

1. Ziele

- Verständnis von bridge-Netzwerken

2. Aufgaben

- Definieren Sie das docker-Netzwerk 192.168.100.0/24 mit Gateway 192.168.100.1

```
docker network create --subnet=192.168.100.0/24 --gateway=192.168.100.1 my_network
```

- Überprüfen Sie den Erfolg mit `docker network ls` und `docker network inspect`

```
docker network ls
```

```
docker network inspect my_network
```

```
docker run -d --network=my_network --name=ubuntu1 ubuntu
```

- Starten Sie 2 ubuntu-Container und ordnen Sie diese dem oben erstellten Netzwerk zu:
Der erste Container soll seine IP-Adresse via DHCP erhalten. Der zweite soll die IP-Adresse 192.168.100.100 erhalten

```
docker run -d --network=my_network --name=ubuntu2 --ip=192.168.100.100 ubuntu
```

```
docker network inspect my_network
```

- Überprüfen Sie den Erfolg mit `docker network inspect`

```
docker network inspect my_network
```

- Installieren Sie in beiden Containern die Pakete `iputils-ping` und `net-tools` (für `ifconfig`)

```
docker exec ubuntu1 apt update
```

```
docker exec ubuntu1 apt install -y iputils-ping net-tools
```

```
docker exec ubuntu2 apt update
```

```
docker exec ubuntu2 apt install -y iputils-ping net-tools
```

- Überprüfen Sie, ob sich die beiden Container gegenseitig anpingen können

```
ping ubuntu1
```

```
ping ubuntu2
```

- Stoppen und löschen Sie die Container, löschen Sie das Netzwerk

```
docker exec ubuntu1 ping 192.168.100.100
```

```
docker exec ubuntu2 ping 192.168.100.1
```

3. Hilfsmittel

<https://gbssg.gitlab.io/m347/docker-netzwerke/>

4. Erwartete Resultate

Mit Screenshots dokumentiertes und kommentiertes Vorgehen

Zeit: 45 Minuten