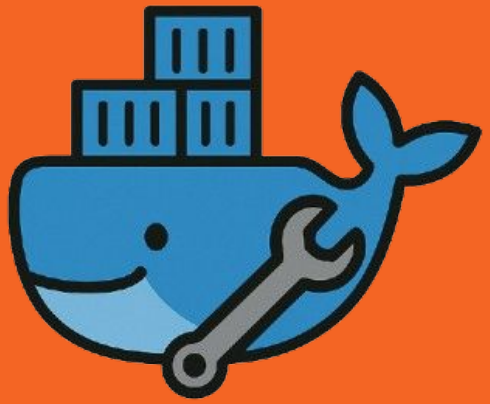


DOCKER FÜR MAKER

Container als Werkzeugkasten für Kreative

Florian Keller 10.10.2025
florian@keller.family

BareMetal/VM/Docker/Kubernetes vergleich 1/2

Bare-Metal-Server

Vorteile Maximale Performance; kein Virtualisierungs-Overhead; volle Kontrolle

Nachteile Schlechte Ressourcenauslastung; keine Skalierbarkeit; hohe Stromkosten; hohe Anschaffungskosten.

Aufwand / Komplexität Hoch Manuelle Verwaltung von OS, Treibern und Updates.

Virtuelle Maschine (VM)

Vorteile Sehr starke Isolation/Sicherheit; flexibles OS (Windows, Linux, macOS).

Nachteile Hoher Ressourcenverbrauch (für jedes OS); lange Boot-Zeiten; geringe Portabilität (große Images); Lizenzkosten für Gast-OS.

Aufwand / Komplexität Mittel Management von Hypervisor und vielen Gast-Betriebssystemen.

BareMetal/VM/Docker/Kubernetes vergleich 2/2

Docker (Container)

Vorteile Extrem schnell (Start in Sekunden); ressourcenschonend; hohe Portabilität ("funktioniert überall gleich"); einfache Versionskontrolle.

Nachteile Geringere Isolation als VM (teilen sich den Host-Kernel); nicht für Apps, die ein komplettes, einzigartiges OS brauchen.

Aufwand / Komplexität Niedrig Lernen des Dockerfile-Konzepts vieles einfach zu Automatisieren, manches selbstheilend.

Kubernetes (K8s) (wie Docker auf Steroiden)

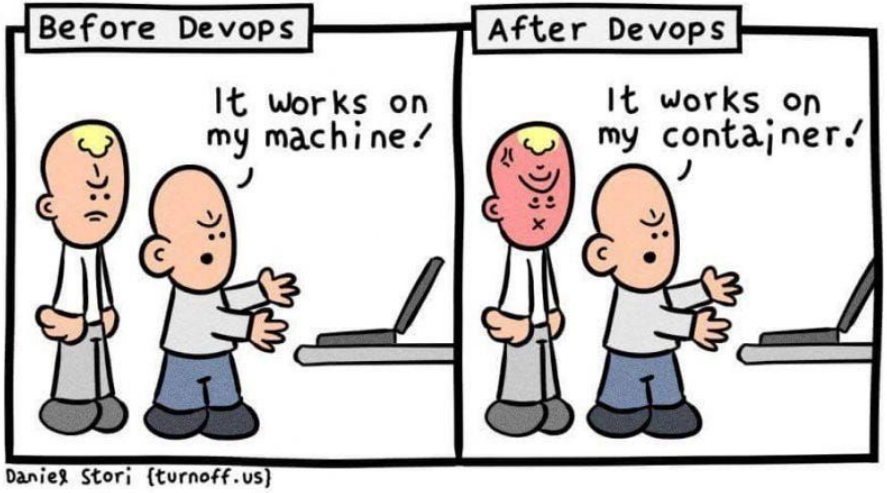
Vorteile Automatisches Healing (fehlerhafte Apps werden neu gestartet); automatische Skalierung (hoch/runter/seitwaerts); Lastverteilung; effiziente Verwaltung von Clustern.

Nachteile Sehr hohe Komplexität im Setup und Betrieb; hoher Lernaufwand; erfordert mindestens 3 Worker-Nodes für Hochverfügbarkeit.

Aufwand / Komplexität Sehr Hoch: Spezialwissen für Orchestrierung, Networking und Speicherung in Clustern erforderlich.

Was ist Docker

Was ist Docker



Docker ist eine Technologie zur **Containerisierung** von Software. Im Kern löst es das klassische Problem: "Es funktioniert auf meinem Rechner, aber nicht auf dem Server." Docker verpackt eine Anwendung **mitsamt aller Abhängigkeiten** (Bibliotheken, Konfigurationsdateien, Laufzeitumgebung) in eine standardisierte, portable Einheit – den sogenannten **Container**. Diese Container sind leichtgewichtig, da sie das Betriebssystem des Host-Rechners gemeinsam nutzen, anders als sperrige Virtuelle Maschinen (VMs).

Vorteile für Maker

Als Maker soll das Zeug Laufen und Funktionieren, man sollte sich nicht rumschlagen müssen.
Als Beispiel eine Datenbank fuer die Messwerte der selbstgebauten Bierbraustation und Abfüllanlage mit Anschliessender Visualisierung des Brauprozesses.

Entweder von Hand alles selbst Programmieren, gewaltiger aufwand.

VM und Datenbank mit visualisierung selbst von Hand einrichten und regelmaessig warten damit es keine Probleme gibt, Breaking changes bei den Updates Linux VM macht auch Probleme nach Major Upgrade…

Alles im Container (Docker)
docker run -d --name=grafana -p 3000:3000 grafana/grafana
docker run -d --name=influxdb -p 8086:8086 --volume /data/influxdb:/var/lib/influxdb2 influxdb:2.0.8

Passende Hardware

Raspberry PI (nicht zu empfehlen)

Sehr hohe kosten fuer sehr wenig leistung.

(Preise Schweiz) PI 16gb+SD+CASE+PSU = 198.4chf

Mini PC

MSI CUBI (n100) 109chf

Billig m.2 Disk 14chf

16gb ram 40chf

Alter PC/Laptop der noch rumfliegt

Je nachdem hoher Stromverbrauch aber reicht fuer den Anfang

Docker Installieren

Windows / MAC

Unter Windows gibt es 2 Varianten welche Virtualisierung auf dem System ist
Passende Anleitung unter
https://docs.docker.com/desktop/setup/install/windows-install/

Auch bei MAC ist es nicht mit einem befehl getan daher hier noch die Anleitung fuer MAC OS
https://docs.docker.com/desktop/setup/install/mac-install/

Installation Linux

```
curl -fsSL get.docker.com | sh
```

Portainer fuer Management

docker volume create portainer_data
docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:lts

docker volume create portainer_data	Erstellt ein Fixes Volume (wie eine Virtuelle Festplatte)
-d	Detached mode, damit der Container im Hintergrund laeuft
-p 8000:8000 -p 9443:9443	Port mappings Links der Port auf dem Host System, Rechts der Port innerhalb des Containers

--name portainer	Name des Containers
--restart=always	Halte den Container am laufen
-v /var/run/docker.sock:/var/run /docker.sock	Verbindet die “Datei” Lesend/Schreibend in den Container Das ist nur bei Speziellen Containern noetig
-v portainer_data:/data	Verbindet das Volume portainer_data in den Container unter dem Pfad “/Data/”

Portainer uebersicht

Environments

Search by name, group, tag, status, URL... | Refresh | Kubeconfig

Click on an environment to manage

Platform | Connection Type | Status | Tags | Groups | Agent Version | Clear all | Sort By

local Up 2025-10-06 10:37:50 Standalone 20.10.21 /var/run/docker.sock
Group: Unassigned No tags Local
4 stacks 16 containers 10 5 0 0 31 volumes 23 images 4 CPU 10.4 GB RAM
Live connect Disconnected

nuc Up 2025-10-06 10:37:50 Standalone 27.5.0 172.25.112.15:9001
Group: Unassigned No tags Agent 2.27.6
3 stacks 20 containers 18 1 0 0 30 volumes 26 images 8 CPU 16.4 GB RAM
Live connect Disconnected

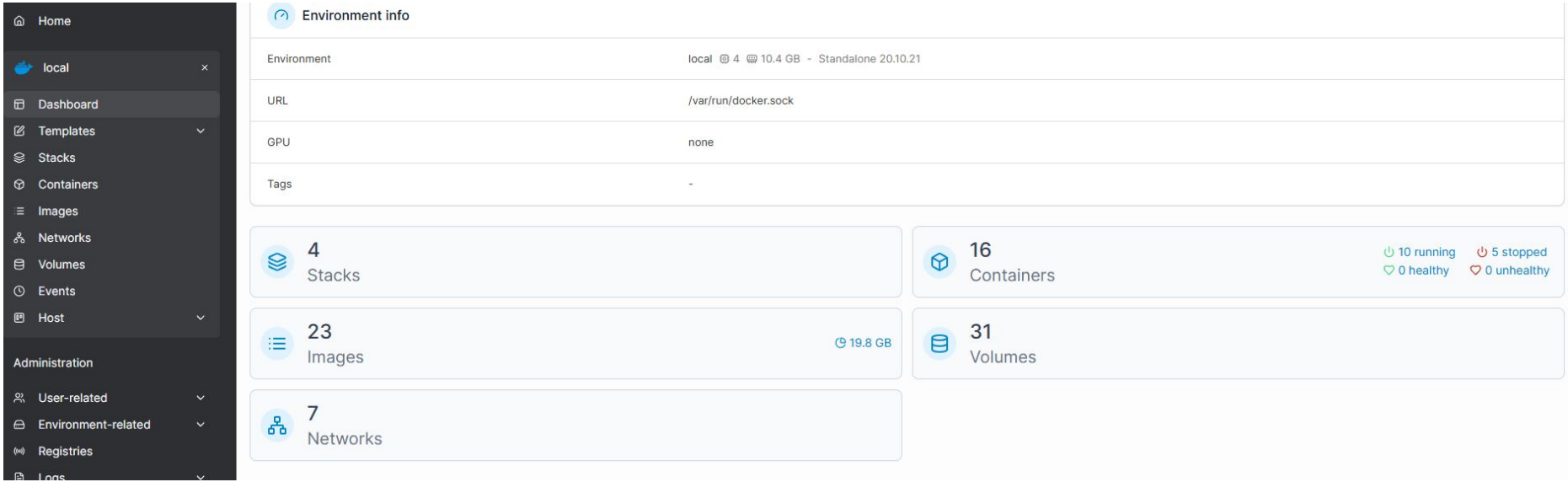
germany Down 2023-09-04 19:34:19 Standalone 23.0.2
Group: Unassigned No tags Agent 2.17.1
0 stacks 2 containers 1 1 0 0 6 volumes 9 images 4 CPU 4.1 GB RAM
Live connect Disconnected

de2_new Up 2025-10-06 10:37:54 Standalone 24.0.7
Group: Unassigned No tags Agent 2.21.2
0 stacks 2 containers 2 0 0 0 0 volumes 2 images 4 CPU 4.1 GB RAM
Live connect Disconnected

pizigbee Up 2025-10-06 10:37:54 Standalone 27.3.1 pizigbee.bier.wtf:9001
Group: Unassigned No tags Agent 2.21.3
0 stacks 3 containers 2 1 0 0 1 volume 6 images 4 CPU 446.4 MB RAM
Live connect Disconnected

Items per page 10

Portainer uebersicht Host



Volumes

Upgrade to Business Edition

PORTAINER.io
COMMUNITY EDITION

- Home
- local
  - Dashboard
  - Templates
  - Stacks
  - Containers
  - Images
  - Networks
  - Volumes
  - Events
  - Host

Volumes

Search... | Remove | + Add volume

Name	Stack	Driver	Mount point	Created	Ownership
13e38c36b203376d827fd7f94ca93ba8bd2a1...	-	local	/var/lib/docker/volumes/1[...]89e1a6c3abb33810482/_data	2025-10-06 05:49:12	administrators
wikijs Unused	-	local	/var/lib/docker/volumes/wikijs/_data	2025-10-03 18:33:17	administrators
3b782d203dfe75e127e70ce73f25fa5e2523f...	-	local	/var/lib/docker/volumes/3[...]85a92667dbc4c371fd6/_data	2025-10-03 18:33:15	administrators
wordpressstack_db_data	wordpressstack	local	/var/lib/docker/volumes/w[...]dpressstack_db_data/_data	2025-06-10 18:32:38	administrators
089c7cee8e98d45f543fdfa9093c6934e4452... Unused	-	local	/var/lib/docker/volumes/0[...]0b6798ab519e3e1726b/_data	2025-06-04 10:14:59	administrators
unifi_config_2025 Unused	-	local	/var/lib/docker/volumes/unifi_config_2025/_data	2025-04-08 21:22:24	administrators
unifi_mongodb Unused	-	local	/var/lib/docker/volumes/unifi_mongodb/_data	2025-04-08 21:22:00	administrators
0a9d9b69df894b59aa49c85b8ae8aa1eaf5b3... Unused	-	local	/var/lib/docker/volumes/0[...]96d11c6a42bb561e213/_data	2025-04-08 21:16:28	administrators
f482bf190afab1fddbc9ca32d0440836e0a40... Unused	-	local	/var/lib/docker/volumes/f[...]df07d0dc59d53c894fd/_data	2025-04-08 21:15:43	administrators

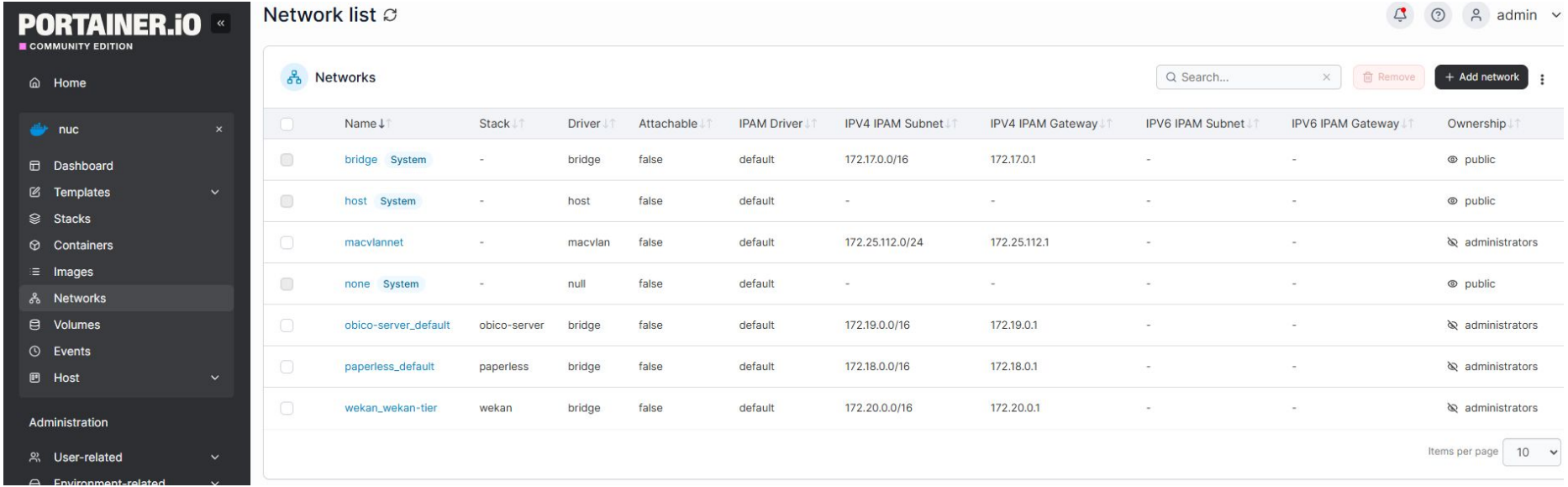
Netzwerk

Die wichtigsten Netzwerkarten

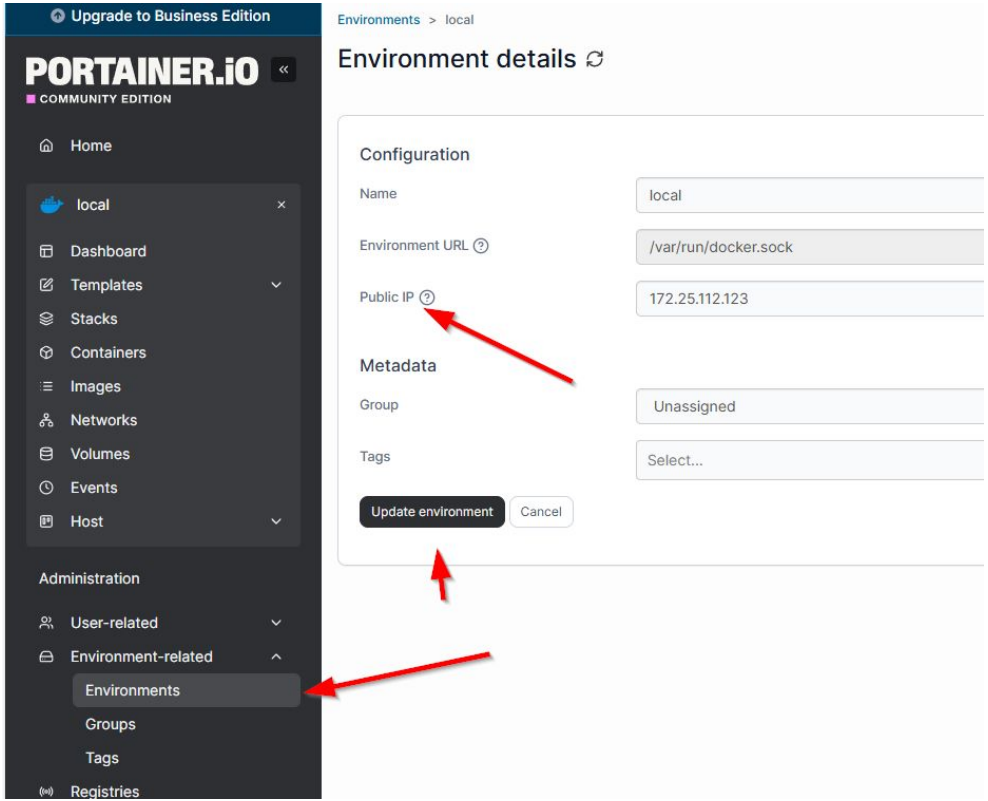
bridge = Netz fuer Container, Container bekommen in diesem Netz eine IP und sprechen via dem Host mit anderen Netzen

host = Der Container verwendet den Netzwerkstack vom Host System (weniger sicher erhoehte Performance, braucht es daher eigentlich nicht fuer normal gebrauch)

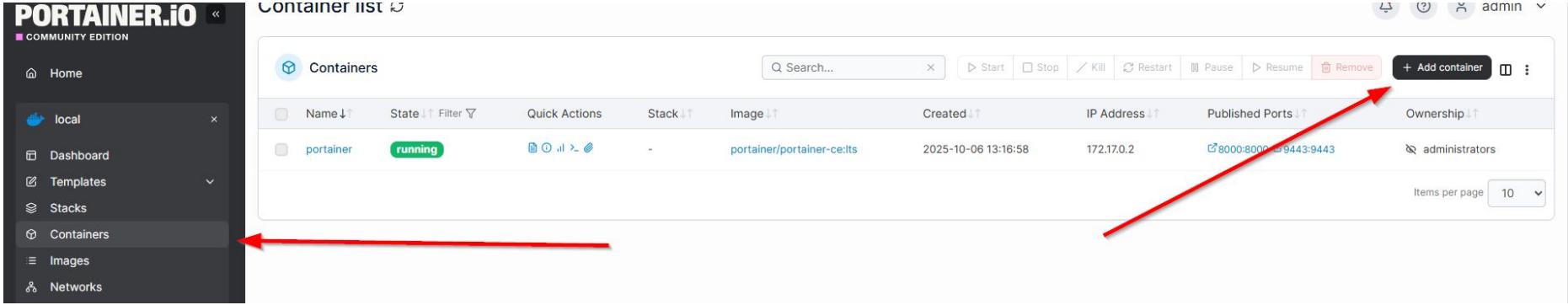
macvlan = der Container bekommt eine eigene MAC und eigene IP, wie eine eigene VM, braucht es fuer Spezielle Anwendungen, zb Smarthome Container



Environment IP setzen



Container hinzufuegen



Wissen, Notizen & Organisation

Container	Einsatzmöglichkeit
Logseq	Markdown-Zettelkasten, lokal und verlinkt
Joplin Server	Synchronisiertes Notizbuch, alternativ zu Evernote
DokuWiki	Einfaches, leichtgewichtiges Wiki
Trilium Notes (beispiel)	Baumstruktur-Notizsystem mit Rich Content
Leanote	Open-Source Notiz-App mit Markdown und Web-GUI

Trilium Notes

Name: trilium

Image Configuration

Registry: Docker Hub (anonymous)

Image: docker.io / triliumnext/notes:develop

Port mapping

Host: 17443 → Container: 8080 TCP UDP

Advanced container settings

Commands & logging | Volumes | Network | Env | Labels

Volume mapping

container: /home/node/trilium-data Volume Bind

→ volume: trilium - local Writable Read-only

Infrastruktur & Netzwerk

Container	Einsatzmöglichkeit
Pi-hole	Werbeblocker im ganzen Netzwerk
Netdata	System- & Netzwerk-Monitoring
Nginx Proxy Manager	Web-Proxy & SSL-Management mit GUI
AdGuard Home	Alternative zu Pi-hole
Glances	Ressourcenüberwachung über Webinterface
Tailscale	VPN für Heimnetz-Zugriff von überall
Dozzle	Live-Log-Viewer für alle Docker-Container
Guacamole	Browser-Zugriff auf RDP, SSH, VNC

Datei- & Medienverwaltung

Container	Einsatzmöglichkeit
Nextcloud	Private Cloud, Kalender, Kontakte, Dateien
Syncthing	Peer-to-Peer Datei-Synchronisierung
Photoprism	KI-gestützte Foto-Galerie
Immich	Alternative zu Google Fotos
TubeArchivist	YouTube-Sammlungen lokal speichern & verwalten
MediaCMS	Eigene Video-Plattform
Samba / cifs	Windows-Netzwerkfreigaben im Container
MinIO	S3-kompatibler Objektspeicher (Backup, Medien)

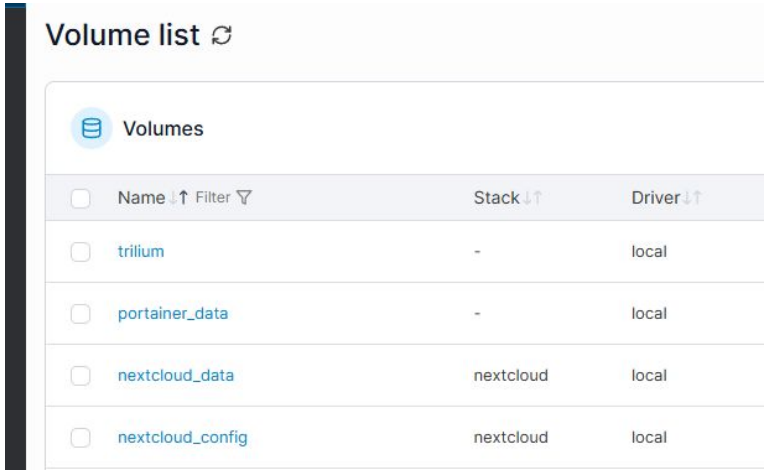
Quellen fuer Docker

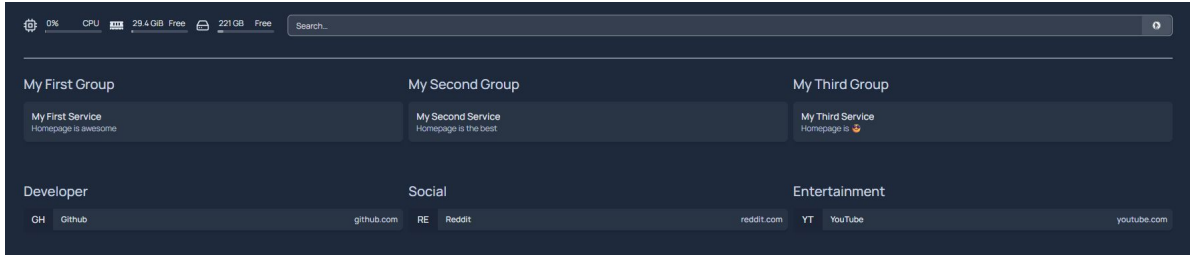
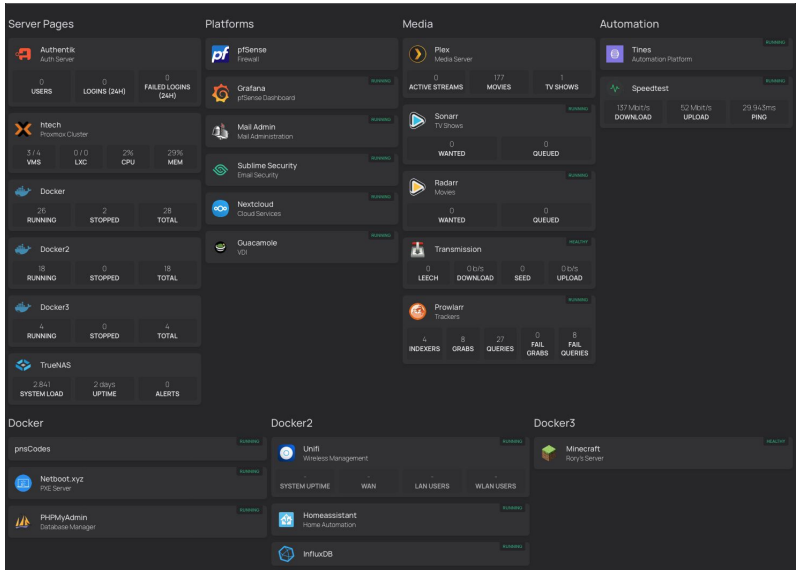
github.com/docker/awesome-compose	Offizielle Docker Compose Beispiele
docs.linuxserver.io	Viele geprüfte Home-/Media-Container
github.com/portainer/templates	Portainer Templates zum Importieren
dockstarter.com	Einfache Einrichtung für Einsteiger
github.com/veggiemonk/awesome-docker	Grosse Liste mit Docker-Tools & Projekten
hub.docker.com	Offizielle Images mit Compose-Beispielen
reddit.com/r/selfhosted	Community für Selfhosting & Docker

NextCloud

```
---
services:
  nextcloud:
    image:
lscr.io/linuxserver/nextcloud:latest
    container_name: nextcloud
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
    volumes:
      - config:/config
      - data:/data
    ports:
      - 4443:443
    restart: unless-stopped

volumes:
  config:
  data:
```



Homepage (Startseite)

```
services:
  homepage:
    image:
ghcr.io/gethomepage/homepage:latest
    container_name: homepage
    environment:
      HOMEPAGE_ALLOWED_HOSTS: "*" #
required, may need port. See
gethomepage.dev/installation/#homepage_
allowed_hosts
      PUID: 1000 # optional, your user id
      PGID: 1000 # optional, your group id
    ports:
      - 3000:3000
    volumes:
      - /home/flo/homepage:/app/config #
Make sure your local config directory exists
      -
/var/run/docker.sock:/var/run/docker.sock
:ro # optional, for docker integrations
    restart: unless-stopped
```