

Spickzettel: Docker – Installation & lokale Konfiguration

Ziel

- Docker lokal auf **Windows**, **Linux** und **macOS** installieren
 - Erste Konfigurationsschritte für CLI & Desktop
-

Windows (Windows 10/11)

Docker Desktop (empfohlen)

1. Download: <https://www.docker.com/products/docker-desktop>
2. Installer ausführen
3. Option "Use WSL 2" aktivieren (bei Windows 10 erforderlich)
4. Neustart und Docker Desktop öffnen
5. Test:

```
docker version
docker run hello-world
```

WSL2-Backend aktivieren

```
wsl --install
wsl --set-default-version 2
```

Optional: CLI in Git Bash nutzen

- docker ist auch über Git Bash verfügbar, wenn PATH korrekt gesetzt
-

Linux (Ubuntu / Debian / Fedora / Arch)

Installation (Beispiel Ubuntu)

```
sudo apt update
sudo apt install ca-certificates curl gnupg lsb-release
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --
dearmor -o /etc/apt/trusted.gpg.d/docker.gpg

echo "deb [arch=$(dpkg --print-architecture)]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io
```

Nach der Installation

```
sudo systemctl enable docker  
sudo systemctl start docker
```

Optional: Docker ohne sudo nutzen

```
sudo usermod -aG docker $USER  
# Danach neu anmelden oder neustarten
```

♂ Docker Rootless Mode (Linux)

Vorteile

- Docker-Engine läuft ohne Root-Rechte → sicherer für Shared-Host-Umgebungen

Einrichtung

1. Voraussetzung: Docker >= 20.10, Benutzer braucht eigenes uidmap

```
sudo apt install uidmap
```

2. Rootless-Docker installieren:

```
dockerd-rootless-setuptool.sh install
```

3. Umgebung setzen:

```
export PATH=/usr/bin:$PATH  
export DOCKER_HOST=unix:///run/user/$(id -u)/docker.sock
```

4. Testen:

```
docker run hello-world
```

Hinweise

- Einige Funktionen (z. B. host-Netzwerk, cgroup) sind eingeschränkt
 - Nicht für produktive Multinode-Szenarien empfohlen
 - Mehr Infos: <https://docs.docker.com/engine/security/rootless/>
-

macOS

Docker Desktop (empfohlen)

1. Download: <https://www.docker.com/products/docker-desktop>
2. .dmg installieren und Docker starten
3. CLI steht nach Installation automatisch zur Verfügung

4. Test:

```
docker version  
docker run hello-world
```

Erste Konfiguration

- **Image-Cache prüfen:** `docker system df`
 - **Standardnetzwerk testen:** `docker network ls`
 - **Docker-Infos anzeigen:** `docker info`
 - **Containerverzeichnis prüfen:** Linux: `/var/lib/docker`, macOS/Windows: virtuell
-

Portainer – Docker-Management im Browser

- Webbasierte UI zur Verwaltung von Containern, Volumes, Netzwerken, Stacks etc.

Portainer starten

```
docker volume create portainer_data  
  
docker run -d \  
-p 9443:9443 \  
-p 9000:9000 \  
--name portainer \  
--restart=always \  
-v /var/run/docker.sock:/var/run/docker.sock \  
-v portainer_data:/data \  
portainer/portainer-ce:latest
```

Zugriff

- Öffne `http://localhost:9000` (oder `https://localhost:9443`)
- Benutzerkonto anlegen & Docker-Umgebung verbinden

Vorteile

- Containerverwaltung ohne CLI
 - Übersicht über Volumes, Images, Netzwerke, Logs
 - Ideal für Lern- & Demo-Setups, kleine Teams, Einsteiger
-

Test

```
docker run hello-world
```

- Sollte eine Erfolgsmeldung ausgeben und bestätigen, dass Docker läuft

Tipps

- Auf Windows/macOS: Docker Desktop bietet GUI & CLI
- Auf Linux: CLI + eigene Tools (z. B. Portainer) möglich
- Regelmäßiges Aufräumen: `docker system prune`
- Debugging: `docker info`, `docker logs`, `docker inspect`