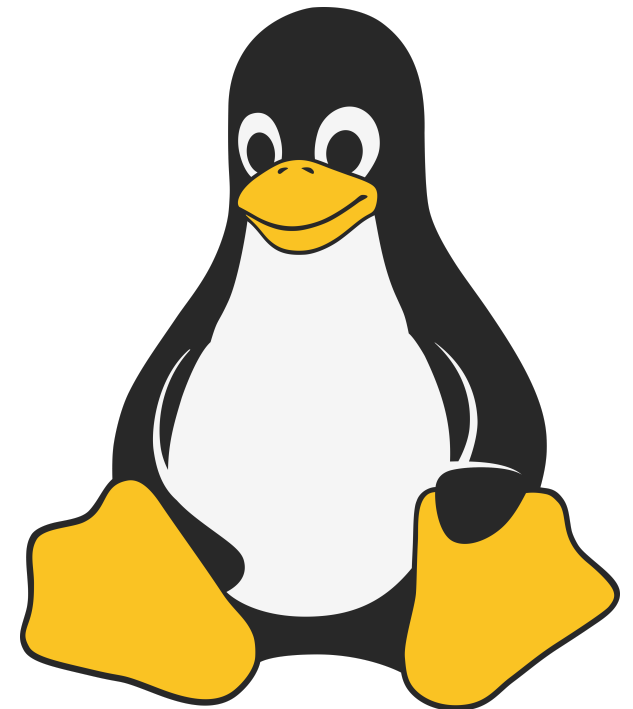


Linux-VM Netzwerk-Konfiguration



Hyper-V Netzwerk-Konfiguration mit Default Switch

Bei der Einrichtung der VM mit Hyper-V wurde eine virtuelle Netzwerkkarte (Default Switch) erstellt. Default Switch stellt ein Netzwerk auf der Basis von NAT (Network Address Translation) zur Verfügung. Damit kann die VM direkt auf den Host und andere VMs zugreifen. Zugriffe auf das Internet werden indirekt und durch das NAT realisiert. Damit kann die VM nicht von außen erreicht werden und die inneren IP-Adressen sind von außen nicht sichtbar. VMs können dennoch von innen auf das Internet zugreifen.

Interne Netzwerk-Konfiguration testen

Auf den folgenden Folien wird die Netzwerk-Konfiguration der VM und des Hosts abgefragt.

- auf dem Windows-Host mit dem Kommando `ipconfig`
- auf der Linux-VM mit dem Kommando `ip address`

IP-Konfiguration des Windows-Hosts

```
C:\Users\hermann>ipconfig /all
```

Windows-IP-Konfiguration

```
    Hostname . . . . . : tuxp14-win11
```

```
    ...
```

Ethernet-Adapter vEthernet (Default Switch):

```
    Beschreibung . . . . . : Hyper-V Virtual Ethernet Adapter
```

```
    Physische Adresse . . . . . : 00-15-5D-53-60-78
```

```
    DHCP aktiviert . . . . . : Nein
```

```
    Autokonfiguration aktiviert . . . : Ja
```

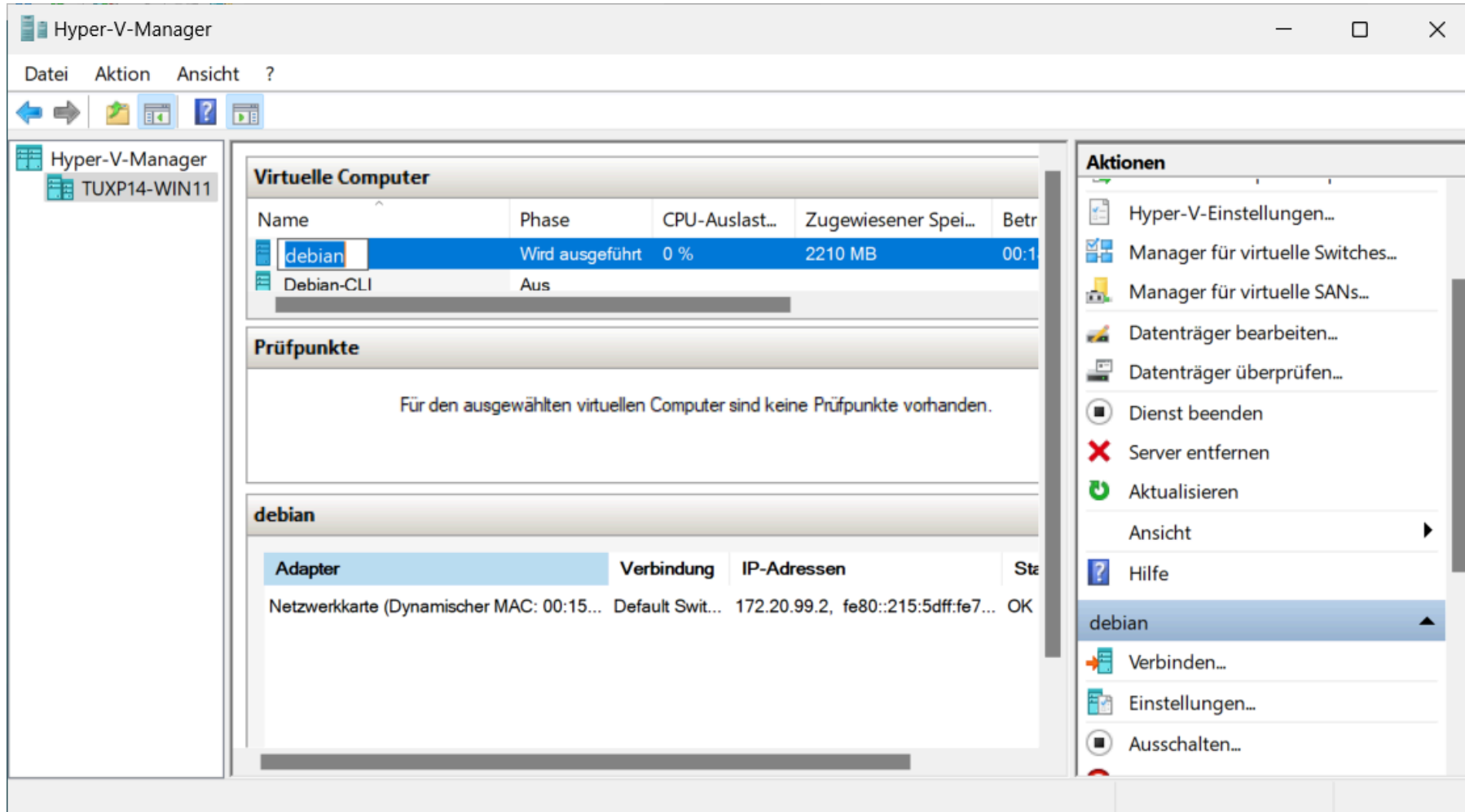
```
    Verbindungslokale IPv6-Adresse . : fe80::d141:b7ce:2480:cd68%17(Bevorzugt)
```

```
    IPv4-Adresse . . . . . : 172.24.144.1(Bevorzugt)
```

```
    Subnetzmaske . . . . . : 255.255.240.0
```

```
    ...
```

IP-Konfiguration der VM (Hyper-V-Manager)



Linux-IP-Konfiguration (in der VM)

```
hermann@debian:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 ...
    link/ether 00:15:5d:7a:a5:02 brd ff:ff:ff:ff:ff:ff
    inet 172.20.99.2/20 brd 172.24.159.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe7a:a502/64 scope link
        valid_lft forever preferred_lft forever
```

Linux-IP-Konfiguration (in der VM) - Kurzform

```
hermann@debian:~$ ip -br address
lo          UNKNOWN      127.0.0.1/8  ::1/128
eth0        UP           172.23.57.184/20 fe80::215:5dff:fe7a:a502/64
```

Ping-Test

Auf den folgenden Folien wird der `ping`-Test zwischen dem Windows-Host und der Linux-VM durchgeführt.

- Ping vom Windows-Host zur Linux-VM
- Ping von der Linux-VM zum Windows-Host

Ping vom Windows-Host zur Linux-VM (IPv6)

```
C:\Users\hermann>ping debian
```

```
Ping wird ausgeführt für debian.local [fe80::215:5dff:fe7a:a502%17] mit 32 Bytes Daten:
```

```
Antwort von fe80::215:5dff:fe7a:a502%17: Zeit=1ms
```

```
Antwort von fe80::215:5dff:fe7a:a502%17: Zeit<1ms
```

```
Antwort von fe80::215:5dff:fe7a:a502%17: Zeit=1ms
```

```
Antwort von fe80::215:5dff:fe7a:a502%17: Zeit=1ms
```

```
Ping-Statistik für fe80::215:5dff:fe7a:a502%17:
```

```
  Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0  
  (0% Verlust),
```

```
Ca. Zeitangaben in Millisek.:
```

```
  Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms
```

Beim **ping** auf den Hostnamen wird die IPv6-Adresse verwendet.

Ping vom Windows-Host zur Linux-VM (IPv4)

```
C:\Users\hermann>ping 172.20.99.2
```

```
Ping wird ausgeführt für 172.20.99.2 mit 32 Bytes Daten:
```

```
Antwort von 172.20.99.2: Bytes=32 Zeit<1ms TTL=64
```

```
Antwort von 172.20.99.2: Bytes=32 Zeit<1ms TTL=64
```

```
Antwort von 172.20.99.2: Bytes=32 Zeit<1ms TTL=64
```

```
Antwort von 172.20.99.2: Bytes=32 Zeit=1ms TTL=64
```

```
Ping-Statistik für 172.20.99.2:
```

```
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0  
    (0% Verlust),
```

```
Ca. Zeitangaben in Millisek.:
```

```
    Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms
```

Ping von der Linux-VM zum Windows-Host

```
hermann@debian:~$ ping -c4 tuxp14-win11
PING tuxp14-win11.mshome.net (172.24.144.1) 56(84) bytes of data.
64 bytes from tuxp14-win11.mshome.net (172.24.144.1): icmp_seq=1 ttl=128 time=0.323 ms
64 bytes from tuxp14-win11.mshome.net (172.24.144.1): icmp_seq=2 ttl=128 time=0.721 ms
64 bytes from tuxp14-win11.mshome.net (172.24.144.1): icmp_seq=3 ttl=128 time=0.563 ms
64 bytes from tuxp14-win11.mshome.net (172.24.144.1): icmp_seq=4 ttl=128 time=1.12 ms

--- tuxp14-win11.mshome.net ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms rtt min/avg/max/mdev = 0.323/0.681/1.117/0.288 ms
```

`ifconfig` -Kommando

Gerne wird die IP-Konfiguration mit dem klassischen Unix-Werkzeug `ifconfig` (Interface Configuration) abgefragt oder konfiguriert. Dieses Kommando ist neben weiteren Kommandos wie `route`, `netstat` und `arp` im Paket `net-tools` enthalten und kann mit `apt` nachinstalliert werden.

`ifconfig` benötigt Root-Rechte und ist deshalb mit `sudo` aufzurufen.

Paket *net-tools* installieren

```
hermann@debian:~$ sudo apt install net-tools
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Statusinformationen werden eingelesen... Fertig
Die folgenden NEUEN Pakete werden installiert:
  net-tools
0 aktualisiert, 1 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
Es müssen 243 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 1.001 kB Plattenplatz zusätzlich benutzt.
Holen:1 http://deb.debian.org/debian bookworm/main amd64 net-tools amd64 2.10-0.1 [243 kB]
Es wurden 243 kB in 0 s geholt (1.696 kB/s).
Vormals nicht ausgewähltes Paket net-tools wird gewählt.
(Lese Datenbank ... 137093 Dateien und Verzeichnisse sind derzeit installiert.)
Vorbereitung zum Entpacken von .../net-tools_2.10-0.1_amd64.deb ...
Entpacken von net-tools (2.10-0.1) ...
net-tools (2.10-0.1) wird eingerichtet ...
Trigger für man-db (2.11.2-2) werden verarbeitet ...
```

Infos zu *net-tools* (`apt show` oder `apt info`)

```
hermann@debian:~$ apt show net-tools
Package: net-tools
Version: 2.10-0.1
...
Download-Size: 243 kB
APT-Manual-Installed: yes
APT-Sources: http://deb.debian.org/debian bookworm/main amd64 Packages
Description: Netzwerk-Werkzeugsatz NET-3
Dieses Paket enthält die wichtigen Werkzeuge zur Kontrolle des
Netzwerk-Subsystems des Linux-Kernels: arp, ifconfig, netstat,
rarp, nameif und route. Zusätzlich enthält es Werkzeuge für
spezielle Arten von Netzwerkhardware (plipconfig, slattach, mii-tool) und
für Feinheiten der IP-Konfiguration (iptunnel, ipmaddr).
...
```

ifconfig (für alle Interfaces)

```
hermann@debian:~$ sudo ifconfig
eth0: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC>  mtu 1500
    inet 172.23.57.184  netmask 255.255.240.0  broadcast 172.23.63.255
    inet6 fe80::215:5dff:fe7a:a502  prefixlen 64  scopeid 0x20<link>
    ether 00:15:5d:7a:a5:02  txqueuelen 1000  (Ethernet)
    ...

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop  txqueuelen 1000  (Lokale Schleife)
    ...
```

`ifconfig` (für Interface `eth0`)

```
hermann@debian:~$ sudo ifconfig eth0
eth0: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC>  mtu 1500
    inet 172.23.57.184  netmask 255.255.240.0  broadcast 172.23.63.255
    inet6 fe80::215:5dff:fe7a:a502  prefixlen 64  scopeid 0x20<link>
    ether 00:15:5d:7a:a5:02  txqueuelen 1000  (Ethernet)
    RX packets 497  bytes 455348 (444.6 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 546  bytes 165138 (161.2 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```