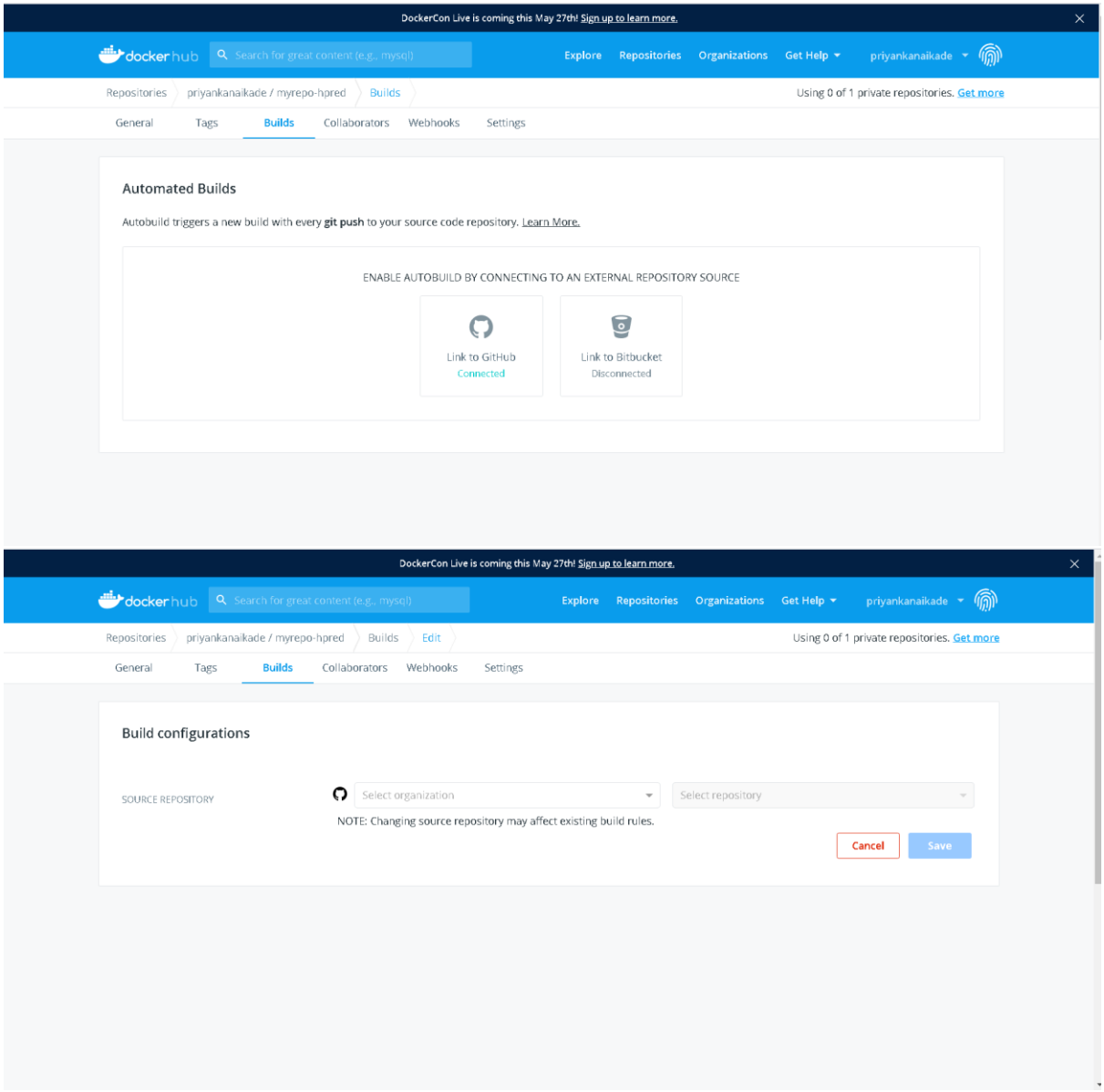
```
(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>dir
 Volume in drive C is Windows
 Volume Serial Number is E897-3D83

 Directory of C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction

12-03-2021  05:09    <DIR>          .
12-03-2021  05:09    <DIR>          ..
12-03-2021  05:09             3,365 app.py
28-02-2021  18:25               638 Dockerfile
28-02-2021  18:12               656 model.pkl
28-02-2021  18:09             2,316 pricepredmodel.py
28-02-2021  06:30                66 requirements.txt
12-03-2021  05:47    <DIR>          templates
               5 File(s)          7,041 bytes
               3 Dir(s)  772,416,737,280 bytes free

(base) C:\Users\Priyanka Naikade\Desktop\House-Price-Prediction>
```

8.3 Update the docker image in the docker hub.

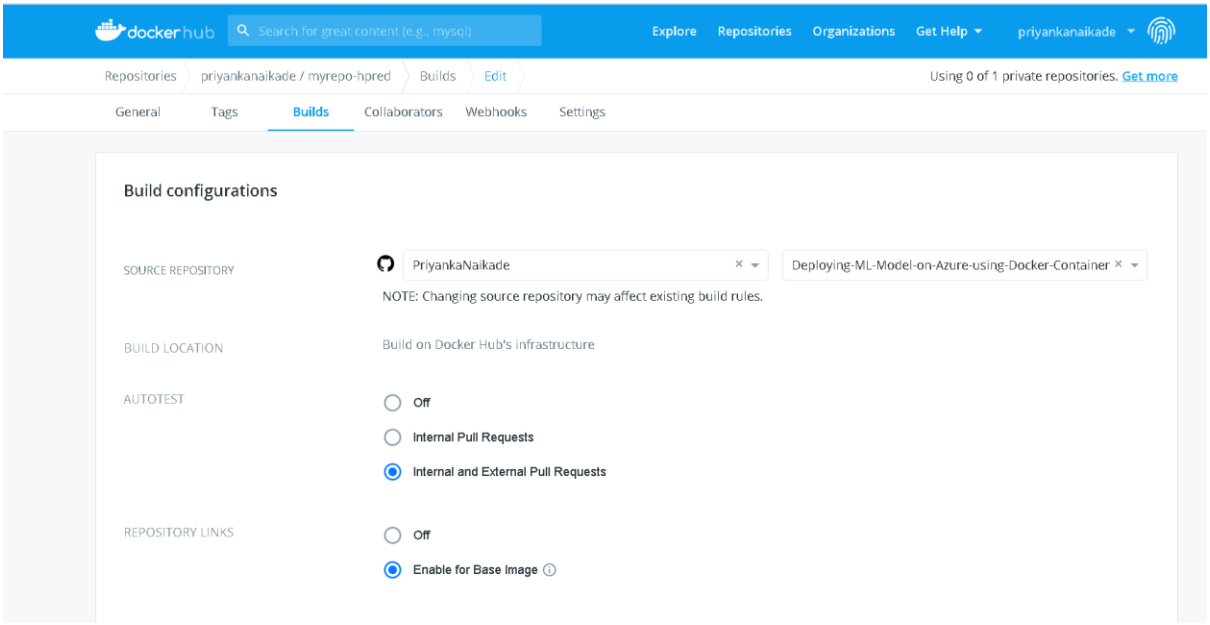
It should be an automated process.

We link the docker hub account to our git hub account under account settings. Then configure the automated build settings. Go to public repository that we created in docker hub -> Builds ->



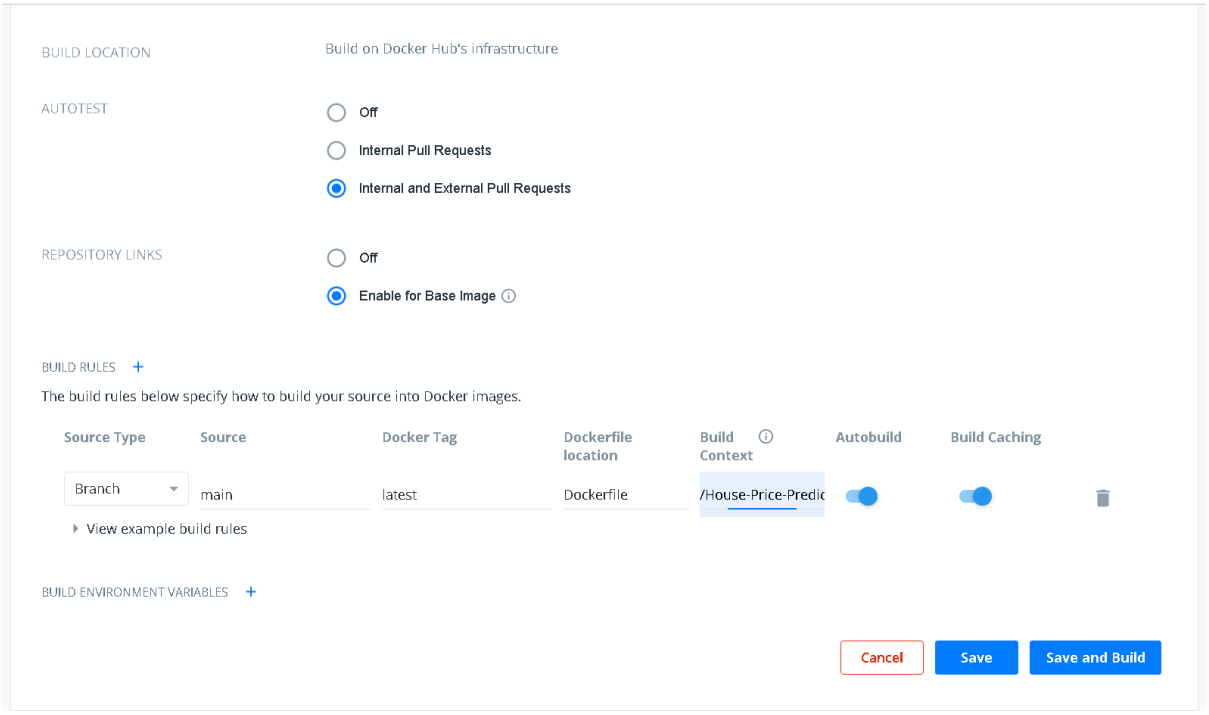


Set the Build Configurations so that whenever there is a new push in the git repository the docker hub will create a new build or update the docker image

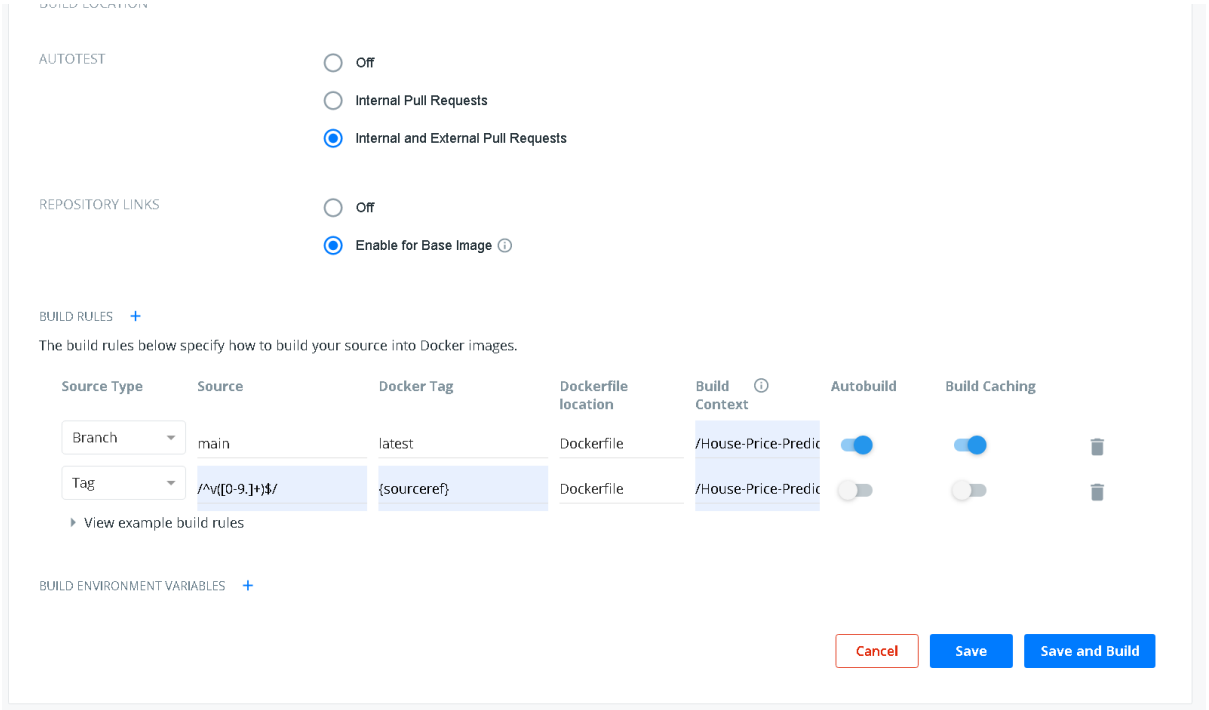


To **Configure automated builds**: provide source repository, repository name, autotest on for internal pull requests, enable automated build creation for any update in the base image of hub and other details.

Set a rule to build the latest image



To maintain the different versions of the image, set another rule



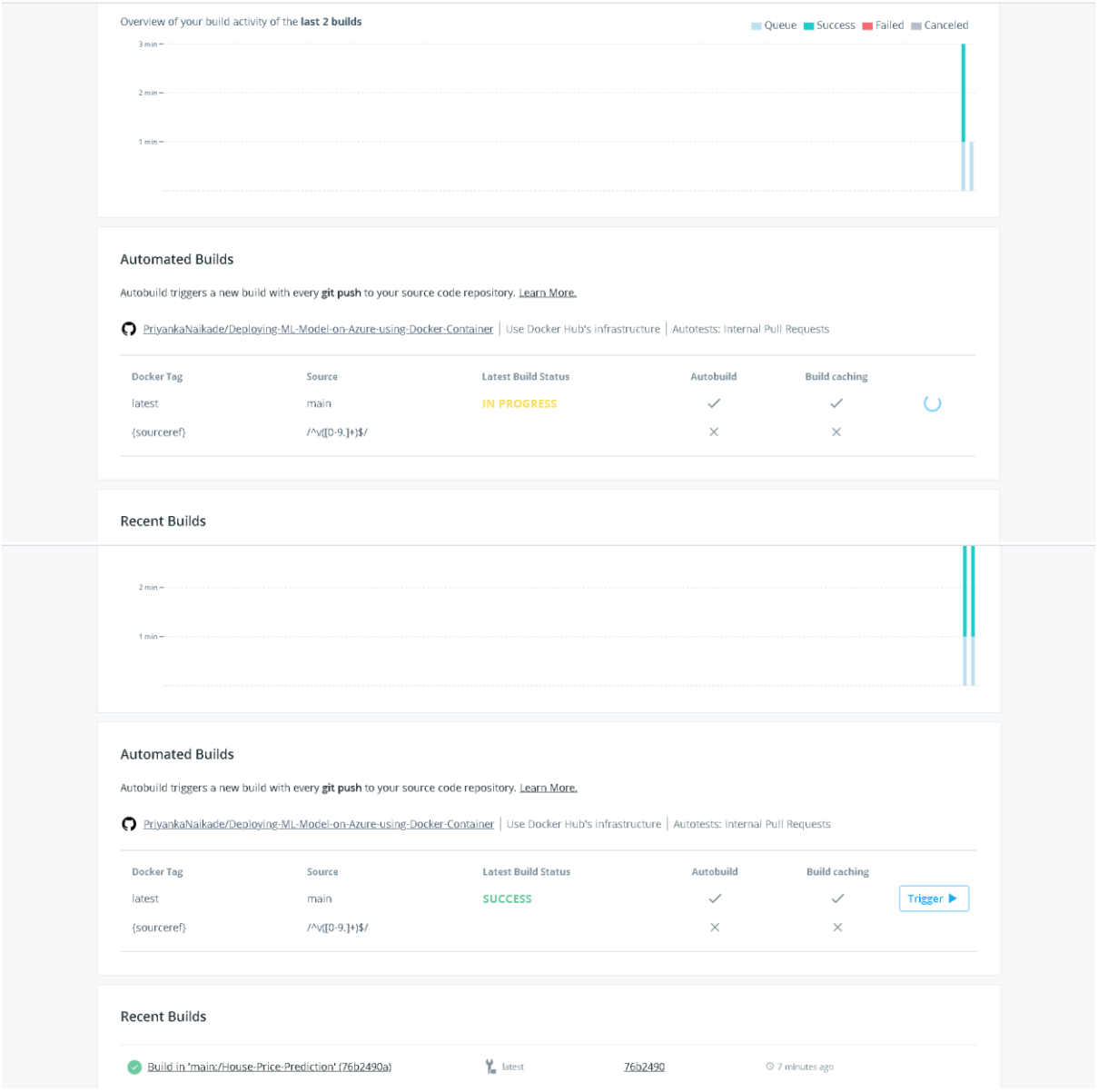
8.4 Push the new file/updated files from local machine to the github repo.

```
priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop$ git clone https://github.com/PriyankaNaikade/Deploying-ML-Model-on-Azure-using-Docker-Container.git
Cloning into 'Deploying-ML-Model-on-Azure-using-Docker-Container'...
remote: Enumerating objects: 68, done.
remote: Counting objects: 100% (68/68), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 68 (delta 29), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (68/68), done.
priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop$ cd Deploying-ML-Model-on-Azure-using-Docker-Container/
priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop/Deploying-ML-Model-on-Azure-using-Docker-Container$ ls
Documentation  House-Price-Prediction  Notebook  README.md  Test Data File
priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop/Deploying-ML-Model-on-Azure-using-Docker-Container$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   House-Price-Prediction/app.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        House-Price-Prediction/templates/

no changes added to commit (use "git add" and/or "git commit -a")
priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop/Deploying-ML-Model-on-Azure-using-Docker-Container$ git add .
priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop/Deploying-ML-Model-on-Azure-using-Docker-Container$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   House-Price-Prediction/app.py
        new file:   House-Price-Prediction/templates/homepage.html

priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop/Deploying-ML-Model-on-Azure-using-Docker-Container$ git commit -m "Modified HomePage using html for testing CI/CD"
[main 76b2490] Modified HomePage using html for testing CI/CD
```

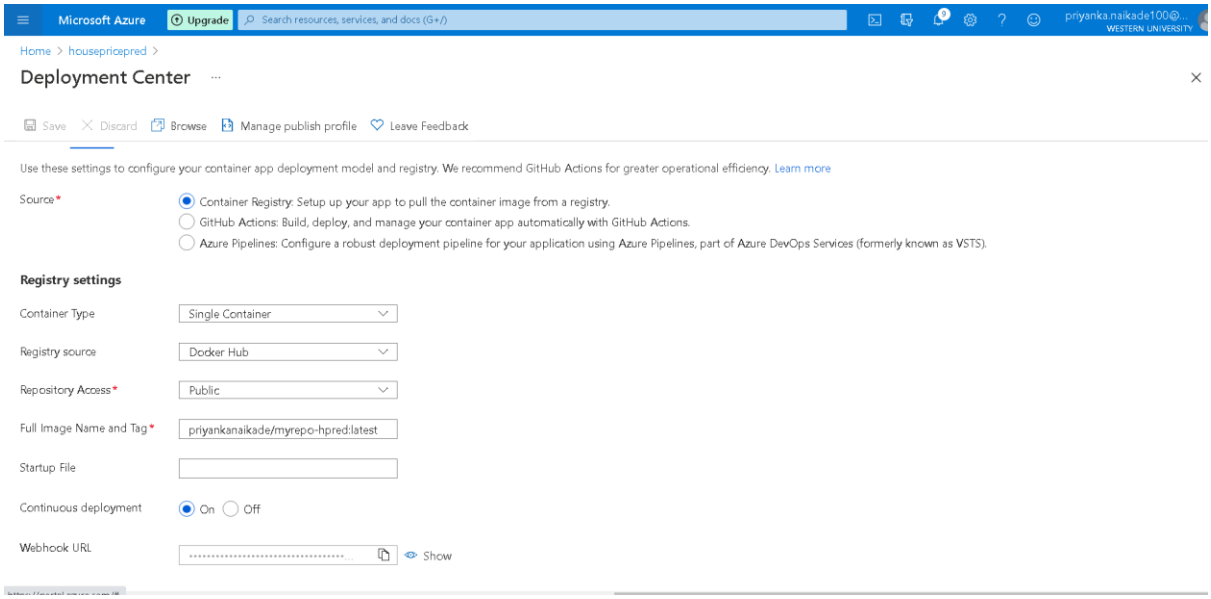
```
priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop/Deploying-ML-Model-on-Azure-using-Docker-Container$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   House-Price-Prediction/app.py
        new file:   House-Price-Prediction/templates/homepage.html

priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop/Deploying-ML-Model-on-Azure-using-Docker-Container$ git commit -m "Modified HomePage using html for testing CI/CD"
[main 76b2490] Modified HomePage using html for testing CI/CD
 2 files changed, 22 insertions(+), 2 deletions(-)
 create mode 100644 House-Price-Prediction/templates/homepage.html
priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop/Deploying-ML-Model-on-Azure-using-Docker-Container$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

```
priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop/Deploying-ML-Model-on-Azure-using-Docker-Container$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop/Deploying-ML-Model-on-Azure-using-Docker-Container$ git push
Username for 'https://github.com': priyanka.naikade100@gmail.com
Password for 'https://priyanka.naikade100@gmail.com@github.com':
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 784 bytes | 261.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/PriyankaNaikade/Deploying-ML-Model-on-Azure-using-Docker-Container.git
   8595d95..76b2490  main -> main
priyanka@LAPTOP-67BKDV1I:/mnt/c/Users/Priyanka Naikade/Desktop/Deploying-ML-Model-on-Azure-using-Docker-Container$
```

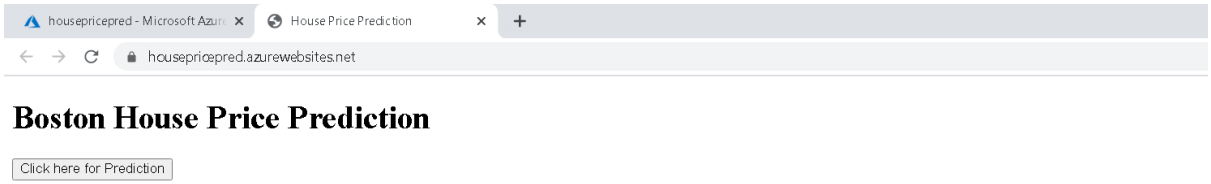
As the push gets successful, the build creation is initiated automatically in the docker hub. The build is in progress right now.



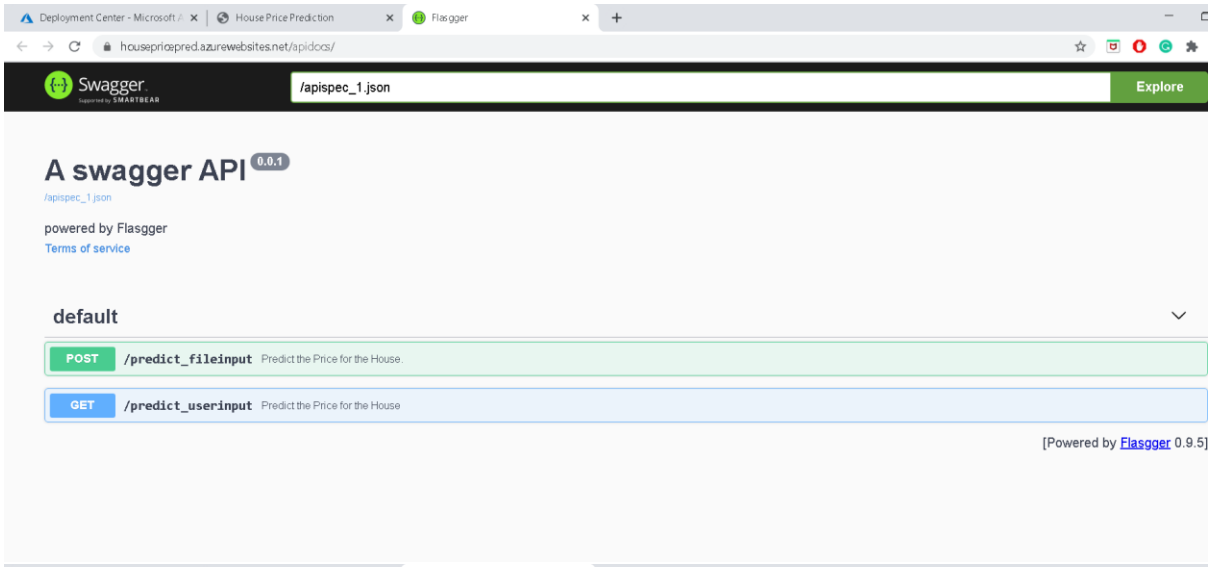
Once it's successful, we can check the application via Azure. Since the continuous deployment is set 'ON' as given in the screenshot below, Azure automatically pulls the updated image from the docker hub.

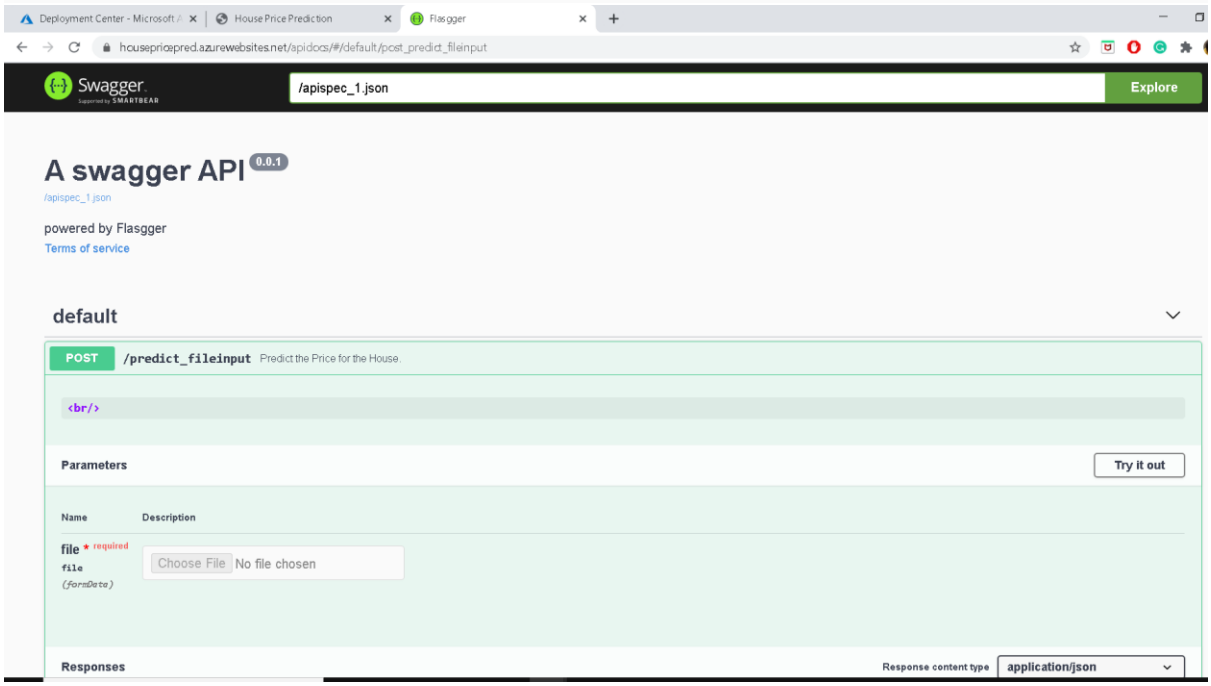


The latest application opened via Azure:



We fixed the navigation part to Swagger webpage by adding a button to the homepage which will navigate to Swagger's apidocs url.


Deployment Center - Microsoft
House Price Prediction
Flasgger
housepricepred.azurewebsites.net/apidocs/
Swagger
/apispec_1.json
Explore
A swagger API
0.0.1
/apispec_1.json
powered by Flasgger
Terms of service
default
POST
/predict_fileinput
Predict the Price for the House.
GET
/predict_userinput
Predict the Price for the House
[Powered by Flasgger 0.9.5]


Deployment Center - Microsoft
House Price Prediction
Flasgger
housepricepred.azurewebsites.net/apidocs/#/default/post_predict_fileinput
Swagger
/apispec_1.json
Explore
A swagger API
0.0.1
/apispec_1.json
powered by Flasgger
Terms of service
default
POST
/predict_fileinput
Predict the Price for the House.

Parameters
Try it out
Name
Description
file
required
file
(formData)
Choose File
No file chosen
Responses
Response content type
application/json