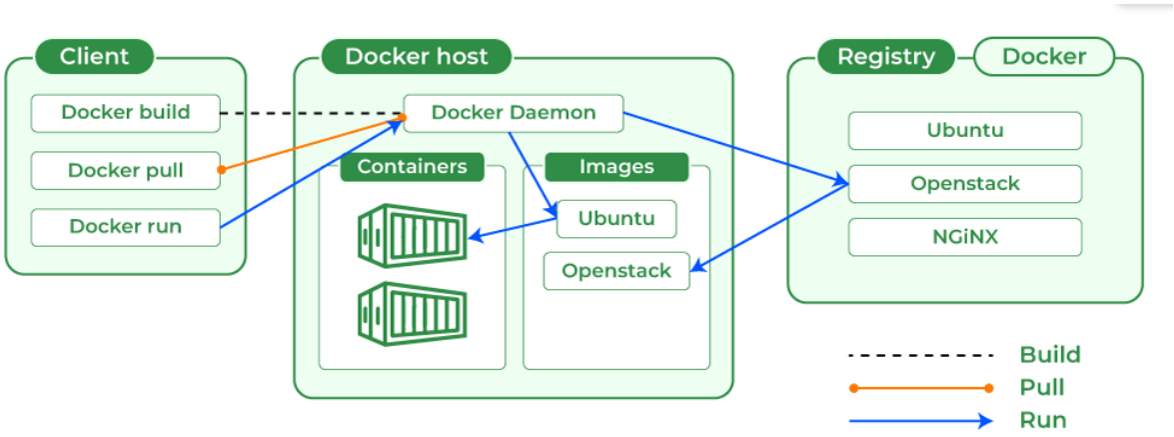
Name: Vaibhav kumar gupta

Date: 24-02-2025

Docker:

Docker is a popular containerization platform that allows developers to package, ship, and run applications in containers. Containers provide a lightweight and portable way to deploy applications, eliminating the need for virtual machines. Docker images are created using a Dockerfile, which specifies the build process and dependencies required by the application. Docker containers can be run on any system that supports Docker, making it easy to deploy applications across different environments. Docker has become a standard tool for modern software development and deployment, especially in cloud-native and DevOps environments.

Architecture of docker:-



Docker Client

With the help of the docker client, the docker users can interact with the docker. The docker command uses the Docker API. The Docker client can communicate with multiple daemons. When a docker client runs any docker command on the docker terminal then the terminal sends instructions to the daemon. The Docker daemon gets those instructions from the docker client withinside the shape of the command and REST API's request.

The main objective of the docker client is to provide a way to direct the pull of images from the docker registry and run them on the docker host. The

common commands which are used by clients are **docker build, docker pull,** and **docker run.**

Docker Host

A Docker host is a type of machine that is responsible for running more than one container. It comprises the Docker daemon, Images, Containers, Networks, and Storage.

Docker Registry

All the docker images are stored in the docker registry. There is a public registry which is known as a **docker hub** that can be used by anyone. We can run our private registry also. With the help of **docker run** or **docker pull** commands, we can pull the required images from our configured registry. Images are pushed into configured registry with the help of the **docker push** command.

How to install docker:-

Note:- remember to do everything in the env so that outside packages wont get interrupted

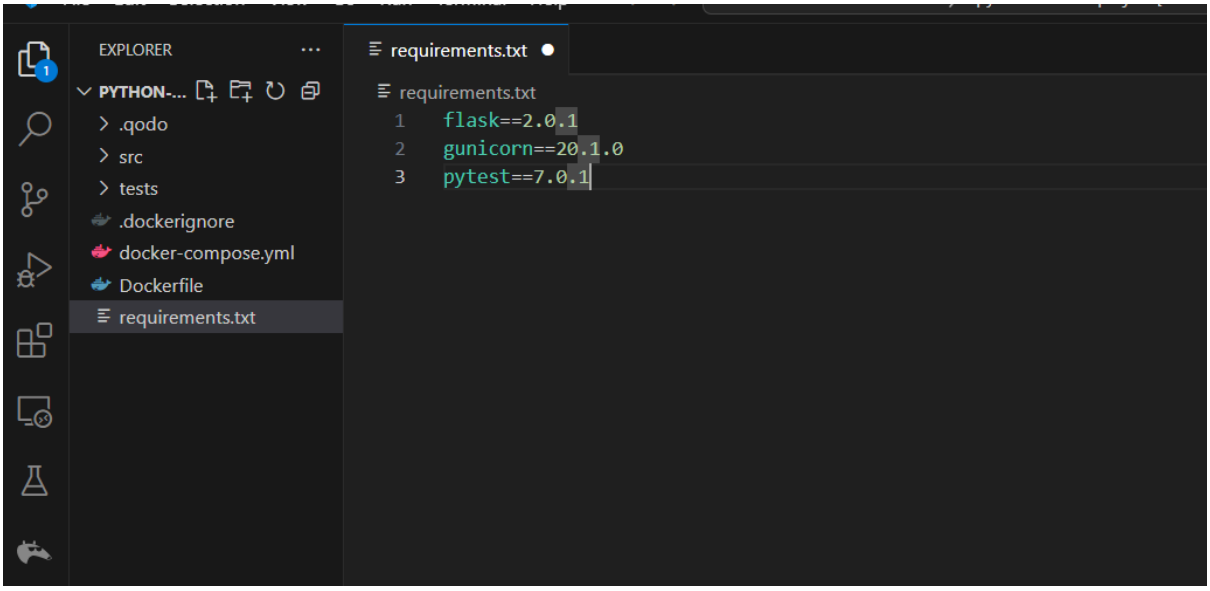
1) Install docker on ubuntu

```
# Update package list
sudo apt update

# Install required dependencies
sudo apt install -y apt-transport-https ca-certificates curl software-properties-common

# Add Docker's GPG key
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/ke

# Add Docker repository
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://downl

# Update package list again
sudo apt update

# Install Docker
sudo apt install -y docker-ce docker-ce-cli containerd.io
```

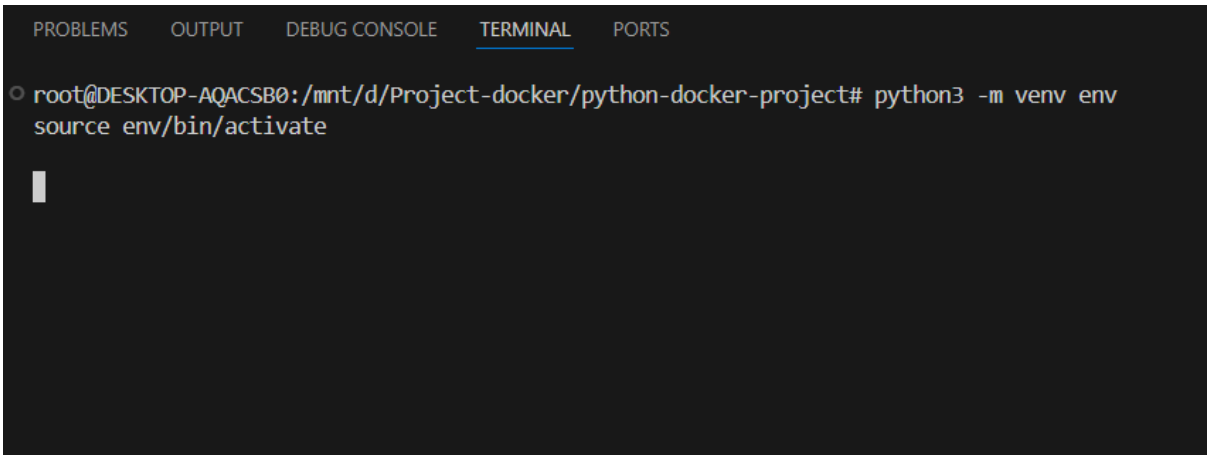
2) Create a directory:-
   mkdir python-docker-project
   cd python-docker-project
   mkdir src tests
   touch src/__init__.py
   touch src/main.py
   touch requirements.txt
   touch Dockerfile
   touch .dockerignore
   touch docker-compose.yml

```
root@DESKTOP-AQACSB0:/mnt/d/Project-docker# mkdir python-docker-project
ython-docker-project
mkdir src tests
touch src/__init__.py
touch src/main.py
touch requirements.txt
touch Dockerfile
touch .dockerignore
touch docker-compose.ymlroot@DESKTOP-AQACSB0:/mnt/d/Project-docker# cd python-docker-project
root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# mkdir src tests
root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# touch src/__init__.py
root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# touch src/main.py
root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# touch requirements.txt
root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# touch Dockerfile
root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# touch .dockerignore
root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# touch docker-compose.yml
root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project#
```

3) Create the rrequirement.txt and add these for flask
flask==2.0.1
gunicorn==20.1.0
pytest==7.0.1



4) Create the environment and run the environment



5) Write the flask code

```python
from flask import Flask,jsonify
app = Flask(__name__)
Qodo Gen: Options | Test this function
@app.route("/health")
def health_check():
    return jsonify({"status": "healthy"})
Qodo Gen: Options | Test this function
@app.route("/")
def hello_world():
    return jsonify({"message":"Hello from Jinesh teaching docker"})
if __name__ == '__main__':
    app.run(host='0.0.0.0',port=5000)
```

6) Install all the requirements.txt

```
(env) root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# pip list
Package      Version
------------ -------
attrs        25.1.0
click        8.1.8
Flask        2.0.1
gunicorn     20.1.0
iniconfig    2.0.0
itsdangerous 2.2.0
Jinja2       3.1.5
MarkupSafe   3.0.2
packaging    24.2
pip          24.0
pluggy       1.5.0
py           1.11.0
pytest       7.0.1
setuptools   75.8.0
tomli        2.2.1
Werkzeug     3.1.3
(env) root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project#
```

7) Try running the flask:

```
(env) root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# python3 src/main.py
 * Serving Flask app 'main'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:50001
 * Running on http://172.23.134.122:50001
Press CTRL+C to quit
172.23.128.1 - - [24/Feb/2025 14:30:25] "GET / HTTP/1.1" 200 -
172.23.128.1 - - [24/Feb/2025 14:30:27] "GET /meta.json HTTP/1.1" 404 -
172.23.128.1 - - [24/Feb/2025 14:30:27] "GET / HTTP/1.1" 200 -
172.23.128.1 - - [24/Feb/2025 14:30:27] "GET /meta.json HTTP/1.1" 404 -
172.23.128.1 - - [24/Feb/2025 14:30:27] "GET /favicon.ico HTTP/1.1" 404 -
172.23.128.1 - - [24/Feb/2025 14:30:31] "GET /health HTTP/1.1" 200 -
172.23.128.1 - - [24/Feb/2025 14:30:31] "GET / HTTP/1.1" 200 -
172.23.128.1 - - [24/Feb/2025 14:30:32] "GET /meta.json HTTP/1.1" 404 -
```

8) Write the docker code

```
FROM python:3.9-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
ENV FLASK_APP=src/main.py
ENV FLASK_ENV=development
ENV PYTHONPATH=/app
EXPOSE 5000
CMD ["gunicorn","--bind","0.0.0.0:5000","src.main:app"]
```

9) Build the docker image

docker build -t python-docker-app .

here -t is the flag and python-docker-app is the name

```
(env) root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# docker build -t az .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  7.168kB
Step 1/10 : FROM python:3.9-slim
 ---> 096343841dd9
Step 2/10 : WORKDIR /app
 ---> Using cache
 ---> 2304f89941be
Step 3/10 : COPY requirements.txt .
 ---> Using cache
 ---> 54eef22ee3e0
Step 4/10 : RUN pip install --no-cache-dir -r requirements.txt
 ---> Using cache
 ---> c2d0a80762d0
Step 5/10 : COPY . .
 ---> 5071b074b277
Step 6/10 : ENV FLASK_APP=src/main.py
 ---> Running in f25d78d9bd13
 ---> Removed intermediate container f25d78d9bd13
 ---> 4978ea8ee2de
Step 7/10 : ENV FLASK_ENV=development
 ---> Running in d11d953cb8ae
 ---> Removed intermediate container d11d953cb8ae
 ---> 536b7f7a35b4
Step 8/10 : ENV PYTHONPATH=/app
 ---> Running in 717aecc1e442
 ---> Removed intermediate container 717aecc1e442
 ---> d5597691cc98
Step 9/10 : EXPOSE 5000
```

10) Run the docker image :

Docker run app_name

Or docker run -p 5000:5000 app-name

```
(env) root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# docker run az
[2025-02-24 14:35:00 +0000] [1] [INFO] Starting gunicorn 23.0.0
[2025-02-24 14:35:00 +0000] [1] [INFO] Listening at: http://0.0.0.0:5000 (1)
[2025-02-24 14:35:00 +0000] [1] [INFO] Using worker: sync
[2025-02-24 14:35:00 +0000] [7] [INFO] Booting worker with pid: 7
[2025-02-24 14:35:55 +0000] [1] [INFO] Handling signal: winch
[2025-02-24 14:35:56 +0000] [1] [INFO] Handling signal: winch
[2025-02-24 14:35:57 +0000] [1] [INFO] Handling signal: winch
[2025-02-24 14:35:57 +0000] [1] [INFO] Handling signal: winch
[2025-02-24 14:35:57 +0000] [1] [INFO] Handling signal: winch
[2025-02-24 14:35:57 +0000] [1] [INFO] Handling signal: winch
[2025-02-24 14:35:57 +0000] [1] [INFO] Handling signal: winch
```

11) Check using the localhost port number mentioned in the docker file

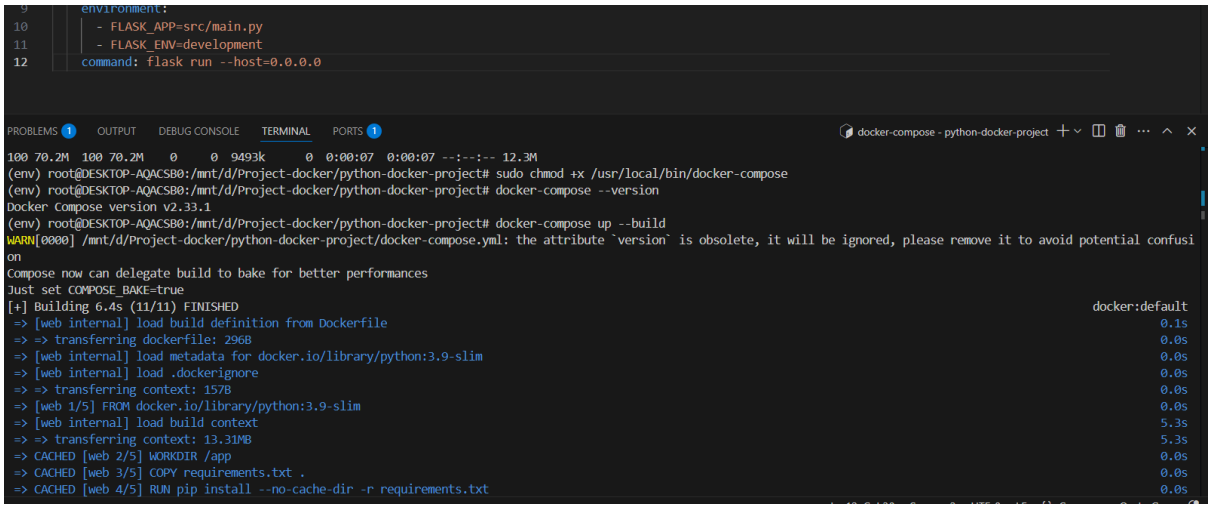
12) Create the docker compose and write the code in it

```
version: '3.8'
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - .:/app
    environment:
      - FLASK_APP=src/main.py
      - FLASK_ENV=development
    command: flask run --host=0.0.0.0
```

Install Docker Compose
sudo curl -L
"https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

Make Docker Compose Executable
sudo chmod +x /usr/local/bin/docker-compose

Build & Start Containers
docker-compose up –build



Connect to docker and push it:-
Login docker in the ubuntu

Docker images to check the images

Tag your image :
docker tag 8067583e48e0 vaibhav3746/python-docker-project-web:latest

push:
docker push vaibhav3746/python-docker-project-web:latest

```
root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# docker images
REPOSITORY                                          TAG        IMAGE ID       CREATED          SIZE
vaibhav3746/python-docker-project-web               latest     8067583e48e0   3 minutes ago    153MB
your-dockerhub-username/python-docker-project-web   latest     8067583e48e0   3 minutes ago    153MB
python-docker-project-web                           latest     322a942e143b   9 minutes ago    153MB
x                                                   latest     21c59424943c   23 minutes ago   153MB
az                                                  latest     5173ccf977fb   42 minutes ago   139MB
ab                                                  latest     9014366ede89   4 hours ago      139MB
a                                                   latest     9014366ede89   4 hours ago      139MB
vaibhav3746/python-docker-project-web               v1         51c787895c0c   8 hours ago      139MB
python-docker-app                                   latest     b597336cab7f   9 hours ago      139MB
vaibhav3746/python-docker-app                       v1         b597336cab7f   9 hours ago      139MB
python                                              3.9-slim   096343841dd9   2 months ago     126MB
root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project# docker push vaibhav3746/python-docker-project-web:latest
The push refers to repository [docker.io/vaibhav3746/python-docker-project-web]
e43d73511eb2: Pushed
db0583aad182: Pushed
86d00814a7d1: Pushed
19b580c21171: Pushed
6022e9b5727d: Pushed
e0dfbff797f9: Mounted from library/python
0eaf13317391: Mounted from library/python
7914c8f600f5: Mounted from library/python
latest: digest: sha256:5cdaf41261bd728f4b7e4e16364c4ba180a2ee29f1fb914b4ee125199869e91d size: 1994
root@DESKTOP-AQACSB0:/mnt/d/Project-docker/python-docker-project#
```