Services

Nextcloud

Installation

Läuft im Docker auf Homeserver

Port: `1313` URL: `https://nextcloud.denode.eu` Datenverzeichnis: `/nextcloud-data` Datenbank: MariaDB (Container `nextcloud-db-1`)

Container:

- nextcloud-app-1
- nextcloud-db-1

wichtige Befehle:

```
docker ps | grep nextcloud
docker logs nextcloud-app-1
docker restart nextcloud-app-1
```

Trusted Domains

Config-Datei in Container

```
docker exec -it nextcloud-app-1 bash
apt update && apt install -y nano
nano /var/www/html/config/config.php
```

Oder mit occ:

```
 docker exec -u www-data nextcloud-app-1 php occ config:system:set trusted_domains 1 --value
="nextcloud.denode.eu"
```

Externe Speicher

Cloud Ordner auf HDD

Scan:

```
docker exec -u www-data nextcloud-app-1 php occ files:scan --path="denode/files/Cloud"
```

Verify:

```
docker exec -u www-data nextcloud-app-1 php occ files_external:verify 2
```

BookStack

Installation

Docker-Compose: `~/bookstack/docker-compose.yml`

Container:

- bookstack-app(Port 4567)
- bookstack-db (MariaDB)

URL: `https://notes.denode.eu`

Login:

- Email: admin@admin.com
- Passwort: `password`

Daten

- Storage: `~/bookstack/data/storage`
- Database: `~/bookstack/data/database`

Struktur

Hierarchie

```
Regal (Shelf)
  └─ Buch (Book)
      └─ Kapitel (Chapter)
          └─ Seite (Page)
```

Empfohlene Struktur

- Server Infrastructure
  - Initial Setup
  - Networking
  - Security
  - Services
  - Backup

Wichtige Befehle

Container:

```
cd ~/bookstack
docker compose ps
docker compose logs app
docker compose restart
```

Backup:

```
cd ~
tar -czf bookstack-backup-$(date +%Y-%m-%d).tar.gz bookstack/
mv bookstack-backup-*.tar.gz /mnt/hdd/backups/
```

Overleaf

Installation

Docker-Compose `~/overleaf/docker-compose.yml`

Container:

- overleaf-app (Port 4568)
- overleaf-mongo (MongoDB 5.0 mit ReplicaSet)
- overleaf-redis

URL: `https://overleaf.denode.eu`

Daten:

- MongoDB: `/mnt/hdd/overleaf/mongo-data`
- Overleaf: `~/overleaf/data/overleaf`
- Redis: `~/overleaf/data/redis`

User Verwaltung

If you're creating your first administrator account we recommend visiting the https://{your-instance-url}/launchpad URL and setting up your account from there.

```
# Legacy docker-compose.yml deployments:
$ docker exec sharelatex /bin/bash -ce "cd /overleaf/services/web && node modules/server-ce-scripts/scripts/create-user --admin --email=joe@example.com"
```

Help: `https://docs.overleaf.com/on-premises/user-and-project-management/user-management`

Wichtige Befehle

Container

```
cd ~/overleaf
docker compose ps
docker compose logs app
docker compose logs mongo
docker compose restart
```

MongoDB ReplicaSet:

```
docker exec overleaf-mongo mongosh --eval "rs.status()"
```

Troubleshooting:

```
# Bei "Transaction" Fehler
docker exec overleaf-mongo mongosh --eval "rs.initiate({_id: 'rs0', members: [{_id: 0, host:
'mongo:27017'}]})"
docker compose restart
```

Vaultwarden

Installation

Läuft in Docker auf Homeserver

Port `8282` URL: Nur über Tailscale MagicDNS

Zugriff:

Warum nicht öffentlich?

-> Passwort-Manager mit allen Passwörtern -> nur privater Zugriff nötig -> Extra Sicherheitsebene

pgcli (datenbank in console)

nützliche befehle

```
\l          -- Alle Datenbanken anzeigen
\dt         -- Alle Tabellen anzeigen
\d tabelle  -- Struktur einer Tabelle zeigen
\c unidb    -- Zu anderer DB wechseln
\i datei.sql -- SQL-Datei ausführen
\e          -- Query in $EDITOR öffnen
\timing     -- Query-Zeit messen (toggle)
\x          -- Expanded display (toggle, gut für breite Tabellen)
```

Im pgcli: \o ausgabe.txt

```
SELECT * FROM studenten;
```

\o -- Output wieder auf Terminal

SQL-Datei ausführen:

```
# queries.sql erstellen, dann:
pgcli unidb < queries.sql

# Oder im pgcli:
\i ~/uni/queries.sql
```

CSV export

```
# Im pgcli:
\copy (SELECT * FROM studenten) TO '/tmp/export.csv' CSV HEADER;
```