

Docker Cheatsheet

<container>: Kann die id oder der Name sein

! steht für nötig

Anzeigebefehle

```
1  docker ps # Zeigt alle laufenden Container an
2      -a # Zeigt alle Container an, auch gestoppte
3
4  docker images # Zeigt alle lokalen Images an
5  docker logs <container> # Gibt den output und logs des Containers an, Toll
   zum debuggen
6  docker network ls # Zeigt alle Networks an
7  docker network inspect <network> # Gibt die Informationen zum genannten
   Network aus
8
9  docker image/container/network/compose # Nur docker oder eine der optionen
   schreiben und man bekommt vorschläge für Befehle
```

Erstellungsbefehle

```
1  docker build # Erstelle Image vom Dockerfile
2      <context> # ! Der Kontext von dem aus gebaut wird. Sollte auf den
   Ordner mit dem Dockerfile zeigen
3      -t <name:tag> # Gibt dem Image einen Namen, optional mit tag
4      -f <dockerfile> # Gibt das Dockerfile an, falls es nicht Dockerfile
   heißt oder es mehrere gibt
5
6  docker create # Gleiche Optionen wie run, nur das der Container nicht
   gestartet wird, nur erstellt
7  docker run # Erstellt und startet einen Container
8      --name <container name> # Name des Containers zur identifizierung
9      --rm # Löscht den Container sobald er gestoppt wird
10     -d # Steht für detached, startet den Container im Hintergrund
11     -p <local-port>:<docker-port> # Weist einem lokalen Port einen
   container Port zu.
12     -v <local-path>:<docker-path> # verbindet einen lokalen Pfad mit
   einem container Pfad, spiegelt alle lokalen Änderungen in den container,
   benutze ${PWD} als Angabe für den derzeitigen Pfad
13     <image-name> # ! Name des Images
```

Andere Befehle

```
1  docker start <container> # Startet den genannten Container
2  docker stop <container> # Stoppt den genannten Container
3  docker rm # Lösche den Container, muss gestoppt sein
4      -f # Falls der Container läuft, beende und lösche ihn
5      <container> # !
6
7  docker exec # Führt einen Befehl im Container aus
8      -it # Etwas esoterisch, am besten immer anfügen (ermöglicht normalen
9  input und output vom Befehl)
10     <container> # !
11     <befehl> # ! Der Befehl, welcher im Container ausgeführt werden
12     soll, mit Befehl 'bash' kann man in den Container hinein
13
14
15  docker image rm <image> # Lösche das Image
16  docker image prune # Löscht 'dangling' images, nützlich wenn man ein image
17  immer wieder baut und dann viele <none> images übrig bleiben
18
19  docker network create <network-name> # Erstelle ein neues network
20  docker network connect <network-name> <container> # Fügt einen laufenden
21  Container dem network hinzu
22
23  docker compose up # Baut, erstellt und startet alle Services (Container) im
24  compose file
25  docker compose down # Stoppt und löscht alle Services (Container) im compose
26  file
```