

## Container mit Docker-Netzwerken verbinden & trennen

---

### Mehrere Netzwerke für einen Container nutzen

Ein Docker-Container kann gleichzeitig mit **mehreren Netzwerken** verbunden sein. Das ermöglicht z. B. Kommunikation zwischen verschiedenen Komponenten oder Migration von einem Netzwerk zum anderen.

---

### Beispiel: Container zwischen Netzwerken bewegen

#### Ausgangslage:

Ein Container läuft bereits im benutzerdefinierten Netzwerk **webapp** – z. B.:

```
docker network create webapp
docker run --network webapp -d --name webapp_container nginx
```

### Container mit einem weiteren Netzwerk verbinden

```
docker network connect bridge webapp_container
```

- Verbindet **webapp\_container** zusätzlich mit dem Standardnetzwerk **bridge**.
- 

### Netzwerk prüfen

```
docker network inspect bridge
```

Im Abschnitt **Containers** sollte **webapp\_container** jetzt auftauchen.

---

### Container von einem Netzwerk trennen

```
docker network disconnect webapp webapp_container
```

- Entfernt den Container aus dem Netzwerk **webapp**.
- 

### Aktuelle Netzwerke des Containers anzeigen

```
docker container inspect webapp_container
```

Suche im Abschnitt **NetworkSettings.Networks** – hier steht genau, **mit welchen Netzwerken** der Container noch verbunden ist.

Nach der Trennung vom **webapp**-Netzwerk sollte nur noch **bridge** angezeigt werden.

## Wichtig zu wissen

Aktion	Ergebnis
<b>network connect</b>	Fügt Container einem weiteren Netzwerk hinzu
<b>network disconnect</b>	Entfernt Netzwerkverbindung
<b>inspect</b>	Zeigt alle aktiven Netzwerkverbindungen
Nur manuell möglich	Container springt <b>nicht automatisch</b> zwischen Netzwerken
IP-Adresse pro Netzwerk	Container bekommt <b>eine IP pro Netzwerk</b>

## Beispiel – Kompletter Ablauf

```
docker network create webapp
docker run --network webapp -d --name webapp_container nginx

docker network connect bridge webapp_container
docker network inspect bridge | grep webapp_container

docker network disconnect webapp webapp_container
docker container inspect webapp_container
```

### Ergebnis:

- Container ist jetzt **nur noch im bridge-Netzwerk**.
- Kommunikation im ursprünglichen **webapp**-Netzwerk ist nicht mehr möglich.

Das ist perfekt, um z.B. **Container im Betrieb umzuhängen**, Netzwerkkonfigurationen zu testen oder auch bestimmte **Sicherheitszonen** aufzubauen.