

Spickzettel: Docker Swarm – Container-Orchestrierung

Was ist Docker Swarm?

- Docker-eigene Lösung zur **Orchestrierung mehrerer Container auf mehreren Hosts**.
- Bietet **Clustering, Load Balancing, Service-Skalierung**.
- In Docker integriert: `docker swarm`, `docker service`, `docker stack`.

Hinweis zur Aktualität (2024)

- **Wird noch unterstützt**, aber **nicht aktiv weiterentwickelt**.
- Docker Inc. empfiehlt langfristig den Umstieg auf **Kubernetes**.
- Für kleine bis mittlere Projekte **lokal oder on-prem** weiterhin nutzbar.

Grundkonzepte

- **Node**: Host im Swarm (Manager oder Worker)
- **Service**: Verteilte Anwendung (mehrere Container)
- **Stack**: Gruppe von Services (definiert in `docker-compose.yml`)
- **Overlay-Netzwerk**: Kommunikation zwischen Services über Hosts hinweg

Einstieg: Swarm starten

```
# Swarm initialisieren
docker swarm init

# Token für Worker anzeigen
docker swarm join-token worker

# Token für Manager anzeigen
docker swarm join-token manager
```

Services erstellen & verwalten

```
# Service erstellen
docker service create --name web -p 80:80 nginx

# Service-Skalierung
docker service scale web=5

# Status prüfen
docker service ls
docker service ps web

# Service stoppen und entfernen
docker service rm web
```

Stack mit Compose-Datei

```
# Stack deployen
docker stack deploy -c docker-compose.yml mein_stack

# Stacks anzeigen / stoppen
docker stack ls
docker stack rm mein_stack
```

Best Practices

- Swarm eher für **einfachere Cluster** ohne externe Abhängigkeiten
- TLS & Zertifikate automatisch verwaltet (ab Init)
- Overlay-Netzwerke für Service-Kommunikation nutzen
- Für komplexe Workloads: auf **Kubernetes** umsteigen

Alternativen

- **Kubernetes (K8s)** – Industriestandard für Container-Orchestrierung
- **Nomad (HashiCorp)** – Leichtgewichtig & einfach
- **Docker Compose + GitHub Actions** – für kleinere automatisierte Deployments