

# 🌐 Docker-Netzwerke – Container vernetzen, verstehen & nutzen

## 🔗 Docker-Netzwerke

- Jeder Container ist **standardmäßig isoliert**.
- Kommunikation ist nur über ein gemeinsames Netzwerk möglich.
- Docker bietet vordefinierte Netzwerke und die Möglichkeit, eigene zu erstellen.

## 📄 Netzwerkarten in Docker

Netzwerkmodus	Beschreibung
bridge	Standardnetzwerk; Container können über interne IPs kommunizieren
host	Container nutzt direkt das Netzwerkinterface des Hosts
none	Container hat <b>kein</b> Netzwerkzugang
Benutzerdefiniert	Manuell erstellte Netzwerke mit DNS-Namensauflösung

## 🔍 Bridge-Netzwerk verstehen

Wenn kein eigenes Netzwerk angegeben wird, landet der Container im Standard-Bridge-Netzwerk.

Eigenschaften:

Attribut	Beispiel
Netzwerkname	bridge
Subnetz	172.17.0.0/16
Gateway	172.17.0.1
Kommunikation	Container können sich anpingen
DNS	Nur in <b>benutzerdef. Netzwerken</b> voll funktionsfähig

## 🔗 Docker-Netzwerke anzeigen & analysieren

```
docker network ls
```

Zeigt alle verfügbaren Netzwerke an.

```
docker network inspect bridge
```

Liefert IP-Bereiche, Container im Netzwerk, Gateway etc.

---

## 🔧 Container im Standardnetzwerk untersuchen

```
docker run -d --name web1 nginx
docker run -d --name ubuntu1 ubuntu sleep 9999
```

IP-Adresse prüfen:

```
docker inspect -f "{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}" web1
```

Ping vom zweiten Container:

```
docker exec -it ubuntu1 bash
apt update && apt install -y iputils-ping
ping <IP-Adresse-von-web1>
```

---

## 🌐 Eigenes Netzwerk erstellen und nutzen

```
docker network create webapp
```

Container gezielt in dieses Netzwerk starten:

```
docker run --network webapp -d --name web2 nginx
docker run --network webapp -it -d --name client ubuntu
```

---

## 💬 Kommunikation über Containernamen testen

```
docker exec -it client bash
apt update && apt install -y iputils-ping
ping web2
```

☒ **Namensauflösung funktioniert**, weil benutzerdefinierte Netzwerke DNS-Support haben.

---

## 🔪 Netzwerk wieder löschen

⚠ Nur möglich, wenn **kein Container** mehr damit verbunden ist:

```
docker network rm webapp
```

## 🧠 Typische Probleme & Hinweise

Problem	Ursache / Lösung
Ping funktioniert nicht	Container sind in unterschiedlichen Netzwerken
Containername kann nicht aufgelöst werden	Nur möglich in benutzerdefinierten Netzwerken
Netzwerk kann nicht gelöscht werden	Container nutzt es noch – erst <b>docker rm</b>

## 🔍 Nützliche Befehle für Debugging & Analyse

```
docker container inspect <container>
docker exec -it <container> bash
ip addr          # Zeigt Netzwerkschnittstellen im Container
ping <ziel>
```

## ☑ Praxisrelevanz

Docker-Netzwerke ermöglichen die Verbindung von:

- **Webservern** und **Datenbanken**
- **Microservices**
- **Frontend + Backend**

Sie sorgen für **Isolation**, **Flexibilität** und **Sicherheit** in modernen Entwicklungsumgebungen.