

How to docker?

# Docker something

Julia Winkler

19.06.2024

# Gliederung

Why use Docker?

## **Trusted by developers. Chosen by Fortune 100 companies.**

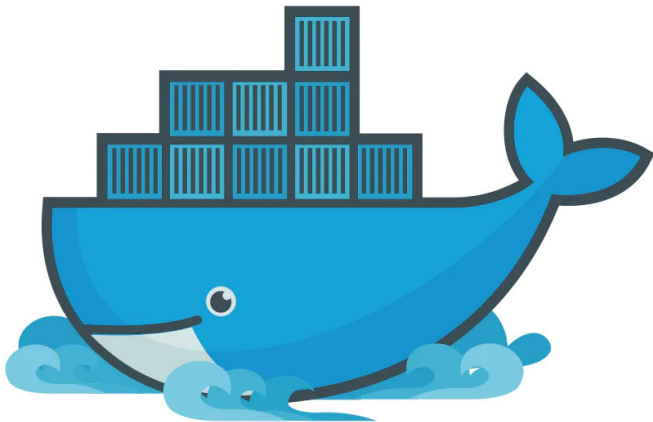
Docker provides a suite of development tools, services, trusted content, and automations, used individually or together, to accelerate the delivery of secure applications.

"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?



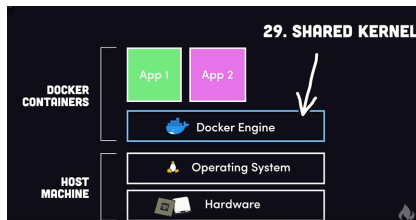
# Was ist Docker?

? (vllt nicht als Folie sondern nur was erwähnen)

Toolbox

sandbox

configurierbare Umgebung



# Was ist Docker?

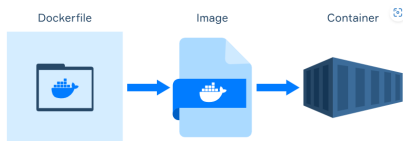


Figure: Zusammenhang der Docker Komponenten

# Wichtige Begriffe

## Docker

freie Software zur Isolierung von Anwendungen mit Hilfe von Docker Containervirtualisierung

## Container

Umgebung in der die tatsächliche Anwendung läuft

## Image

Blaupausen, um einen Container zu erstellen

## Dockerfile

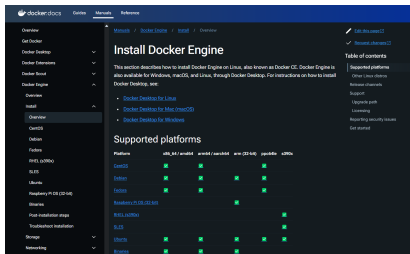
Anleitung, um ein Image zu erstellen

## Registry

z.B. Docker Hub, EAC.... Ort an dem viele verschiedene Images gespeichert und geteilt werden können



# Wie kriege ich dieses "Docker"?



The screenshot shows the Docker documentation website. The main heading is "Install Docker Engine". Below it, a paragraph explains that this section describes how to install Docker Engine on Linux, also known as Docker CE. It mentions that Docker Engine is also available for Windows, macOS, and Linux through Docker Desktop. There are three links: "Docker Desktop for Linux", "Docker Desktop for Mac/Linux/OS", and "Docker Desktop for Windows".

Below the links is a section titled "Supported platforms" with a table showing compatibility for various operating systems. The table has columns for Platforms, x86\_64, amd64, s390x, ppc64le, and ARMv8.

Platforms	x86_64	amd64	s390x	ppc64le	ARMv8
CentOS	✓	✓			
Ubuntu	✓	✓	✓	✓	
Fedora	✓	✓			
RHEL 8 (x86_64)	✓	✓			
RHEL 8 (s390x)			✓		
RHEL 8 (ppc64le)				✓	
SLES	✓	✓			
Ubuntu	✓	✓	✓	✓	
RHEL 8	✓	✓			

Doku

# Hello World

---

```
$ docker -v
```

```
$ docker --help
```

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

---

Ab ins Terminal

# docker run

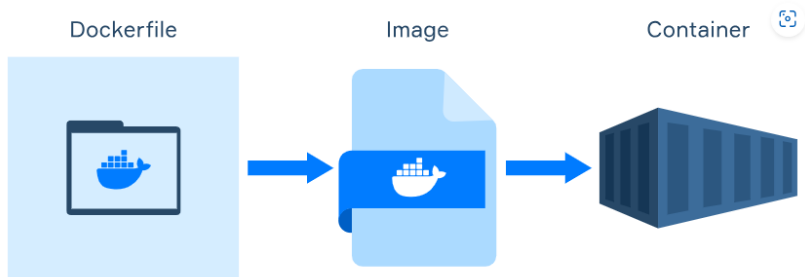
---

```
$ docker  
  adf  
  sadf
```

---

```
docker build [OPTIONS] PATH | URL | - Build an  
  -f, --file string Name of the Dockerfile (default is  
  -t, --tag stringArray Name and optionally  
  PATH in most cases .
```

# Was ist Docker?



# Dockerfile

Ein Dockerfile ist die Anleitung um ein Image zu erstellen.

Image zu einem Dockerfile erstellen: `docker build .`

Standardmäßig heißt die Datei 'Dockerfile'

wieter Optionen mit `docker buildx build`

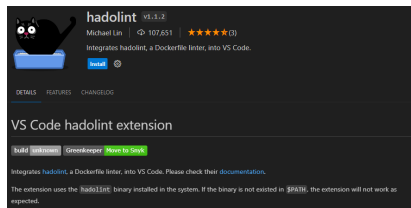


Figure: Plugin für die Arbeit mit Docker

# docker build command

```
docker build [OPTIONS] PATH | URL | -
```

Build an image from a Dockerfile [OPTIONS]

**-f, --file string** Name of the Dockerfile (default:  
"PATH/Dockerfile")

**-t, --tag stringArray** Name and optionally a tag (format:  
"name:tag")

**PATH** in most cases .

# Dockerfile

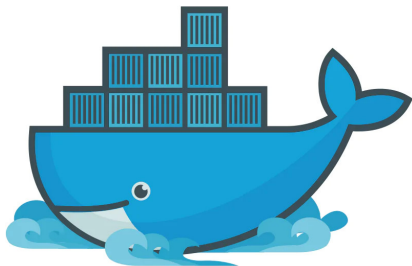


Figure: Code Tour



# CMD vs. ENTRYPOINT

---

```
docker build -t example:cmd -f Dockerfile.cmd .  
docker build -t example:entry -f Dockerfile.entry .  
  
docker run example:cmd  
docker run example:cmd hello  
docker run example:entry hello
```

---

CMD: wird überschrieben EP: bestimmte den command,  
argumente können hinzugefügt werden beide: bestimmen, was  
nach dem start ausgeführt wird



# RUN

---

```
docker build -t example:single -f Dockerfile.single .  
docker build -t example:multi -f Dockerfile.multi .  
  
docker ps -a  
docker images
```

---

spart speicher schnellerer build

# 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
```

---

title

OpenDrone Map

tes Arbeit

liwi arbeit

rame work

# Cheatsheet

`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/>