Compose

Mit Docker Compose gibt es eine Möglichkeit, die Konfiguration von Docker Containern komfortabel in einer Datei zu beschreiben und diese dann an den Docker Daemon zu übergeben, sodass diese den Container entsprechend der Konfiguration erstellt und ausführt. Hierbei können mehrere Container in einer Datei beschrieben werden. Zusätzlich wird automatisch ein gemeinsames internes Docker Netzwerk für die Container erstellt, sodass diese direkt miteinander kommunizieren können.

Die Konfiguration wird im sog. YAML Format eingegeben, dieses hat ein sehr striktes Format, welches einzuhalten ist. Eine solche Compose YAML Konfiguration kann z. B. wie folgt aussehen:

```
services:
  bookstack:
    image: lscr.io/linuxserver/bookstack
    container_name: bookstack
    environment:
      - DB_HOST=bookstack_db
      - DB_PORT=3306
      - DB_USER=bookstack
      - DB_PASS=Ein_langes_Passwort
      - DB_DATABASE=bookstackapp
    volumes:
      - ./bookstack_app_data:/config
    ports:
      - 80:80
    restart: unless-stopped
    depends_on:
      - bookstack_db
  bookstack_db:
    image: lscr.io/linuxserver/mariadb
    container_name: bookstack_db
    environment:
      - MYSQL_ROOT_PASSWORD=Noch_ein_extra_langes_Passwort
      - MYSQL_DATABASE=bookstackapp
      - MYSQL_USER=bookstack
      - MYSQL_PASSWORD=Ein_langes_Passwort
    volumes:
```

```
      - ./bookstack_db_data:/config
    restart: unless-stopped
```

Die Konfiguration wird in eine Datei mit dem Namen `docker-compose.yml` eingegeben. Es kann auch ein beliebiger Name vergeben werden, jedoch ist es übersichtlicher und einfacher, für jedes Projekt einen eigenen Ordner mit jeweils dieser Datei anzulegen. Dieser Name ist für Compose der Standard und wird standardmäßig bei anderen Befehlen verwendet (dann muss nicht extra der Name der Konfiguration nochmals angegeben werden).

Hierbei werden die selben Parameter verwendet, wie sie z. B. beim `docker run` Befehl übergeben werden. Sie müssen nur in das YAML Format übertragen werden. Einige Parameter können nicht 1:1 übernommen werden, da Compose einige zusätzliche Möglichkeiten bietet.

in diesem Beispiel werden 2 Container (Services) konfiguriert: *bookstack* und *bookstack_db*.

Alle Optionen sollten bereits vom docker run bekannt sein, auch das Image wurde dort verwendet, nur das es nicht expliziert als Image bezeichnet wurde (`sudo docker run --name test -d lscr.io/linuxserver/bookstack`).

Ein Parameter ist neu: `depends_on`. Mit diesem Parameter können Abhängigkeiten zwischen den Containern definiert werden. So muss in diesem Beispiel zuerst der Container *bookstack_db* erstellt werden und erst danach wird von Compose der Container *bookstack* erstellt.

Weiterhin konfiguriert Compose automatisch ein internes Default Netzwerk vom Typ Bridge für die beiden Container. Sodass diese sich untereinander mit den Namen aus der Config (*bookstack* und *bookstack_db*) ansprechen können. Diese Kommunikation erfolgt dann innerhalb des Docker Daemons und verlässt nicht den Host. D. h. in diesem Fall wäre die Datenbank (*bookstack_db*) nur für den anderen Container erreichbar. Weder von extern noch andere Container auf dem Docker Server können mit dem Container kommunizieren. Nur der Container *bookstack* wäre über Port 80 von extern und für andere Container erreichbar.

Um die Docker Compose Konfiguration auszuführen, kann am besten in das Verzeichnis der YAML Datei gewechselt werden. Danach wird je nach nach gewählter Installation `sudo docker-compose up -d` oder `sudo docker compose up -d` (keine Bindestrich zwischen docker und compose) eingegeben, um die Standard Konfiguration `docker-compose.yml` zu starten. Compose erstellt dann die gewünschten Container mit den angegeben Optionen. Sollten die Container bereits mit dieser Compose Konfiguration erstellt worden sein, so werden die Container in dieser neu erstellt, dessen Konfiguration geändert wurde.

Version #3
Erstellt: 2024-02-23 21:45:25 CET von Marcel
Zuletzt aktualisiert: 2024-03-29 19:37:17 CET von Marcel