

Assignment 06

HPCSA 0824 Cloud & Security Lab Assignment 6 1.

Q1

Create a python program that takes 2 numbers from the user and displays addition and multiplication result. Now create an image of this program. Run the container from this image and display the results.

Mkdir app

cd app

Product.py

sudo nano product.py

```
# Python Program to add and multiply two numbers
def main():
    num1 = float(input("Enter the first number: "))
    num2 = float(input("Enter the second number: "))

    addition_result = num1 + num2
    multiplication_result = num1 * num2

    print(f"The sum of {num1} and {num2} is: {addition_result}")
    print(f"The product of {num1} and {num2} is: {multiplication_result}")

if __name__ == "__main__":
    main()
```

DockerFile

sudo Dockerfile

```
GNU nano 6.2 Dockerfile
# Use an official Python runtime as a base image
FROM python:3.9-slim

# Set the working directory inside the container
WORKDIR /app

# Copy the Python script into the container
COPY Product.py /app/Product.py

# Install any dependencies (if needed)
# RUN pip install --no-cache-dir <package-name>

# Set the command to run the Python program
CMD ["python", "Product.py"]
```

Step 3: Build the Docker Image

docker build -t python-add-multiply .

check

Sudo hduser@suraj:~/app\$ sudo docker images ls

Step 4: Run the Docker Container

sudo docker run -it python-add-multiply

```
hduser@suraj:~/app$ ^C
hduser@suraj:~/app$ docker run -it python-add-multiply
docker: permission denied while trying to connect to the Docker dae
mon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Fru
n%2Fdocker.sock/v1.24/containers/create": dial unix /var/run/docker
.sock: connect: permission denied.
See 'docker run --help'.
hduser@suraj:~/app$ sudo docker run -it python-add-multiply.0
Unable to find image 'python-add-multiply.0:latest' locally
^[[Adocker: Error response from daemon: pull access denied for pyth
on-add-multiply.0, repository does not exist or may require 'docker
login': denied: requested access to the resource is denied.
See 'docker run --help'.
hduser@suraj:~/app$ sudo docker run -it python-add-multiply.
docker: invalid reference format.
See 'docker run --help'.
hduser@suraj:~/app$ sudo docker run -it python-add-multiply
Enter the first number: 10
Enter the second number: 20
The sum of 10.0 and 20.0 is: 30.0
The product of 10.0 and 20.0 is: 200.0
hduser@suraj:~/app$ sudo docker run -it python-add-multiply
Enter the first number: 
```

Q2

Create a directory. Create an index.html file. Create a image using httpd image. Copy index.html inside the container. Run the container and map port 8200 using your image and display the webpage is displayed.

mkdir my-webpage

Create a directory called "my-webpage"

cd my-webpage

Navigate into the "my-webpage" directory

nano index.html

Open the "index.html" file in nano editor to create/edit it

sudo docker run -d -p 8200:80 --name my-webpage-container httpd

Run a Docker container with Apache (httpd), map port 8200 to 80, and name it "my-webpage-container"

sudo docker ps

List running Docker containers to confirm the container is running

sudo docker cp index.html my-webpage-container:/usr/local/apache2/htdocs/index.html

Copy "index.html" to the container's web directory

curl my-webpage-container

Try to use curl to access the container

```

248 docker build -t python-add-multiply
249 docker run -it python-add-multiply
250 sudo docker run -it python-add-multiply
251 sudo docker image ls
252 docker build -t python-add-multiply .
253 sudo docker build -t python-add-multiply .
254 nano Dockerfile
255 sudo docker build -t python-add-multiply .
256 sudo docker images ls
257 sudo docker images
258 docker run -it python-add-multiply
259 sudo docker run -it python-add-multiply.0
260 sudo docker run -it python-add-multiply.
261 sudo docker run -it python-add-multiply
262 cd
263 sudo docker ps -a
264 sudo docker rm b8c30d1bf6dc 0b99d214580d
265 mkdir my-webpage
266 cd my-webpage
267 nano index.html
268 sudo docker run -d -p 8200:80 --name my-webpage-container ht
tpd
269 sudo docker ps
270 sudo docker cp index.html my-webpage-container:/usr/local/ap
ache2/htdocs/index.html
271 curl my-webpage-container
272 history
nduser@suraj:~/my-webpage$ █

```



Q3

Create a image of the test.py file. Create a network by name app2. Create a MYSQL container and connect it to this network. Now create a container of test.py application and connect it to the same network and display that the database is created.

```
# Create a custom network
Sudo docker network create app2
```

Run a MySQL container connected to the app2 network

```
Sudo docker run --name mysql-container --network app2 -e
MYSQL_ROOT_PASSWORD=root -e MYSQL_DATABASE=testdb -d
mysql:latest
```

Dockerfile

```
FROM python:3.9-slim
WORKDIR /app
COPY . /app
RUN pip install mysql-connector-python
CMD ["python", "test.py"]
```

test.py

```
import mysql.connector
import os

MYSQL_HOST = os.getenv("MYSQL_HOST", "mysql-container")
# Get the MySQL host from environment variables or use the default "mysql-container"

MYSQL_USER = os.getenv("MYSQL_USER", "root")
# Get the MySQL user from environment variables or use the default "root"

MYSQL_PASSWORD = os.getenv("MYSQL_PASSWORD", "root")
# Get the MySQL password from environment variables or use the default "root"

MYSQL_DB = os.getenv("MYSQL_DB", "testdb")
# Get the MySQL database name from environment variables or use the default "testdb"

connection = mysql.connector.connect(
    host=MYSQL_HOST,
    user=MYSQL_USER,
    password=MYSQL_PASSWORD,
    database=MYSQL_DB
)
# Connect to the MySQL server using the provided credentials and database
```

```

cursor = connection.cursor()
# Create a cursor to execute SQL queries

cursor.execute("SHOW DATABASES LIKE %s", (MYSQL_DB,))
# Query to check if the specified database exists

db = cursor.fetchone()
# Fetch the result of the query, will return None if no match is found

if db:
    print(f'Database '{MYSQL_DB}' exists!')
    # If the database exists, print a success message

else:
    print(f'Database '{MYSQL_DB}' does not exist!')
    # If the database doesn't exist, print an error message

cursor.close()
# Close the cursor to release database resources

connection.close()
# Close the database connection to clean up

```

3. Build and Run the Python App Container

```

# Build the Python app Docker image
sudo docker build -t test-app .

# Run the Python app container connected to the same network

sudo docker run --name test-app-container --network app2 -d test-app

```

4. Checking the Logs

```

sudo docker logs test-app-container

```

```

Successfully built d2606ca2f2fc
Successfully tagged test-app:latest
hduser@suraj:~$ sudo docker run --name test-app-container --network
app2 -d test-app
805175c15ae55a2a06437e5a2b11a02d8ad373e514eccfcf9f0b21a66312b79e
hduser@suraj:~$ docker logs test-app-container
permission denied while trying to connect to the Docker daemon sock
et at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocke
r.sock/v1.24/containers/test-app-container/json": dial unix /var/ru
n/docker.sock: connect: permission denied
hduser@suraj:~$ sudo logs test-app-container
sudo: logs: command not found
hduser@suraj:~$ sudo docker logs test-app-container
Traceback (most recent call last):
  File "/app/test.py", line 45, in <module>
    Database 'testdb' exists!
    kkkkkkkkkkkkkkkkkkkk
NameError: name 'kkkkkkkkkkkkkkkkkkk' is not defined
hduser@suraj:~$ █

```

Q 4.

Create a docker compose file to start 4 containers using httpd image. Map ports 5000, 6000, 7000 and 8000. Port 8000 website should display “Welcome to ACTS”. Port 5000 should display “Welcome to HPCSA”. Port 6000 should display “Welcome to DITISS” and port 7000 should display “Welcome to DAI”. Use docker-compose command.

```

mkdir myproject
213 sudo nano compose.yml
214 mkdir web_content
215 mv compose.yml myproject
216 cd web_content
217 mkdir hpcsa
218 cd hpcsa
219 mkdir index.html
220 nano index.html
221 ls
222 sudo rm -rm index.html
223 sudo rm -rf index.html
224 nano index.html
225 cd .
226 mkdir ditiss
227 cd ditiss
228 index.html
229 nano index.html

```

```
230 cd .
231 clear
232 ls
233 cd
234 ls
235 sudo rm -rf myproject
236 ls
237 clear
238 mkdir myproject/
239 cd myproject/
240 sudo nano docker-compose.yml
241 mkdir web_content
242 cd web_content/
243 mkdir hpcsa
244 cd hpcsa
245 sudo nano index.html
246 cd ..
247 mkdir ditiss
248 cd ditiss/
249 sudo nano index.html
250 cd ..
251 mkdir dai
252 cd dai
253 sudo index.html
254 sudo nano index.html
255 cd ..
256 mkdir acts
257 cd acts
258 sudo nano index.html
259 docker-compose up -d
260 snap install docker
261 sud snap install docker
262 sudo apt install docker-compose
docker-compose up -d
```

```
myproject/
├─ docker-compose.yml
├─ web_content/
│   └─ hpcsa/
│       └─ index.html
│   └─ ditiss/
│       └─ index.html
│   └─ dai/
│       └─ index.html
│   └─ acts/
│       └─ index.html
```


Yaml file

version: '3.8'

services:

hpcsa:

image: httpd:latest

container_name: hpcsa

ports:

- "5000:80"

volumes:

- ./web_content/hpcsa:/usr/local/apache2/htdocs/

ditiss:

image: httpd:latest

container_name: ditiss

ports:

- "6000:80"

volumes:

- ./web_content/ditiss:/usr/local/apache2/htdocs/

dai:

image: httpd:latest

container_name: dai

ports:

- "7000:80"

volumes:

- ./web_content/dai:/usr/local/apache2/htdocs/

acts:

image: httpd:latest

container_name: acts

ports:

- "8000:80"

volumes:

- ./web_content/acts:/usr/local/apache2/htdocs/

```
249 sudo nano index.html
250 cd ..
251 mkdir dai
252 cd dai
253 sudo index.html
254 sudo nano index.html
255 cd ..
256 mkdir acts
257 cd acts
258 sudo nano index.html
259 docker-compose up -d
260 snap install docker
261 sud snap install docker
262 sudo apt install docker-compose
263 history
hduser@suraj:~/myproject/web_content/acts$ ^C
hduser@suraj:~/myproject/web_content/acts$ sudo docker-compose up -d
Creating network "myproject_default" with the default driver
Creating ditiss ... done
Creating dai ... done
Creating hpcsa ... done
Creating acts ... done
hduser@suraj:~/myproject/web_content/acts$
```

