

Spickzettel: Docker Volumes & persistente Daten

Ziel

Daten auch nach Container-Neustart oder -Neuerstellung erhalten – für Entwicklung, Datenbanken, Logs & Co.

Was sind Volumes?

- Externe Speicherbereiche für Containerdaten
 - Leben **außerhalb** des Containers
 - Ideal für persistente Daten: Datenbanken, Konfiguration, Uploads
-

Arten von Volumes

Anonyme Volumes

```
docker run -v /app/data myimage
```

Benannte Volumes

```
docker volume create appdata
```

```
docker run -v appdata:/app/data myimage
```

Bind Mounts (Pfad vom Host)

```
docker run -v $(pwd)/data:/app/data myimage
```

- Achtung auf Pfadrechte & OS-Unterschiede
-

Volumes verwalten

```
docker volume ls          # Alle Volumes anzeigen
docker volume inspect NAME # Details (Mountpoint etc.)
docker volume rm NAME     # Entfernen (nur wenn nicht in Nutzung)
```

In docker-compose.yml

```
version: '3.9'
```

```
services:
  db:
    image: postgres
    volumes:
      - dbdata:/var/lib/postgresql/data

volumes:
  dbdata:
```

Best Practices

- Nutze **benannte Volumes**, um Lebensdauer & Wiederverwendbarkeit zu sichern
 - Keine wichtigen Daten direkt im Container speichern
 - Volumes sind unabhängig vom Container – beim Neuaufbau wiederverwendbar
 - Versionierte Daten (z. B. Migrationsskripte) ≠ persistente Daten
-

Docker-Volumes sind die Basis für alle zustandsbehafteten Container – sie trennen Daten und Code sauber.