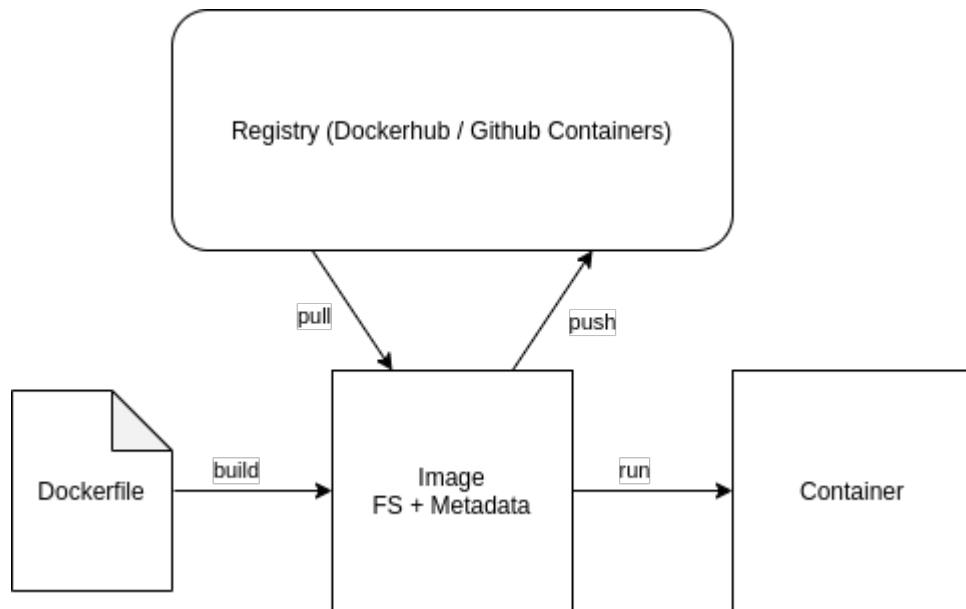


# Docker: Basics

- Ein Container ist ein Prozess
- Isolation mittels Linux-Kernelfunktionen
- (Auf Windows läuft Docker in VM)
- Dieser wird Ressourcenlimitiert (CPUs/memory)
- Der Prozess bekommt ein eigenes RootFS
- Speicher/Netzwerk werden von Docker manipuliert
- Im RootFS sind alle Abhängigkeiten dabei
- Security (Mehrere Prozesse auf einer Maschine)
- Portability (Abhängigkeiten sind mit dabei)
- Skalierbar (Docker Swarm / Kubernetes)



Weiterführende Links:

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

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

# Docker: Pull image and execute

- Besuch [hub.docker.com](https://hub.docker.com)
- Finde offiziellen Ubuntu container und lese Beschreibung
- Lade Image herunter
- Starte neuen ubuntu:latest Container im Hintergrund
- Starte Befehl `/bin/bash` in dem neuen Container
- Fahr den Container herunter

Download container	<code>docker pull</code>
Start container	<code>docker run -itd --name &lt;c_name&gt; &lt;i_name:tag&gt;</code>
Execute in container	<code>docker exec -it &lt;c_name&gt; &lt;command&gt;</code>
Stop container	<code>docker stop &lt;c_name&gt;</code>

Weiterführende Links:

<https://docs.docker.com/engine/reference/commandline/docker/>

<https://docs.docker.com/engine/reference/commandline/run/>

# Dockerfile: Create new image

- git clone <https://github.com/klosskopf/mandelbrot.git>
- Verstehe das Dockerfile
- Baue neues Image aus Dockerfile
- Starte neuen Dockercontainer aus Image
- Führe “.mandelbrot” im Container aus

Weiterführende Links:

<https://docs.docker.com/engine/reference/builder/>

Build Docker Image	Docker build -t <i_name> <verzeichnis>
Starte Container	docker run -it --name <c_name> <i_name:tag>
mandelbrot	./mandelbrot <x> <y> <zoom> <iteration>

# Github: upload image to ghcr.io

- Erstelle PAT in Github (mit packages:read/write Rechten)
- Login in Komandozeile bei ghcr.io
- Upload mandelbrot image zu ghcr.io

Erstelle PAT	In Github → Settings → Developer → Personal Access Token → generate new token Wähle “packages:write” → Generate token (speicher den Token; später nicht mehr erreichbar)
Login to ghcr.io	<code>docker login ghcr.io -u &lt;username&gt;</code> (Mit PAT als passwort)
Tag image	<code>docker tag &lt;i_name&gt; &lt;tag&gt;</code> ( <code>docker tag mandelbrot ghcr.io/&lt;username&gt;/mandelbrot:latest</code> )
Upload image	<code>docker push &lt;tag&gt;</code>

Weiterführende Links:

<https://docs.github.com/en/packages/working-with-a-github-packages-registry/working-with-the-container-registry>

# Github: Use docker in actions

- Erstell ein neues Github Repository
- Geb Image für neues Repository frei
- Erstell neue Action und ersetze Inhalt mit action.yml
- Commit, untersuche Action und erfreue dich am Artefakt

Erstelle PAT	In Github → Settings → Developer → Personal Access Token → generate new token Wähle “packages:write” → Generate token (speicher den Token; später nicht mehr erreichbar)
Login to ghcr.io	docker login ghcr.io -u <username> (Mit PAT als passwort)
Tag image	docker tag <i_name> <tag> (docker tag mandelbrot ghcr.io/<username>/mandelbrot:latest )
Upload image	docker push <tag>

Weiterführende Links:

<https://docs.github.com/en/packages/working-with-a-github-packages-registry/working-with-the-container-registry>

# Docker-compose: build network

- Untersuche docker-compose.yml
- Starte docker-compose
- Inspect network
- Get shell in container\_2
- Ping container\_1 und container\_3

Start docker-compose	docker-compose up -d
Show networks	docker networks ls
Inspect network	docker network inspect <networkname>
Execute in container	docker exec -it <c_name> <command>
Stop docker-compose	docker-compose down

Weiterführende Links:

<https://docs.docker.com/compose/>

<https://docs.docker.com/compose/compose-file/compose-file-v3/>

# Docker-compose: volumes

- Start the docker-compose
- Erstell Ordner und Datei in container\_2
- Restart docker-compose
- Suche die Datei
- Stop docker-compose
- Füge Volume in docker-compose.yml hinzu
- Wiederhole die obigen Schritte

Start docker-compose	<code>docker-compose up -d</code>
Execute in container	<code>docker exec -it &lt;c_name&gt; &lt;command&gt;</code>
Create Ordner Create Datei Change Directory Show content	<code>mkdir &lt;name&gt;</code> <code>touch &lt;path&gt;/&lt;name&gt;</code> <code>cd &lt;path&gt;</code> <code>ls</code>
Restart docker-compose	Docker-compose restart
Stop docker-compose	<code>docker-compose down</code>

Weiterführende Links:

<https://docs.docker.com/storage/volumes/>

<https://docs.docker.com/compose/compose-file/compose-file-v3/>