**📁 Project Structure**

socket\_docker/

├── docker-compose.yml

├── server/

│ ├── Dockerfile

│ └── server.py

└── client/

├── Dockerfile

└── client.py

# server/server.py

import socket

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

host = '0.0.0.0' # Accept connections from any IP

port = 12345

server\_socket.bind((host, port))

server\_socket.listen(1)

print(f"[SERVER] Listening on {host}:{port}...")

client\_socket, addr = server\_socket.accept()

print(f"[SERVER] Connected to client at {addr}")

while True:

data = client\_socket.recv(1024).decode()

if not data or data.lower() == 'exit':

print("[SERVER] Connection closed.")

break

print(f"[CLIENT] {data}")

reply = input("[YOU] Enter response: ")

client\_socket.send(reply.encode())

client\_socket.close()

server\_socket.close()

# client/client.py

import socket

client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

host = 'server' # Docker Compose service name for the server

port = 12345

client\_socket.connect((host, port))

print(f"[CLIENT] Connected to server at {host}:{port}")

while True:

message = input("[YOU] Enter message: ")

client\_socket.send(message.encode())

if message.lower() == 'exit':

break

response = client\_socket.recv(1024).decode()

print(f"[SERVER] {response}")

client\_socket.close()

# server/Dockerfile

FROM python:3.10

WORKDIR /app

COPY server.py .

CMD ["python", "server.py"]

# client/Dockerfile

FROM python:3.10

WORKDIR /app

COPY client.py .

CMD ["python", "client.py"]

# docker-compose.yml

version: '3.8'

services:

server:

build: ./server

container\_name: python\_socket\_server

ports:

- "12345:12345"

stdin\_open: true

tty: true

client:

build: ./client

container\_name: python\_socket\_client

depends\_on:

- server

stdin\_open: true

tty: true

**🚀 HOW TO RUN**

1. Create the folder structure and save each file accordingly.
2. From the socket\_docker/ directory, build and run containers:

docker-compose up --build

1. In two separate terminals, attach to the containers:

docker attach python\_socket\_server

docker attach python\_socket\_client

1. Type messages interactively. Type exit to end the connection.