Full Stack Development
Lab Docker 6

Docker Compose

1. Lab objectives

In this lab you will create create and run a two container application using docker compose. The application is a WordPress blog running in one container and a Maria-db database running in another container. The persistent data is stored in two volumes that compose application creates automatically.

2. Download the Docker Compose File

The docker compose file is in the labs directory for the docker module.

```
services:

 db:
  # We use a mariadb image which supports both amd64 & arm64 architecture
  image: mariadb:10.6.4-focal
  # If you really want to use MySQL, uncomment the following line
  #image: mysql:8.0.27
  command: '--default-authentication-plugin=mysql_native_password'
  volumes:
   - db_data:/var/lib/mysql
  restart: always
  environment:
   - MYSQL_ROOT_PASSWORD=somewordpress
   - MYSQL_DATABASE=wordpress
   - MYSQL_USER=wordpress
   - MYSQL_PASSWORD=wordpress
  expose:
   - 3306
   - 33060
 wordpress:
  image: wordpress:latest
  volumes:
   - wp_data:/var/www/html
  ports:
   - 80:80
  restart: always
  environment:
   - WORDPRESS_DB_HOST=db
   - WORDPRESS_DB_USER=wordpress
   - WORDPRESS_DB_PASSWORD=wordpress
   - WORDPRESS_DB_NAME=wordpress
volumes:
 db_data:
 wp_data:
```

3. Preparation

First check to see that there are no volumes existing. If there are, delete the volumes so that you can see the auto-creation of the volumes. Also, remove any networks that have been created by you so that your environment looks like this.

```
D:\docker>docker volume ls
DRIVER    VOLUME NAME

D:\docker>docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
e056bbfa7d6e   bridge    bridge    local
74dbc19ddc61   host      host      local
b0a01d01f8a8   none      null      local
```

4. Run the app.

All that you have to do to run the application is to use the “docker compose up” command in the directory that contains the docker-compose.yaml file. You should see something like this (you may see a bunch of lines that indicate images are being pulled

```
[+] Running 5/5
 - Network docker_default        Created
                    0.8s
 - Volume "docker_wp_data"       Created
                    0.0s
 - Volume "docker_db_data"       Created
                    0.0s
 - Container docker-db-1         Created
                    0.6s
 - Container docker-wordpress-1  Created
                    0.6s
Attaching to docker-db-1, docker-wordpress-1
docker-db-1         | 2023-04-24 22:31:55+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server
~focal started.
docker-wordpress-1  | WordPress not found in /var/www/html - copying now...
docker-db-1         | 2023-04-24 22:31:56+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
docker-db-1         | 2023-04-24 22:31:56+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server
~focal started.
docker-db-1         | 2023-04-24 22:31:56+00:00 [Note] [Entrypoint]: Initializing database files
docker-wordpress-1  | Complete! WordPress has been successfully copied to /var/www/html
docker-wordpress-1  | No 'wp-config.php' found in /var/www/html, but 'WORDPRESS_...' variables supplied;
fig-docker.php' (WORDPRESS_DB_HOST WORDPRESS_DB_NAME WORDPRESS_DB_PASSWORD WORDPRESS_DB_USER)
docker-db-1         | 2023-04-24 22:31:56 0 [Warning] 'default-authentication-plugin' is MySQL 5.6 / 5.7
```

5. Inspect the application

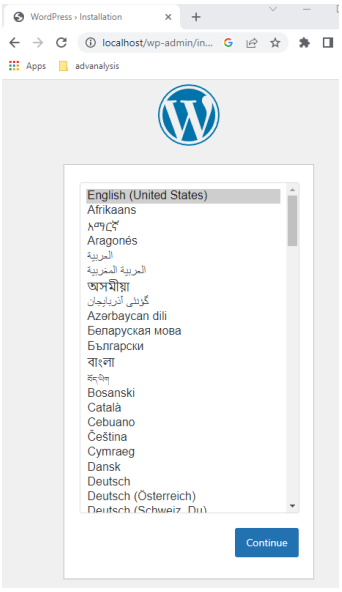
In another window check to see the processes that are running, the networks and the volumes

```
D:\docker>docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED         STATUS         PORTS                    NAMES
94a5c4df1422   mariadb:10.6.4-focal   "docker-entrypoint.s…"   5 minutes ago   Up 5 minutes   3306/tcp, 33060/tcp      docker-db-1
6707f75fe8e7   wordpress:latest       "docker-entrypoint.s…"   5 minutes ago   Up 5 minutes   0.0.0.0:80->80/tcp       docker-wordpress-1
24dc494565e0   nginx                  "/docker-entrypoint.…"   3 hours ago     Up 3 hours     0.0.0.0:8081->80/tcp     webby2
f7dff17700a7   nginx                  "/docker-entrypoint.…"   3 hours ago     Up 3 hours     0.0.0.0:8080->80/tcp     webby1

D:\docker>docker volume ls
DRIVER    VOLUME NAME
local     docker_db_data
local     docker_wp_data

D:\docker>docker network ls
NETWORK ID     NAME              DRIVER    SCOPE
e056bbfa7d6e   bridge            bridge    local
d7c6c410a4a4   docker_default     bridge    local
74dbc19ddc61   host              host      local
b0a01d01f8a8   none              null      local
```

If you want to be a bit more adventurous, use the exec command to inspect the inner workings of the containers in the application. Check to see that the WordPress application is running on your localhost port 80



6. Shut down the application

Shut down the application using “docker compose down” from the same directory you started the app in. This will require using a different command prompt

Examine the environment and notice that the volumes remain so that you can restart the app without losing any persistent data, however, the temporary network has been removed.

```
D:\docker>docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
e056bbfa7d6e   bridge    bridge    local
74dbc19ddc61   host      host      local
b0a01d01f8a8   none      null      local


D:\docker>docker volume ls
DRIVER    VOLUME NAME
local     docker_db_data
local     docker_wp_data
```

End Lab