Docker Compose

Commands für Docker Compose

- `docker-compose build` baut die Services
- `docker-compose up -d [SERVICENAME]` erstellt die Container und startet diese. Die Container werden in der korrekten Reihenfolge erstellt und gestartet. Wenn die Container bereits erstellt sind und laufen, bewirkt dieser Befehl einen Restart. -d : startet die Container im Hintergrund
- `docker-compose down` stoppt und entfernt alle Container, Netzwerke, Images und Volumes
- `docker-compose start [SERVICENAME]` startet die Container, dazu müssen diese aber vorher eingestellt worden sein.
- docker-compose stop [SERVICENAME] stoppt alle zu dem Stack gehörenden Container.
- `docker-compose rm -f '' entfernt alle zu dem Stack gehörenden Container, diese müssen gestoppt sein. -f : erzwingt das Löschen der Container * ''docker-compose logs` zeigt die logs aller zu dem Stack gehörenden Container an. Wenn der Stack mit docker-compose up -d im Hintergrund gestartet wurde können so die Logsfiles verfolgt werden.
- `docker-compose ps` zeigt die Liste aller dem Stack zugehörenden Container und deren Status an.
- `docker-compose exec SERVICENAME /bin/sh` gleich wie bei docker exec wird ein Befehl innerhalb des Containers ausgeführt. Da die Container, welche von docker-compose erstellt werden, immer den Namen des Ordners in dem sich das docker-compose.yml file befindet, enthält, ist der Name des Containers meist ORDERNAME_SERVICENAME_1 (für die erste Instanz des Services) und könnte auch mit docker exec ORDERNAME_SERVICENAME_1 angesprochen werden. Bei docker-compose exec kann der Container aber eben direkt über den SERVICENAMEN angesprochen werden. Hier werden die Parameter -it nicht benötigt.
- `docker-compose scale SERVICENAME=X` die Anzahl der laufenden Containerinstanzen des Service SERVICENAME wird auf X erhöht oder verringert. Dabei muss darauf geachtet werden, dass die Definition des Services im **`docker-compose.yml`** File skalierbar ist. Wenn zum Beispiel **`ports: -80:80`** angegeben ist kann es zu einem Konflikt kommen. Es kann nur ein Container von diesem Service erstellt werden, da ein zweiter ebenfalls auf den Port 80 des Hostsystem gemappt haben möchte, dieser aber bereits in Verwendung ist. In der **`docker-compose.yml`** muss **`ports: -80:80`** zu **`ports: -80`** geändert werden. Docker vergibt und managed die internen Ports selbst.

Einführung in Docker Compose

eine gute Einführung in die Syntax und Bedeutung der Befehle findest du hier. oder hier

Installation von Docker Compose

Wie man docker-compose installiert und ein erstes yml file erstellt findest du hier

Docker Compose erklärt

From:
http://wiki.waldhofer.at/ - **Wiki von Franz**

Permanent link:
http://wiki.waldhofer.at/doku.php?id=docker:compose&rev=1729241112

Last update: **2024/10/18 10:45**