

# Docker

- [Docker auf Ubuntu installieren](#)
- [Docker Ressourcen begrenzen](#)

# Docker auf Ubuntu installieren

Der Artikel Beschreibt die Installation von Docker auf der Ubuntu-VM unter Berücksichtigung der `systemd` Proxykonfiguration.

```
for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do apt remove $pkg; done
```

## GPG-Key der Docker Pakete installieren

Erstellen Sie eine Schritt-für-Schritt-Anleitung:

1. `apt update`
2. `apt install ca-certificates curl gnupg`
3. `install -m 0755 -d /etc/apt/keyrings`
4. `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg`
5. `chmod a+r /etc/apt/keyrings/docker.gpg`

## Das Repository zu den APT Ressourcen hinzufügen

```
echo \  
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \  
  "$(. /etc/os-release && echo "$VERSION_CODENAME)" stable" | \  
  tee /etc/apt/sources.list.d/docker.list > /dev/null  
apt update
```

## Aktuelle Docker-Version installieren

```
apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

# Docker testen

```
docker run hello-world
```

Dieser Test schlägt u.a. dann fehl, wenn das System eine Proxy verlangt und die `systemd` Proxykonfiguration noch fehlt.

## `systemd` Proxykonfiguration einrichten

1. `mkdir -p /etc/systemd/system/docker.service.d`
2. Datei `/etc/systemd/system/docker.service.d/http-proxy.conf` mit folgendem Inhalt anlegen:

```
[Service]
Environment="HTTPS_PROXY=<Proxy-IP:Port>"
Environment="HTTP_PROXY=<Proxy-IP:Port>"
```

3. `reboot` !

# Docker Ressourcen begrenzen

Damit ein Container nicht alle System-Ressourcen für sich beanspruchen kann...

Was z.B. Nextcloud gelegentlich tut :-)

```
services:
  nextcloud:
    image: nextcloud:latest
    # ... deine anderen Einstellungen ...
    deploy:
      resources:
        limits:
          cpus: '2.0'      # Maximal 2 CPU-Kerne nutzen
          memory: 2G      # Maximal 2 Gigabyte RAM nutzen

  db:
    image: mariadb:10.11
    # ... deine anderen Einstellungen ...
    deploy:
      resources:
        limits:
          cpus: '1.5'
          memory: 1G
```