

Docker Assignment

Sara chatraei _motahare Raofimanesh

Problem 1: PostgreSQL

Step 1: Create a Docker Volume for Persistent Storage

To ensure that PostgreSQL data persists even after container restarts, we will create a Docker volume. The volume will be mounted to the PostgreSQL container to store its data.

```
C:\Users\Sara.CH>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
C:\Users\Sara.CH>docker volume create postgres_data
postgres_data
```

Step 2: Run PostgreSQL in a Docker Container

Now that we have a volume for persistent storage, we will run the PostgreSQL container. The following command will set up a PostgreSQL container with environment variables for user credentials, expose the necessary port, and mount the volume for persistent data storage.

```
C:\Users\Sara.CH>docker run --name my_postgres -e POSTGRES_USER=Sara -e POSTGRES_PASSWORD=fghjkl;' -p 5432:5432 -v postgres_data:/var/lib/postgresql/data -d postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
254e724d7786: Pull complete
5ed504ed4ebe: Pull complete
6b1012ab103c: Pull complete
688bd672efd5: Pull complete
18c4f6c1b1e7: Pull complete
5b2c2f1196a6: Pull complete
451329d20b9f: Pull complete
33518fbc18a9: Pull complete
8ac7e820633f: Pull complete
f17c4120c789: Pull complete
8937a896f225: Pull complete
d65a61d71ceb: Pull complete
13b75ce6e0b4: Pull complete
bb9034c7832a: Pull complete
Digest: sha256:304ab813518754228f9f792f79d6da36359b82d8ecf418096c636725f8c930ad
Status: Downloaded newer image for postgres:latest
1de2278810a4472bed260426a581f0ef4b252643ce17400cc0fa551625740e46
```

Step3: Connect to PostgreSQL

Once the container is running, we need to connect to PostgreSQL using the following command:

```
C:\Users\Sara.CH>docker exec -it my_postgres psql -U Sara
psql (17.4 (Debian 17.4-1.pgdg120+2))
Type "help" for help.
```

Step 4: Create a Database and Table and Insert Data

Inside the PostgreSQL interactive session, create a new database and a table to store team information for the CTF competition

```
C:\Users\Sara.CH>docker exec -it my_postgres psql -U Sara -d ctf_database
psql (17.4 (Debian 17.4-1.pgdg120+2))
Type "help" for help.

ctf_database=# SELECT current_database();
 current_database
-----
 ctf_database
(1 row)

ctf_database=# \dt
Did not find any relations.
ctf_database=# CREATE TABLE teams (
    id SERIAL PRIMARY KEY,
    team_name VARCHAR(100),
    challenge_name VARCHAR(100)
);
CREATE TABLE
ctf_database=# INSERT INTO teams (team_name, challenge_name) VALUES ('Team sara',
'Java To-Do Challenge');
INSERT INTO teams (team_name, challenge_name) VALUES ('Team moti', 'OWASP Juice Sh
op');
INSERT 0 1
INSERT 0 1
```

Step 5: Verify Data Insertion

To verify that the data has been inserted correctly, run the following query:

```
ctf_database=# SELECT * FROM teams;
 id | team_name | challenge_name
----+-----+-----
  1 | Team sara | Java To-Do Challenge
  2 | Team moti | OWASP Juice Shop
(2 rows)
```

Step 6: Test Data Persistence

The next step is to test whether the data persists even after stopping and removing the container. Follow these steps:

```
C:\Users\Sara.CH>docker stop my_postgres
my_postgres

C:\Users\Sara.CH>docker rm my_postgres
my_postgres

C:\Users\Sara.CH>docker run --name my_postgres -e POSTGRES_USER=Sara -e POSTGRES_PASSWORD=fghjkl;' -p 5432:5432 -v postgres_data:/var/lib/postgresql/data -d postgres
ce225354dbb8880c524a6937b9afebb08ba08ddfc922c7b6bb796f5eb4800eeb
```