**Docker**

Ist ein standardisiertes Paket, dass die Softwareanwendungen mit allen Abhängigkeiten portabel und isoliert verpackt

So können Anwendungen schnell, einfach und reproduzierbar auf verschiedenen Systemen ausgeführt werden.

Vorteile Docker

- Platzersparnis durch Verzicht auf ganzes Betriebssystem

- Isolation eines Containers zu anderen Prozessen

- Standardisierter Deploymentprozess unterstützt durch effiziente Tools

- Gewisse Unabhängigkeit von Betriebssystem und Cloudanbieter

- Identisches Verhalten auf allen Betriebssystemen

- geringer Ressourcenverbrauch

- schnellere Startzeit als VM

VM vs. Docker

Hardware-Virtualisierung vs. Isolation

Eigener Kernel vs. Mitbenutzung Hostkernel

Funktionsweise

Docker benutzt Funktionen des Linuxkernels zur Isolation:

Dockerbefehle

|  |  |
| --- | --- |
| docker run [-p hostport:containerport] <image> | Startet neuen Container auf Basis der angegebenen Image Datei  Akzeptiert weitere Parameter (Ports) |
| docker ps [-a] | Zeigt alle laufenden Container  (-a auch die gestoppten) |
| docker start <container> | Startet einen bereits existierten, aber gestoppten Container |
| docker stop <container> | Stoppt einen laufenden Container |
| docker rm <container> | Entfernt einen Container |
| docker build <dockerfile> | Baut Dockerimage aus Dockerfile |
| docker push <image> | Veröffentlicht Dockerimage in Registry |

Dockerimages = Bauplan eines Containers

Docker Container = gestartete Instanz eines Dockerimages

Registry

- Es gibt eine Standard Registry

- man kann auch eigene betreiben

- Es gibt auch kommerzielle Angebote

Images

Ein Image wird von einem Dockerfile definiert:

FROM <basisimage> beschreibt das Basisimage auf das gebaut wird

COPY <file/ordner> kopiert eine Datei oder einen Ordner ins Image

RUN <befehl> führt während des Bauens einen Befehl aus

CMD <einstiegspunkt> beschreibt den Standard-Einstiegspunkt