

Spickzettel: Docker Compose – Mehrere Container verwalten

Was ist Docker Compose?

- Tool zum **Definieren & Starten von Multi-Container-Anwendungen**.
- Konfiguration in einer **docker-compose.yml**-Datei.
- Ideal für lokale Entwicklung mit mehreren Services (z. B. App + Datenbank).

Beispiel: Compose-Datei für Web-App mit Datenbank

```
version: '3.8'
services:
  web:
    build: .
    ports:
      - "8000:8000"
    volumes:
      - ./app
    environment:
      - DEBUG=1
    depends_on:
      - db

  db:
    image: postgres:15
    restart: always
    environment:
      POSTGRES_USER: user
      POSTGRES_PASSWORD: password
      POSTGRES_DB: appdb
    volumes:
      - db-data:/var/lib/postgresql/data

volumes:
  db-data:
```

Wichtige Befehle

```
# Alle Container starten (im Hintergrund)
docker-compose up -d

# Alle Container stoppen
docker-compose down

# Logs anzeigen
docker-compose logs -f

# Einzelnen Service neu starten
docker-compose restart web

# Container bauen (z. B. nach Dockerfile-Änderungen)
docker-compose build
```

Weitere Optionen

- **build:** – Nutzt `Dockerfile` im aktuellen Verzeichnis oder `context:` definieren
- **image:** – Statt selbst bauen: Image direkt verwenden (z. B. `redis`, `mongo`)
- **volumes:** – Daten dauerhaft speichern oder Quellcode mounten
- **depends_on:** – Legt Startreihenfolge der Container fest (aber kein Healthcheck)
- **networks:** – Benutzerdefinierte Netzwerke konfigurieren (optional)

Best Practices

- `.env`-Datei nutzen für Secrets & Umgebungsvariablen
- Services mit `restart:` `always` absichern
- Für Produktion: eigene `docker-compose.prod.yml` definieren
- Datenbank-Initialisierung über `Volumes` + `init.sql` oder `entrypoint`-Script