Networking for Docker containers: solution to the hands-on lab challenge

Goal: create a service based on two containers:

- WordPress
- MariaDB

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
```

Create a network

```
docker network create wordpress-network
```

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 \
--net wordpress-network \
--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
```

No need to inspect the MariaDB container to find its private IP

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 \
--net wordpress-network \
--publish 80:80 \
-e WORDPRESS_DB_HOST=db \
-e WORDPRESS_DB_USER=wordpress-user \
-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.