

Assignment 1:

Demonstration of some of the most commonly Docker commands.

- **docker --version**

This command returns the version of the Docker which is installed.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ADIO>docker --version
Docker version 20.10.17, build 100c701
```

- **docker pull <image name>**

This command is used to pull particular image which is mentioned from the Docker hub.

Ex : docker pull hello-world

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ADIO>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
cf92e523b49e: Pull complete
Digest: sha256:35fb073f9e56eb84041b0745cb714eff0f7b225ea9e024f703cab56aaa5c7720
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

- **docker images**

To list all the Docker images that we have in our local system.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ADIO>docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
ubuntu              latest          216c552ea5ba   7 days ago     77.8MB
docker/getting-started latest         cb90f98fd791   6 months ago   28.8MB
hello-world          latest         feb5d9fea6a5   12 months ago  13.3kB
```

- **docker --help**

This command will returns a list of commands available in Docker along with the possible flags (options).

```
Usage: docker [OPTIONS] COMMAND

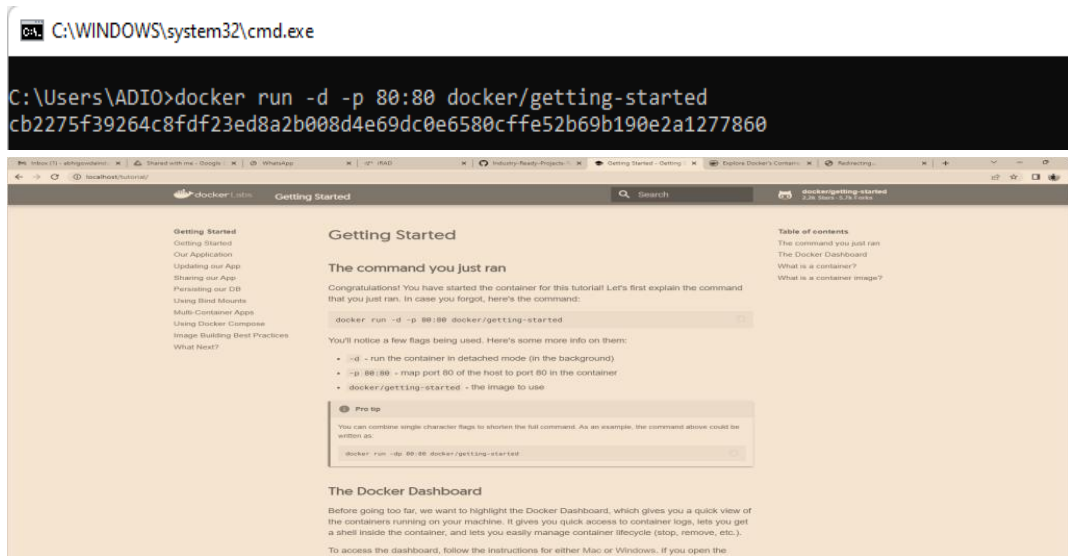
A self-sufficient runtime for containers

Options:
  -c, --context string          The name of the context to use to connect to the
                                daemon (overrides DOCKER_HOST env var and
                                default configuration of your system)
  -H, --host string             Daemon socket(s) to connect to
  -l, --log-driver string       Logging driver
  -L, --log-opt string          Logging driver options
  -m, --metrics string          Enable metrics
  -p, --pid string              Container PID file
  -s, --storage string          Storage driver
  -S, --storage-opt string       Storage driver options
  -t, --tls string              Enable TLS
  -u, --tls-key string           TLS key
  -v, --tls-cert string         TLS certificate
  -V, --tls-cacert string       TLS CA certificate
  --tls-verify                  Enable TLS verification
  -w, --workdir string          Working directory
  --debug                       Enable debug mode
  --help                        Display this help message
  --help-top                    Display just the top part of the help message
  --help-section string         Display just the section of the help message
                                specified
  --help-flag string            Display just one flag of the help message
  --help-arg string             Display just one argument of the help message

Commands:
  build             Build an image from a Dockerfile
  commit            Create a new image from a container's changes
  cp                Copy files/folders between a container and the host
  create            Create a new container
  diff              Inspect changes to files or directories on a running
                                container's filesystem
  exec              Execute a command in a running container
  export             Export a container's filesystem as a tar archive
  format             Format the Docker metadata
  get               Get information on a container or image
  history            Show the history of an image
  images            List images
  import             Import the contents from a tar archive to a new image
  info              Display system-wide information
  inspect            Return low-level information on Docker objects
  kill              Kill the container
  load              Load an image from a tar archive or STDIN
  login              Log in to Docker Hub
  logout            Log out from Docker Hub
  ls                List containers
  mkdir             Create a new directory inside a container
  mv                Move files/folders from a container to the host or
                                another container
  network           Manage Docker networks
  pull              Pull an image from a registry
  ps                List containers
  push              Push an image to a registry
  rename            Rename a container
  rm                Remove containers
  rmi               Remove images
  run              Run a command in a new container
  save              Save an image to a tar archive
  search            Search Docker Hub for images
  set               Set or clear the default context
  start             Start a container
  stats             Display the real-time statistics for the system's
                                containers
  stop              Stop the container
  system            Manage Docker
  tag               Create a new tag for an image
  top              Display the running processes of a container
  unpause           Unpause a container
  update            Update the configuration of a container
  wait              Wait until a container stops
```

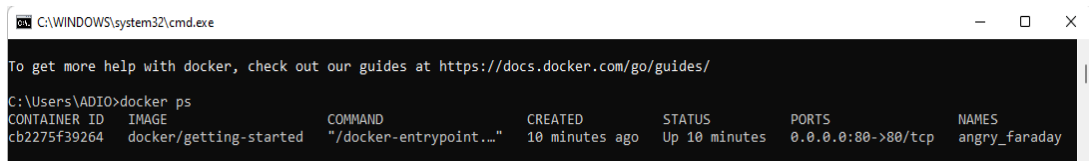
- **docker run**

This command executes a Docker image on your local repo and creates a running container out of it.



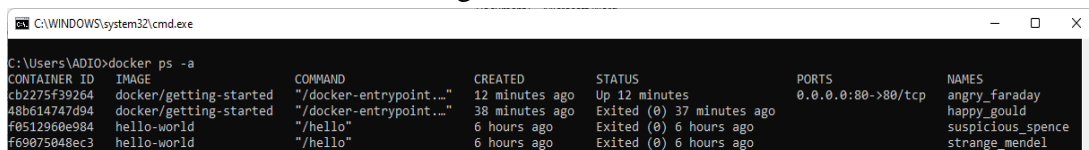
- **docker ps**

This command lists all the running containers in the host.



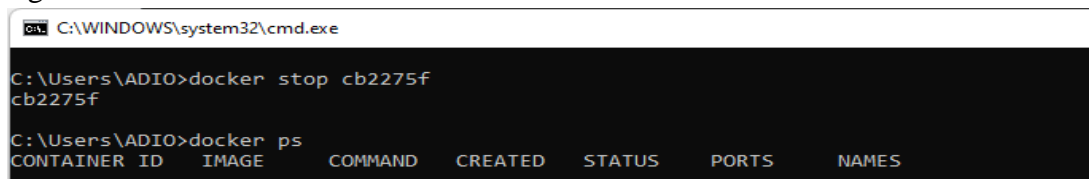
- **docker ps -a**

This command lists both the running and the shut downed containers in the host.



- **docker stop <container-id>**

This command shut down the container whose container-id is specified in the arguments.



- **docker kill <container-id>**

This command kills the container by stopping its execution immediately.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ADIO>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                NAMES
efc61f625b49   docker/getting-started   "/docker-entrypoint..."  34 seconds ago    Up 31 seconds    0.0.0.0:80->80/tcp    wizardly_merkle

C:\Users\ADIO>docker kill efc61f6
efc61f6

C:\Users\ADIO>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                NAMES
C:\Users\ADIO>
```

- **docker rm <container-id>**

This command removes the container whose container-id is specified in arguments.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ADIO>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                NAMES
C:\Users\ADIO>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                NAMES
efc61f625b49   docker/getting-started   "/docker-entrypoint..."  4 minutes ago    Exited (137) 3 minutes ago    wizardly_merkle
cb2275f39264   docker/getting-started   "/docker-entrypoint..."  23 minutes ago    Exited (0) 6 minutes ago    angry_faraday
48b614747d94   docker/getting-started   "/docker-entrypoint..."  48 minutes ago    Exited (0) 48 minutes ago    happy_gould
f851296ae984   hello-world   "/hello"                  6 hours ago      Exited (0) 6 hours ago    suspicious_spence
f69075048ec3   hello-world   "/hello"                  6 hours ago      Exited (0) 6 hours ago    strange_mendel

C:\Users\ADIO>docker rm efc61f
efc61f

C:\Users\ADIO>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                NAMES
cb2275f39264   docker/getting-started   "/docker-entrypoint..."  23 minutes ago    Exited (0) 7 minutes ago    angry_faraday
48b614747d94   docker/getting-started   "/docker-entrypoint..."  49 minutes ago    Exited (0) 48 minutes ago    happy_gould
f851296ae984   hello-world   "/hello"                  6 hours ago      Exited (0) 6 hours ago    suspicious_spence
f69075048ec3   hello-world   "/hello"                  6 hours ago      Exited (0) 6 hours ago    strange_mendel
```

- **docker rmi <image-name>**

This command removes the image whose name is specified in arguments.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ADIO>docker images
REPOSITORY      TAG       IMAGE ID       CREATED        SIZE
ubuntu          latest    216c552ea5ba   7 days ago     77.8MB
docker/getting-started latest    cb90f98fd791   6 months ago   28.8MB
hello-world     latest    feb5d9fea6a5   12 months ago  13.3kB

C:\Users\ADIO>docker rmi ubuntu
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:35fb073f9e56eb84041b0745cb714eff0f7b225ea9e024f703cab56aaa5c7720
Deleted: sha256:216c552ea5ba7b0e3f6e33624e129981c39996021403518019d19b8843c27cbc
Deleted: sha256:17f623af01e277c5ffe6779af8164907de02d9af7a0e161662fc735dd64f117b

C:\Users\ADIO>docker images
REPOSITORY      TAG       IMAGE ID       CREATED        SIZE
docker/getting-started latest    cb90f98fd791   6 months ago   28.8MB
hello-world     latest    feb5d9fea6a5   12 months ago  13.3kB
```

- **docker logs <container-id>**

This will returns the logs of the container whose container-id is specified in arguments.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ADIO>docker logs cb2275
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2022/10/12 11:05:26 [notice] 1#1: using the "epoll" event method
2022/10/12 11:05:26 [notice] 1#1: nginx/1.21.0
2022/10/12 11:05:26 [notice] 1#1: built by gcc 10.3.1 20211027 (Alpine 10.3.1 git20211027)
2022/10/12 11:05:26 [notice] 1#1: OS: Linux 5.10.102.1-microsoft-standard-WSL2
2022/10/12 11:05:26 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2022/10/12 11:05:26 [notice] 1#1: start worker processes
2022/10/12 11:05:26 [notice] 1#1: start worker process 32
2022/10/12 11:05:26 [notice] 1#1: start worker process 33
2022/10/12 11:05:26 [notice] 1#1: start worker process 34
2022/10/12 11:05:26 [notice] 1#1: start worker process 35
2022/10/12 11:05:26 [notice] 1#1: start worker process 36
2022/10/12 11:05:26 [notice] 1#1: start worker process 37
172.17.0.1 - - [12/Oct/2022:11:06:10 +0000] "GET /tutorial/ HTTP/1.1" 304 0 "http://localhost/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36" "-"
172.17.0.1 - - [12/Oct/2022:11:06:18 +0000] "GET /assets/images/favicon.png HTTP/1.1" 200 521 "http://localhost/tutorial/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36" "-"
2022/10/12 11:21:44 [notice] 1#1: signal 3 (SIGQUIT) received, shutting down
2022/10/12 11:21:44 [notice] 3#32: gracefully shutting down
2022/10/12 11:21:44 [notice] 3#34: gracefully shutting down
2022/10/12 11:21:44 [notice] 3#35: gracefully shutting down
```

- **docker start <container-id>**

This command is used to run the container whose container-id is specified in the arguments.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ADIO>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
cb2275f39264   docker/getting-started   "/docker-entrypoint..."   34 minutes ago   Exited (0) 18 minutes ago          angry_faraday
48b614747d94   docker/getting-started   "/docker-entrypoint..."   About an hour ago   Exited (0) 59 minutes ago          happy_gould
f0512960e984   hello-world            "/hello"                   6 hours ago       Exited (0) 6 hours ago             suspicious_spence
f69075048ec3   hello-world            "/hello"                   6 hours ago       Exited (0) 6 hours ago             strange_mendel

C:\Users\ADIO>docker start cb2275
```

- **docker build**

This command is used to compile Dockerfile, for building custom docker images based on the dockerfile.

```
C:\WINDOWS\system32\cmd.exe

(firstdocker) C:\Users\ADIO\Desktop\lNeuron\lNeuron Assignment -2>docker build -t firstfastapiapp .
[+] Building 289.1s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 144B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.8
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load build context
=> => transferring context: 288B
=> CACHED [1/4] FROM docker.io/library/python:3.8@sha256:e0725100b7234ecfd9a685db22c310c8b269231e595b79e5028a655
=> [2/4] COPY . /app
=> [3/4] WORKDIR /app
=> [4/4] RUN pip install -r requirements.txt
=> exporting to image
=> => exporting layers
=> => writing image sha256:73ce528fd9a492d923fc706e30a3c45197e27abd1fef6c5f584d5db1dc6fa24e
=> => naming to docker.io/library/firstfastapiapp
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

- **docker push**

This command is used to push the specified docker image to the docker hub.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ADIO>docker push 8950539357/firstfastapiapp:firstfastapiapp
The push refers to repository [docker.io/8950539357/firstfastapiapp]
dbf8405a9193: Layer already exists
5f70bf18a086: Layer already exists
f7afc6a11679: Layer already exists
08f7737fec66: Layer already exists
dee03037c4fc: Layer already exists
17517a754285: Layer already exists
0c7daf9a72c8: Layer already exists
75ba02937496: Layer already exists
288cf3a46e32: Layer already exists
186da837555d: Layer already exists
955c9335e041: Layer already exists
8e079fee2186: Layer already exists
firstfastapiapp: digest: sha256:ebc56466f1a08319c68da4aa87e9d635ffdf667f6ddc8047f28684de7bd954ea size: 2843
```

Assignment – 2

Hello World Docker Image Run Hello World Docker Image Locally.

- docker pull hello-world

```
CA: Command Prompt

C:\Users\Admin>docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:18a657d0cc1c7d0678a3fba8b7eb4918bba25968d3e1b0adebfa71caddbc346
Status: Downloaded newer image for hello-world:latest
```

- docker run hello-world

```
CA: Command Prompt

C:\Users\Admin>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

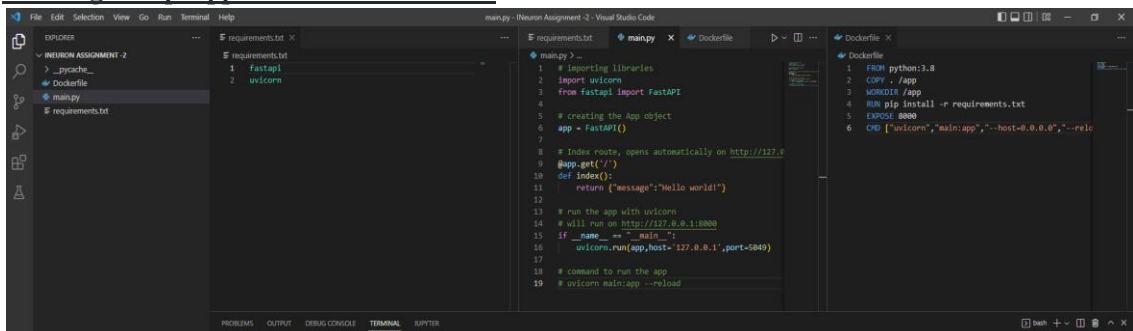
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

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

Assignment – 3

Create a hello world fastapi application. Create a Dockerfile for your fastapi hello world application. Build Docker image using Docker file. Run docker image build in previous step. Push your Docker image to Docker Hub.

- Creating fastapi application and Dockerfile



The screenshot shows a Visual Studio Code editor with three files open: `requirements.txt`, `main.py`, and `Dockerfile`.

`requirements.txt` contains:

```
1 fastapi
2 uvicorn
```

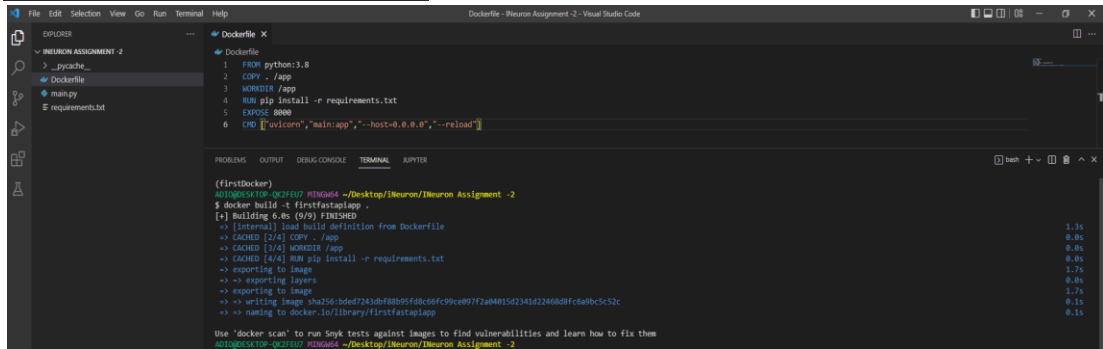
`main.py` contains:

```
1 # Importing libraries
2 import uvicorn
3 from fastapi import FastAPI
4
5 # creating the app object
6 app = FastAPI()
7
8 # Index route, opens automatically on http://127.0.0.1
9 @app.get("/")
10 def index():
11     return {"message": "Hello world!"}
12
13 # run the app with uvicorn
14 # will run on http://127.0.0.1:8000
15 if __name__ == "__main__":
16     uvicorn.run(app, host="127.0.0.1", port=5049)
17
18 # command to run the app
19 # uvicorn main:app --reload
```

`Dockerfile` contains:

```
1 FROM python:3.8
2 COPY ./app
3 WORKDIR /app
4 RUN pip install -r requirements.txt
5 EXPOSE 8000
6 CMD ["uvicorn", "main:app", "--host=0.0.0.0", "--reload"]
```

- **Build Docker image using Docker file**



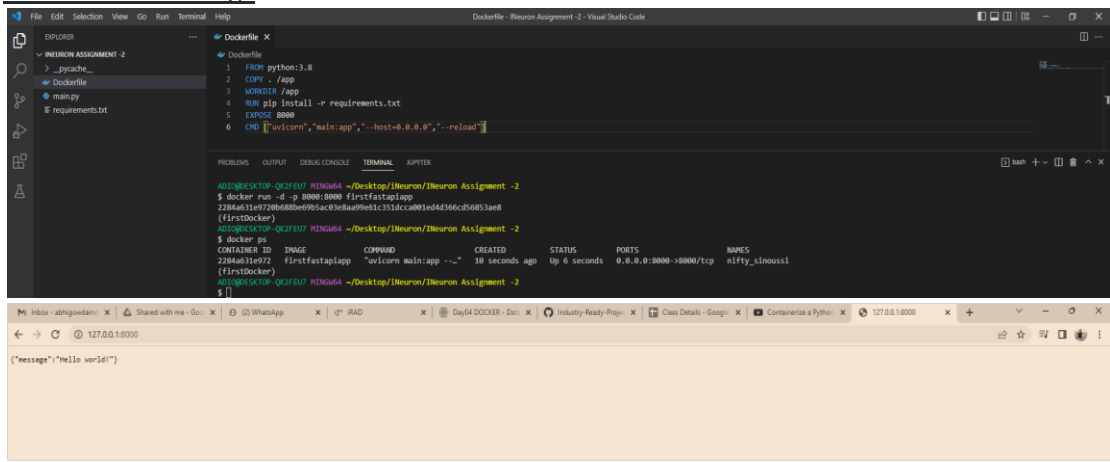
The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Dockerfile contains the following instructions:

```
1 FROM python:3.8
2 COPY . /app
3 WORKDIR /app
4 RUN pip install -r requirements.txt
5 EXPOSE 8000
6 CMD ["uvicorn", "main:app", "--host=0.0.0.0", "--reload"]
```

The terminal output shows the successful build of the Docker image:

```
(firstDocker)
$ docker build -t firstfastapiapp
$ docker build -t firstfastapiapp
[+] Building 6.8s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load .dockerignore
=> [internal] fetch source repository
=> [1/4] COPY . /app
=> [2/4] WORKDIR /app
=> [3/4] RUN pip install -r requirements.txt
=> [4/4] EXPOSE 8000
=> exporting to image
=> exporting layers
=> writing image sha256:b0d7243dbf88b95f8c6d6f9c0e0772a4015d2341d22460d8fca8b5c53c
=> naming to docker.io/library/firstfastapiapp
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
$ docker run --rm -p 8000:8000 firstfastapiapp
```

- **Run the docker image**

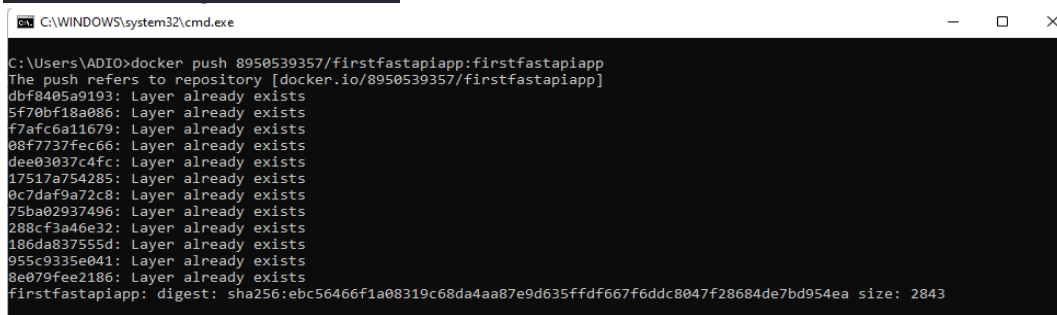


The screenshot shows the Visual Studio Code interface with the Dockerfile open. The terminal output shows the container being run and the application starting:

```
$ docker run -d -p 8000:8000 firstfastapiapp
2284a31e9728b688b6d95ac83b8aa99e61c35dcca0e1ed4d36cd56853ae8
(firstDocker)
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS              PORTS               NAMES
2284a31e9728        firstfastapiapp    "uvicorn main:app --..." 10 seconds ago      Up 6 seconds       0.0.0.0:8000->8000/tcp   nifty_sinoussi
(firstDocker)
$ docker logs -f firstfastapiapp
{"message": "hello world!"}
```

A browser window is open at <http://127.0.0.1:8000>, displaying the output: `{"message": "hello world!"}`.

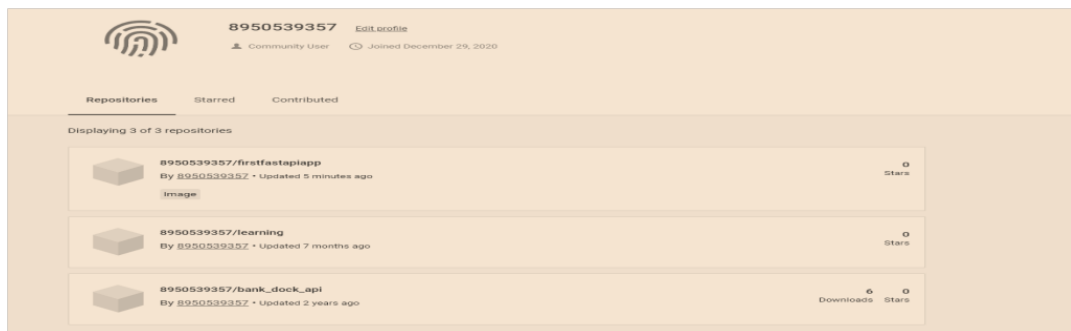
- **Push Docker image to Docker Hub**



The screenshot shows a Windows command prompt window with the following commands and output:

```
C:\WINDOWS\system32\cmd.exe

C:\Users\ADIO>docker push 8950539357/firstfastapiapp:firstfastapiapp
The push refers to repository [docker.io/8950539357/firstfastapiapp]
dbf8405a9103: Layer already exists
5f70bf18a086: Layer already exists
f7afc6a11679: Layer already exists
08f7737fec66: Layer already exists
dee03037c4fc: Layer already exists
17517a754285: Layer already exists
0c7daf9a72c8: Layer already exists
75ba02937496: Layer already exists
288cf3a46e32: Layer already exists
186da837555d: Layer already exists
955c9335e041: Layer already exists
8e079fee2186: Layer already exists
firstfastapiapp: digest: sha256:ebc56466f1a08319c68da4aa87e9d635ffdf667f6ddc8047f28684de7bd954ea size: 2843
```

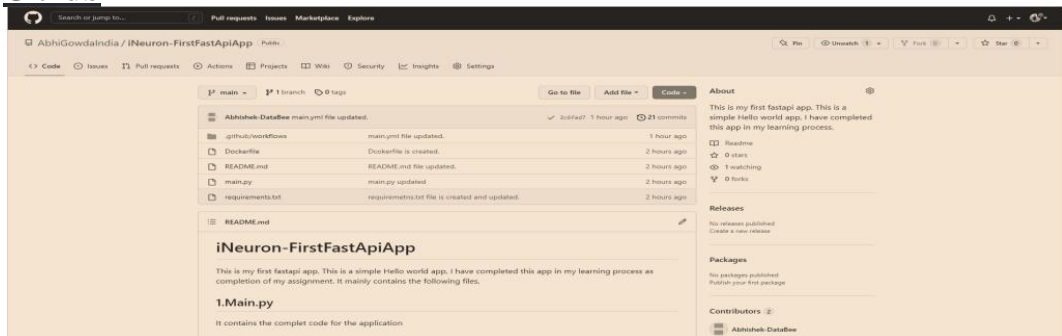


Assignment 4:

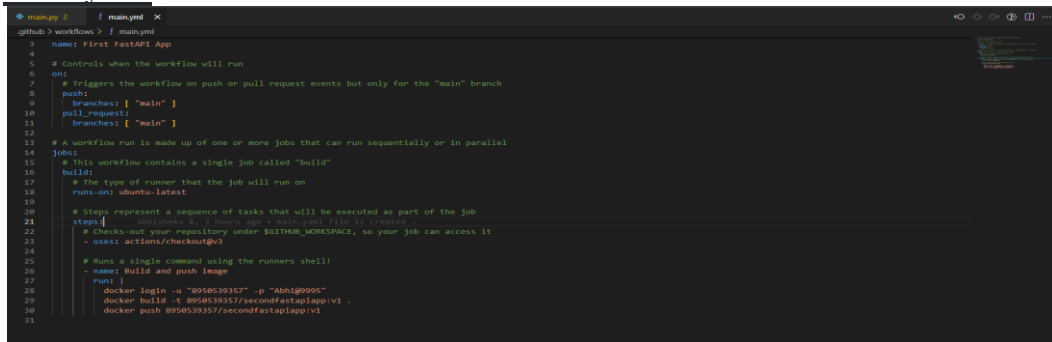
Automate the bellow tasks

- Build docker image, and
 - Push docker image to the docker hub.
- I have automate the above mentioned tasks using github actions.

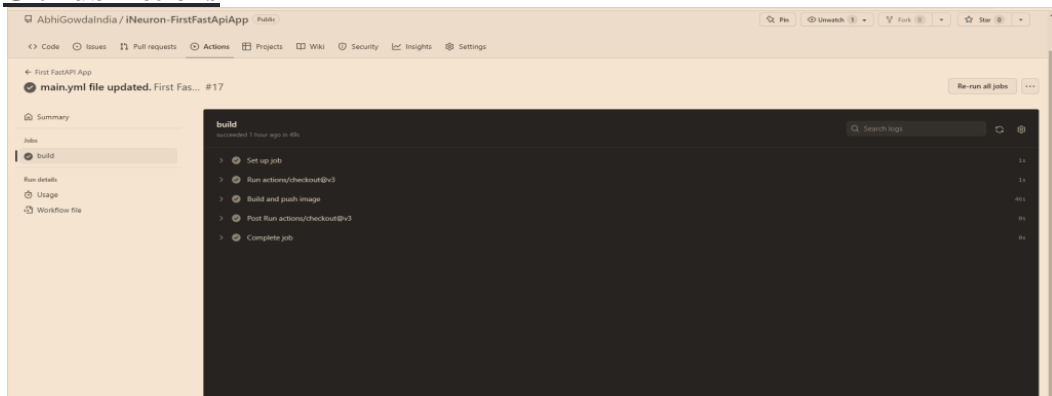
Github



Main.yaml



GitHub Actions



Docker Repository

