

Docker CheatSheet

Tobias Köhler, Niklas Nikisch

Hochschule Mannheim, Fakultät für Informatik

Build

Baut ein Image aus einem Dockerfile im momentanen Verzeichnis und *tagged* es.

docker build -t appname

Listet alle Docker Images auf.

docker images

Entfernt das angegeben Image.

docker images rm Imageld

Ship

Einloggen um auf die Registry der Dockerseite zuzugreifen.

docker login

Lädt das getaggte Image in die Registry hoch.

docker push username/repository:tag

Tagged das Image für späteres hochladen.

docker tag <image> username/repository:tag

Zieht ein Image oder ein Repository aus der Registry.

docker pull appname

Suche in der Registry nach einem Image.

docker search name

Run

Startet die App und nimmt mit dem Befehl -p ein Portmapping vor, wodurch der Port 80 aus dem Container auf den Port 4000 des PCs gesetzt wird. Mit dem Befehl -d kann die App auch im Hintergrund gestartet werden.

docker run -p 4000:80 appname

Startet die App aus dem Repository.

docker run -p 4000:80 username/repository:tag

Others

Listet alle laufenden Container auf.

docker container ls

Stoppt den angegebenen Container.

docker container stop ContainerID

Entfernt den angegebenen Container.

docker container rm ContainerID

Listet alle laufenden Services auf.

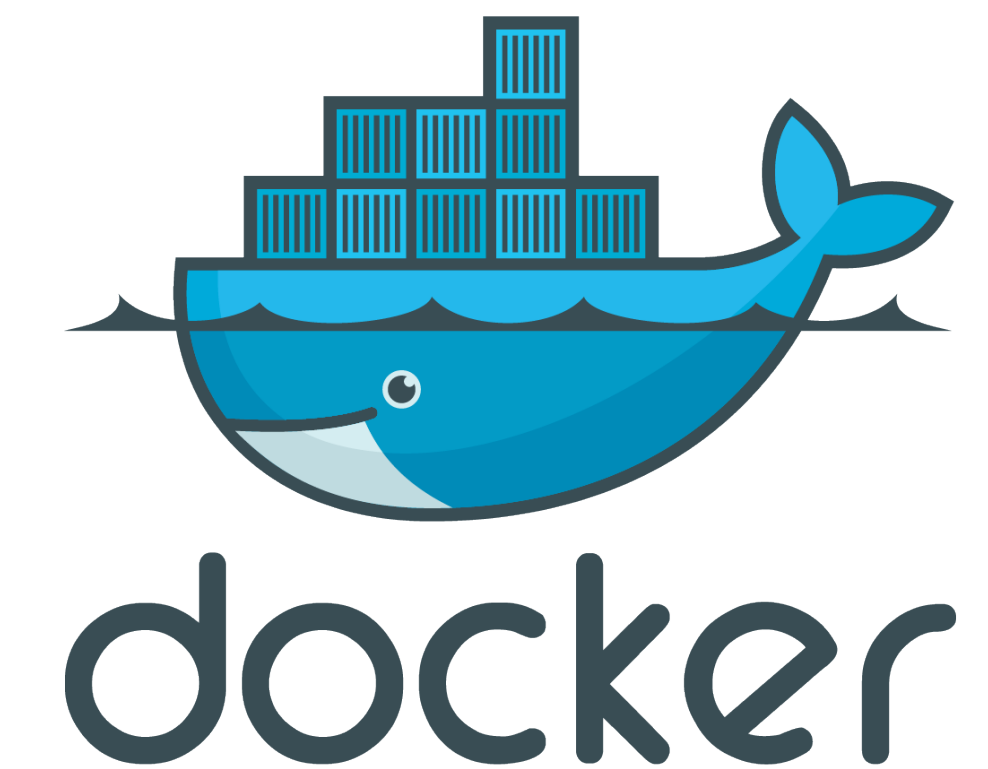
docker service ls ContainerID

Initialisiert einen Swarm damit er genutzt werden kann.

docker swarm init ContainerID

Startet die App als Swarm mit den Konfigurationen des composefiles.

docker stack deploy -p composefile appname



Dockerfile

Nutzen eines Vaterimages
FROM (Vaterimage)

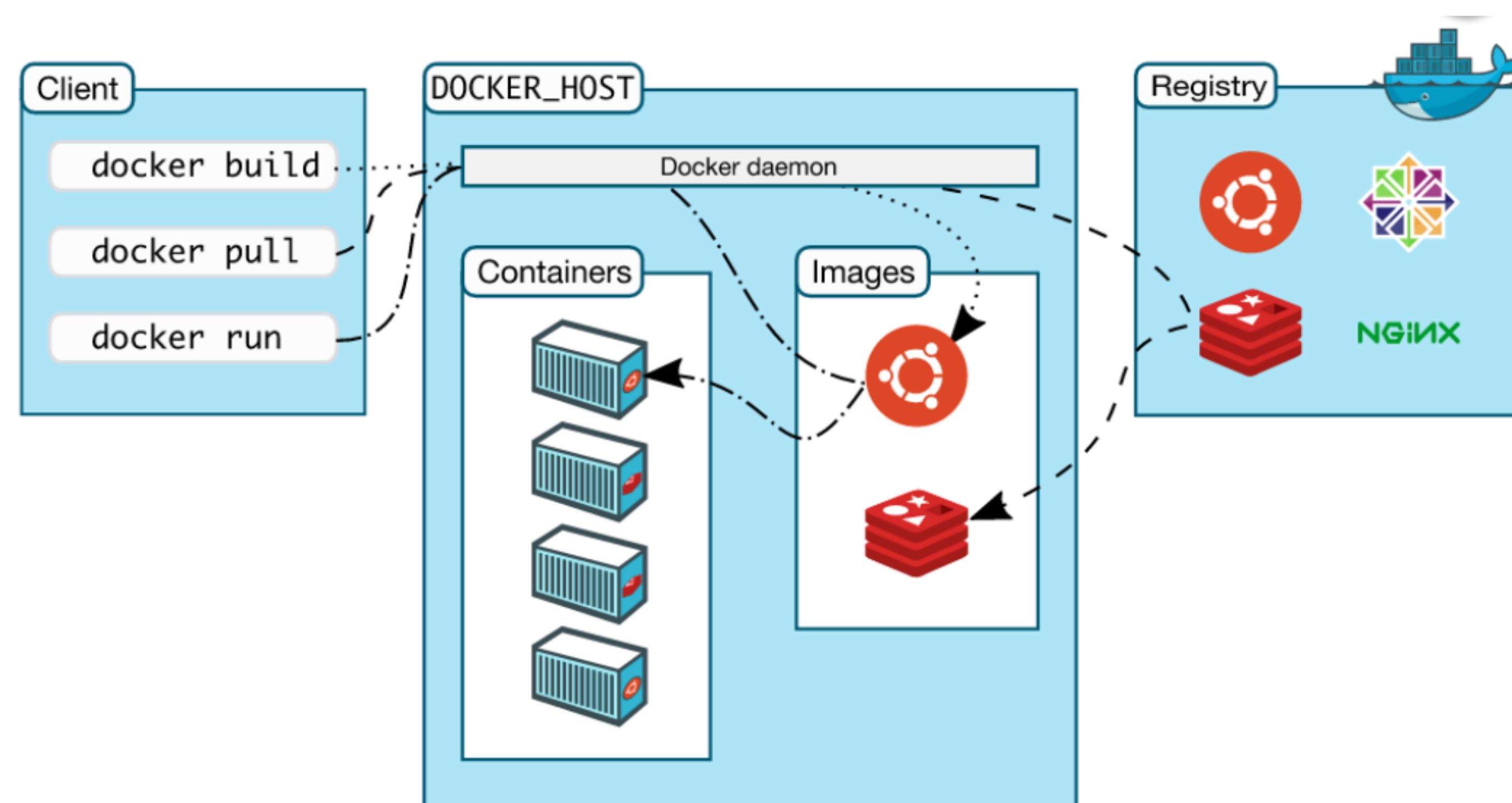
Setzt den Ordner in dem alle Befehle ausgeführt werden
WORKDIR (Ordner)

Kopiert Daten von (src) nach (des)
ADD (src) (des)

Gibt einen Port frei
EXPOSE (Nummer)

Startet ein Programm mit übergebenen Parametern
CMD [(Programm), (Parameter)]

Docker Ecosystem



Tobias Köhler



Niklas Nikisch