

How to docker?

Docker something

Julia Winkler

19.06.2024

Gliederung

Why use Docker?

Trusted by developers. Chosen by Fortune 100 companies.

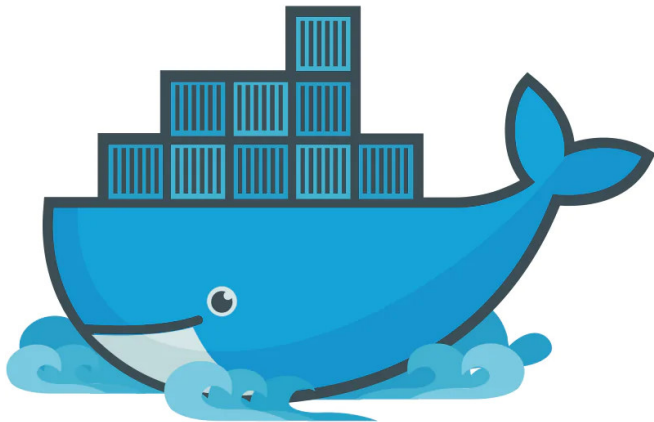
Docker provides a suite of development tools, services, trusted content, and automations, used individually or together, to accelerate the delivery of secure applications.

"a sandboxed process on your machine that is isolated from all other processes on the host machine"

"It works on my computer"

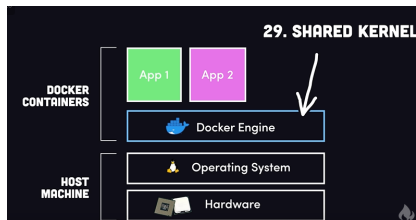
"faster onboarding and testing while also simplifying the deployment of services"

Wer ist Moby Dock?



Was ist Docker?

- ▶ ? (vllt nicht als Folie sondern nur was erwähnen)
- ▶ Toolbox
- ▶ sandbox
- ▶ konfigurierbare Umgebung



Was ist Docker?

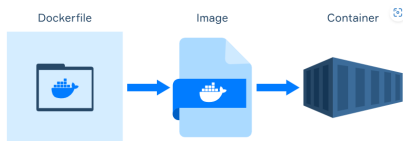


Figure: Zusammenhang der Docker Komponenten

Wichtige Begriffe

Docker - freie Software zur Isolierung von Anwendungen mit Hilfe von Containervirtualisierung
Container - Umgebung in der die tatsächliche Anwendung läuft
Image - Blaupause, um einen Container zu erstellen
Dockerfile - Anleitung, um ein Image zu erstellen
Docker Compose - Orchestrierungstool (?), Wrapper für einen oder mehrere Container
Container Registry (Docker Hub) - Ort an dem viele verschiedene Images gespeichert und geteilt werden können
Docker Desktop - UI für alles was man in der CLI machen kann (?)

Wie kriege ich dieses "Docker"?

Hello World

```
$ docker -v
```

```
$ docker --help
```

```
$ docker run hello-world
```

Ab ins Terminal

Was ist Docker?

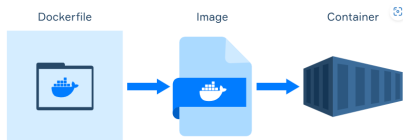


Figure: Zusammenhang der Docker Komponenten

Dockerfile

- ▶ Ein Dockerfile ist die Anleitung um ein Image zu erstellen.
- ▶ Standardmäßig heißt die Datei 'Dockerfile'

```
make lc-line-concept|
```

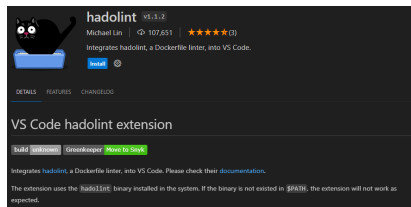


Figure: Plugin für die Arbeit mit Docker

Dockerfile

required

FROM base_image:tag

- ▶ FROM, CMD & Entrypoint, RUN, COPY, ARG
- ▶ Layer

Python bsp

```
FROM alpine
```

```
# Exec form
```

```
CMD ["echo", "Hello World."]
```

```
#shell form
```

```
CMD echo Hello Students
```

Volumes



React

```
FROM alpine
```

```
# Exec form
```

```
CMD ["echo", "Hello World."]
```

```
#shell form
```

```
CMD echo Hello Students
```

Multistage builds

► 1

React - Multistage

```
FROM alpine
```

```
# Exec form
```

```
CMD ["echo", "Hello World."]
```

```
#shell form
```

```
CMD echo Hello Students
```

Dockerfile Best practices

- ▶ RUN commands
- ▶ Order of COPY
- ▶ Volumes
- ▶ Multistage

Docker Compose

- ▶ Vorteile
- ▶ UseCases

Docker Compose zu Python

```
FROM alpine
```

```
# Exec form
```

```
CMD ["echo", "Hello World."]
```

```
#shell form
```

```
CMD echo Hello Students
```

Docker Compose Webapp

```
FROM alpine
```

```
# Exec form
```

```
CMD ["echo", "Hello World."]
```

```
#shell form
```

```
CMD echo Hello Students
```

important Commands

- ▶ `docker run`
- ▶ `docker build`
- ▶ `docker push`, `pull`
- ▶ `docker ps -a`
- ▶ `docker rm / rmi`
- ▶ ...

Cooler Quellen und so weiter

- ▶ <https://www.docker.com/>
- ▶ <https://docs.docker.com/get-started/>
- ▶