

### Docker auf Windows installieren

#### 1. Voraussetzungen:

- Windows 10 oder 11 (64-bit)
- WSL2 (Windows Subsystem for Linux 2) aktiviert (wird bei der Installation unterstützt)

#### 2. Installation:

- Lade Docker Desktop von https://www.docker.com/products/docker-desktop herunter.
- Installiere es und aktiviere dabei die WSL2-Integration.
- Nach der Installation: Docker Desktop starten.

#### 3. Testen, ob alles funktioniert:

docker --version



### 间 Images – Vorlagen für Container

Ein **Image** ist eine schreibgeschützte Vorlage, die alles enthält, was eine Anwendung braucht: Quellcode, Laufzeit, Bibliotheken und Abhängigkeiten. Container werden aus Images erstellt.

Befehl	Beschreibung
docker pull <image/>	Image aus Docker Hub laden
docker images	Lokale Images auflisten
docker rmi <image/>	lmage löschen
docker build -t name:tag .	Image aus Dockerfile bauen

Ein **Container** ist eine laufende Instanz eines Images. Er ist leichtgewichtig, portabel und isoliert. Du kannst ihn dir wie eine kleine virtuelle Maschine vorstellen – nur schneller und effizienter.

Befehl	Beschreibung
docker run <image/>	Container starten
docker run -it <image/> /bin/bash	Interaktiver Start mit Shell
docker run -d <image/>	Im Hintergrund starten
docker run -p 8080:80 <image/>	Port-Weiterleitung (host:container)
docker ps	Laufende Container anzeigen
docker ps -a	Alle Container (auch gestoppte) anzeigen
docker stop <container></container>	Container stoppen
docker start <container></container>	Gestoppten Container starten
docker rm <container></container>	Container löschen
docker logs <container></container>	Logs anzeigen
docker exec -it <container> <cmd></cmd></container>	Befehl im laufenden Container ausführen

# Volumes – Daten persistent speichern

**Volumes** ermöglichen es, Daten außerhalb eines Containers zu speichern, sodass diese beim Löschen des Containers erhalten bleiben. Ideal für Datenbanken oder Logs.

Befehl	Beschreibung
docker volume create <name></name>	Volume erstellen
docker volume ls	Volumes anzeigen
<pre>docker run -v <host_path>:<container_path></container_path></host_path></pre>	Volume mounten
<pre>docker volume inspect <name></name></pre>	Details anzeigen

# ■ Dockerfile – Eigene Images erstellen

Ein **Dockerfile** ist eine einfache Textdatei, die Schritt für Schritt beschreibt, wie ein Image gebaut wird. Du definierst dort z. B. Basis-Image, kopierte Dateien und Startbefehle.

```
# Basis-Image
FROM openjdk:17-jdk-alpine

# Arbeitsverzeichnis setzen
WORKDIR /app

# JAR-Datei kopieren
COPY target/app.jar app.jar

# Befehl beim Start
ENTRYPOINT ["java", "-jar", "app.jar"]
```

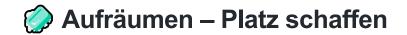
#### Image bauen:

```
docker build -t mein-image .
```

## ★ Docker Compose – Mehrere Container koordinieren

**Docker Compose** erlaubt es, mehrere Container (z. B. App + Datenbank + Webserver) mit einer einzigen Datei (docker-compose.yml) zu konfigurieren und zu starten. Ideal für Entwicklungsumgebungen.

Befehl	Beschreibung
docker-compose up	Container starten (laut YML-Datei)
docker-compose down	Container stoppen und aufräumen



Docker speichert standardmäßig vieles im Cache. Die folgenden Befehle helfen, ungenutzte Daten zu entfernen:

docker system prune
docker container prune
docker image prune

# Alles Ungenutzte löschen
# Ungenutzte Container löschen

# Ungenutzte Images löschen