Exercise Worksheet

[www.vomtom.at](http://www.vomtom.at)

# From the Course: Understanding Docker Run, Dockerfile, Docker-Compose for Beginners

## Databases and Data Persistence in Host-Volume Mounted Directories in Docker Step by Step

In the previous example we were spinning up our database with our web-app. But docker containers are ephemeral. This means they are losing all data once removed and re-started. So, somehow, we must find a way to make data persistent if we want to keep it.

Use this docker-compose.yml file:



* MYSQL\_DATABASE will create an empty database with the name “somedatabase” at first spin-up
* depends\_on waits for the container to start on the other containers

Run the command:

docker-compose up -d

* starts both containers in detached mode

docker-compose ps

* will show the “db” running, while the other one stopped

docker-compose run --rm dbclient

* should connect to the “db” and open a mariadb shell

Enter the following SQL queries:

USE somedatabase;

SHOW TABLES;

Observe that it’s empty!

CREATE TABLE mytable (id INT);

SHOW TABLES;

Now you have one table in your database, which we created.

Exit the console with

exit;

Now stop the container:

docker-compose stop

and remove the containers

docker-compose rm

* removes all (not running) containers from docker-compose

docker-compose up -d

* Re-Run the containers based on the images
* Only the database container is running

Login to the mysql shell again:

docker-compose run --rm dbclient

USE somedatabase;

SHOW TABLES;

Observe it’s empty again.

exit;

docker-compose stop

docker-compose rm

* Stop and remove the containers before we continue

So, how can we make data persistent, even we remove the containers? With volumes and a host-mounted data directory.

mkdir data

* Create a new “data” directory on the host

And add a volumes key to your docker-compose.yml file:



Now start the db service:

docker-compose db -d up

* Observe the data directory, it gets populated with data

We can now also enter the container. Recreate the table:

docker-compose run --rm dbclient

USE somedatabase;

SHOW TABLES;

Then create the table, because it is empty right now.

CREATE TABLE mytable (id INT);

SHOW TABLES;

And then exit and stop and remove the container:

exit;

docker-compose stop

docker-compose rm

Now spin up the db container again:

docker-compose up -d db

Then login to the mysql client:

docker-compose run --rm dbclient

USE somedatabase;

SHOW TABLES;

Observe the database is now persistent. But maybe writing on the host into a directory is not the best solution in this case. Maybe a named volume would be better…