

LU06b - Docker-Compose-File

Ziele

1. Ich verstehe die Struktur eines Docker-Compose-Files.
2. Ich kann ein Docker-Compose-Files nach Vorgaben anpassen.
3. Ich kann mögliche Fehlermeldung lesen und interpretieren.
4. Ich kann mindestens eine Massnahme bei Fehlermeldung vorschlagen und umsetzen.

Docker-Compose

Create a collection of interconnected containers, networks, and volumes with a single command.

Command-Line-Interface (CLI)

The most important commands regarding docker-compose:

```
docker-compose up: create all containers, networks and volumes described in
our docker-compose file
docker-compose up -d: same as above, but run containers in detached mode
docker-compose -f <filename> up: create containers based on a different
docker-compose file
docker-compose down: remove all containers and networks
docker-compose down -v: remove all containers, networks, and volumes
```

Docker-Compose file

The docker-compose tool uses files written in a data-serialization language called YAML (extension is .yml)

At the top level of each file, include the version of docker-compose, then you can list services (containers), plus any volumes or networks if necessary.

```
version: '3.8'

services:
  # any services go here
networks:
  # networks go here
volumes:
  # volumes
```

services

The next level down will be the name of the service/container.

Useful keywords

Keyword Purpose

- **build:** specify where to find the Dockerfile to build image
- **image:** can name a newly built image, or specify name of image to build from
- **volumes:** specify volumes or bind mounts to connect to image

networks will connect container to specified networks environment add environment variables
ports publish ports

```
services:
  service_name:
    build:
      context: ./folder_with_dockerfile
      dockerfile: Dockerfile-alternate.Dockerfile
    image: whatevername # like the -t flag when you are building your image
    # if you don't have a build command, Docker will try to build from an
    existing image with that name
    volumes:
      # with a named volume - also list name under top-level volumes
      - volume_name:/path/to/volume/on/container
      # for bind mounts
      - ./path/locally:/path/to/bind/mount/on/container
    # by default, docker-compose will create a single network for all
    containers
    networks:
      - network_name
    environment:
      DATABASE_URL: postgresql://username:password/localhost
      ANOTHER_VARIABLE: more-stuff
    ports:
      # <external port>:<internal port>
      - 8000:80
    second_service_name:
      #... all the keywords for this service
```

Volumes

If you include any named volumes, list their names under the top-level “volumes” key.
Anonymous volumes and bind mounts don’t need to be listed.

```
volumes:
  some_named_volume:
```

```
another_named_volume:
```

Networks

If you don't specify a network, all the containers in the docker-compose file will be put onto one network together.
 If you want multiple networks, specify all the names under the top-level volumes key.

```
networks:  
  network_name:  
  second_network_name:
```

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/modul/m347/learningunits/lu06/lu06b?rev=1741356756>

Last update: **2025/03/07 15:12**

