

# Docker auf der Kommandozeile

## Container starten und stoppen

- Starten

```
$ docker run --name foo hello-world
```

- Laufende Container anzeigen

```
$ docker ps
```

```
$ docker ps -a
```

- Stoppen

```
$ docker stop foo
```

```
$ docker kill foo
```

```
$ docker rm foo
```

- Ordner auf Host verfügbar machen

```
$ docker run -v $HOSTDIR:$DOCKERDIR
```

- Ports weiterleiten

```
$ docker run -P  
$ docker run -p$HOST_PORT:$GUEST_PORT
```

- Alle stoppen/entfernen

```
$ docker stop $(docker ps -a -q)  
$ docker rm $(docker ps -a -q)
```

## Laufende Container steuern

- Befehl an laufenden Container senden

```
$ docker exec <name> df -h
```

- Shell verbinden

```
$ docker attach <name>  
$ docker exec -it <name> sh
```

- Verbindung trennen ohne Container zu stoppen

```
CTRL-P, dann CTRL-Q
```

# Images

- Docker Registry (z.B. [dockerhub.com](https://dockerhub.com))

```
$ docker push  
$ docker search  
$ docker pull
```

- Lokale Images

```
$ docker images
```

- Erzeugen aus Dockerfile

```
$ docker build -t <name>:<version> <path>
```

- Lokales Image löschen

```
$ docker rmi <name>
```

## Export/Import

**load/save** erhält die History, **import/export** nicht.

- Load/Save

```
$ docker save <name>:<tag> > file.tar.gz  
$ docker load < file.tar.gz
```

- Import/Export

```
$ docker export <name>:<tag> > file.tar.gz  
$ cat file.tar.gz | docker import - <name>:<tag>
```

## Docker Networking

- Netzwerke verwalten

```
$ docker network create -d bridge test  
$ docker network rm test  
$ docker network connect test nginx  
$ docker network disconnect test nginx
```

- Infos zu vorhandenen Netzwerken

```
$ docker network ls  
$ docker network inspect <name>
```

- Container des selben Netzwerks sind über den Containernamen erreichbar

```
$ docker network create net1  
$ docker run -d --network=net1 --name ng1 nginx  
$ docker run -d --network=net1 --name ng2 nginx  
$ docker exec -it ng1 ping -c 3 ng1
```