

Data management in Docker: solution to the lab challenge

Goal: create a service based on two containers:

- WordPress (as CMS)
- MariaDB (as DBMS)

and then use your web browser to access WordPress on port 80.

Create a volume `db_data` to provide persistent storage for the DBMS

```
docker volume create db_data
```

Launch MariaDB (image name: `mariadb:10.6.4-focal`) with container name `db`, using the previously created volume, and pass environment variables to the container to configure it

```
docker container run -d \  
--name db \  
--mount type=volume,source=db_data,destination=/var/lib/mysql \  
-e MYSQL_ROOT_PASSWORD=difficultrootpassword \  
-e MYSQL_DATABASE=wordpress-db \  
-e MYSQL_USER=wordpress-user \  
-e MYSQL_PASSWORD=wordpress-password \  
mariadb:10.6.4-focal
```

Inspect the MariaDB container finding its private IP: you can look for the value of `IPAddress` in the output of the `docker inspect db` command, or parse the output using `jq` (a tool to inspect/manipulate JSON data; if it's not installed, install it with `sudo apt install jq`), putting the IP address in a environment variable

```
DB_CONTAINER_IP=$(docker inspect db | jq -r '[] | .NetworkSettings |  
.Networks | .bridge | .IPAddress')
```

Launch WordPress (image name: `wordpress:latest`) with container name `wordpress`, and pass environment variables to the container to configure it (using as DB host the IP of the MariaDB container)

```
docker container run -d \  
--name wordpress \  
--publish 80:80 \  
-e WORDPRESS_DB_HOST=$DB_CONTAINER_IP \  
-e WORDPRESS_DB_USER=wordpress-user \  
wordpress:latest
```

```
-e WORDPRESS_DB_PASSWORD=wordpress-password \  
-e WORDPRESS_DB_NAME=wordpress-db \  
wordpress:latest
```

Find the IP address of your VM (if you don't know it) executing `hostname -I`, then open your browser, paste the IP address in the URL/search bar and press return: you should see the Wordpress installation page.