

Reelles VLAN für Docker Container

Diese Anleitung soll helfen, ein am Docker Host anliegendes VLAN für Docker Container verfügbar zu machen. Dadurch können die Container komplett in das Netzwerk eingebunden werden. Einzelne Port-Freigaben für Container werden nicht mehr benötigt. Eine Doppelbelegung von Ports kann so vermieden werden.

Anlegen des VLAN auf dem Host

Konfigurationsdatei erstellen (netplan)

In dieser Anleitung soll das VLAN mit der ID 2, anliegend auf dem Interface `enp2s0f0` für Dockercontainer verfügbar gemacht werden.

```
sudo vi /etc/netplan/mynetplan-config.yaml
```

Inhalt von `mynetplan-config.yaml`

```
network:
  version: 2
  ethernets:
    enp2s0f0: {}
  vlans:
    vlan.4:
      id: 4
      link: enp2s0f0
      dhcp4: no
      addresses: [192.168.3.9/24]
      gateway4: 192.168.3.1
      nameservers:
        addresses: [192.168.3.1]
    vlan.2:
      id: 2
      link: enp2s0f0
      dhcp4: no
      addresses: [192.168.1.10/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [192.168.1.1]
```

Konfiguration übernehmen `sudo netplan apply`

Konfigurationsdatei erstellen (Synology NAS)

In dieser Anleitung soll das VLAN mit der ID 2, anliegend auf dem Interface bond0 für Dockercontainer verfügbar gemacht werden.

```
cd /etc/sysconfig/network-scripts  
sudo vi ifcfg-bond0.2
```

Inhalt von ifcfg-bond0.2

```
DEVICE=bond0.2  
VLAN_ROW_DEVICE=bond0  
VLAN_ID=2  
ONBOOT=yes  
BOOTPROTO=static  
IPADDR=192.168.1.10  
NETMASK=255.255.255.0
```

Netzwerkdienste neustarten `sudo /etc/rc.network restart`

MACVLAN erstellen

```
ip link del MYLAN  
ip link add MYLAN link bond0.2 type macvlan mode bridge  
ip addr add 192.168.1.80/32 dev MYLAN  
ip link set MYLAN up  
ip route add 192.168.1.80/28 dev MYLAN
```

Docker Netzwerk erstellen

```
docker network create -d macvlan \  
  --subnet=192.168.1.0/24 \  
  --gateway=192.168.1.1 \  
  --ip-range=192.168.1.80/28 \  
  --aux-address 'host=192.168.1.80' \  
  -o parent=bond0.2 MYLAN
```

Testen

```
$ docker run --rm -itd \
  --network MYLAN \
  --name my-alpine \
  alpine:latest \
  ash
```

Netzwerk vom neuen Container prüfen:

```
docker container inspect my-alpine

...truncated...
"Networks": {
  "MYLAN": {
    "IPAMConfig": null,
    "Links": null,
    "Aliases": [
      "12f5c3c9ba5c"
    ],
    "NetworkID":
    "c6203997842e654dd5086abb1133b7e6df627784fec063afcbee5893b2bb64db",
    "EndpointID":
    "aa08d9aa2353c68e8d2ae0bf0e11ed426ea31ed0dd71c868d22ed0dcf9fc8ae6",
    "Gateway": "192.168.1.1",
    "IPAddress": "192.168.1.81",
    "IPPrefixLen": 24,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "MacAddress": "02:42:ac:10:56:02",
    "DriverOpts": null
  }
}
...truncated
```

Quellen

- <http://www.stueben.de/mehrere-vlans-mit-einem-netzwerkadapter-auf-der-synology/>
- <https://docs.docker.com/network/network-tutorial-macvlan/>

From:
<https://wiki.koechr.de/> - **KoeChrs DokuSpace**

Permanent link:
https://wiki.koechr.de/public/anleitungen/real_vlan_fur_docker_container

Last update: **2020/09/27 14:53**

