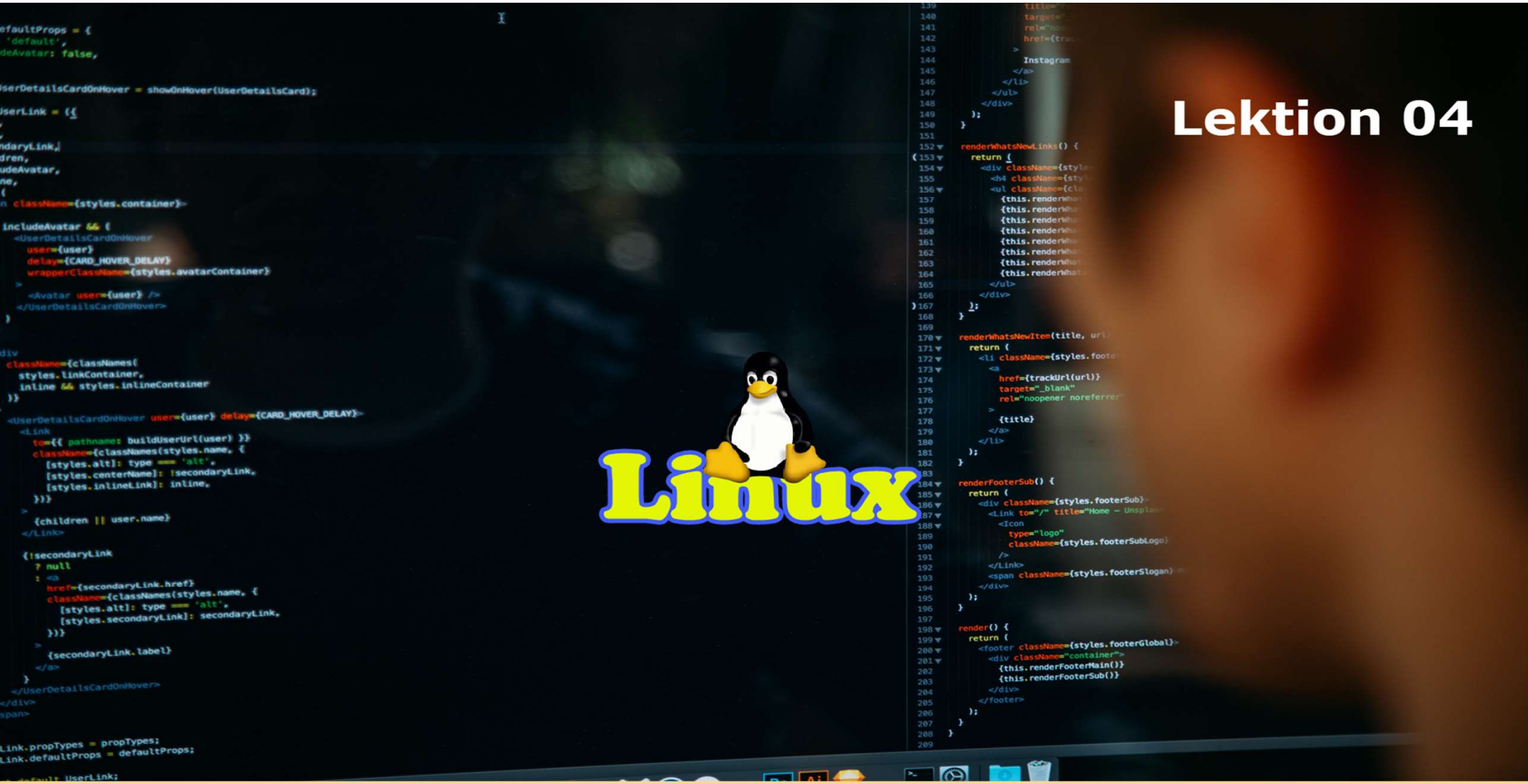

Lektion 04
Linux

Linux-Server mit SSH fernwarten

- Unter Debian und deren Derivaten nutzt man die freie Variante von Secure Shell (SSH) nämlich OpenSSH
- Es ist üblich Linux- und Windows-Server über SSH zu administrieren
- Für die Netzwerkverbindung wird kaum Bandbreite benötigt, da SSH eine einfache Konsole anbietet
- Durch SSH entsteht der Eindruck, als säße man direkt vor dem Linux-Server
- Mit SSH kann man alle Tätigkeiten ausführen, die auch lokal auf der Konsole möglich sind
- Es ist sogar möglich, andere unsichere Protokolle durch SSH zu tunneln

Konfiguration

- Der SSH-Client wird bei der normalen Installation mit installiert
- Der SSH-Server muss auf dem Rechner installiert werden, der als Server eingerichtet werden soll, dieser heißt „deb-s1“ (für Debian-Server1)
- Als Benutzer root führt man auf der Konsole von deb-s1 die Installation des Pakets *openssh-server* durch folgende Eingabe aus:
- `# apt install openssh-server`

```
helmut@deb-s1:~$ su -
Passwort:
root@deb-s1:~# apt install openssh-server
```

Windows als Client

- Auf Linux- und Apple-Systemen wird der Client automatisch installiert und steht für die Nutzung bereit
- Für Windows-Rechner als SSH-Client nutzt man z.B. die Powershell. Es empfiehlt sich gesondert die Nutzung des Programmpakets putty, welches man von putty.org als msi-Datei herunterladen kann.
- Mit putty lässt sich die Verwaltung mehrerer Server vereinfachen und eine symmetrische Verschlüsselung erstellen
- Dazu später mehr

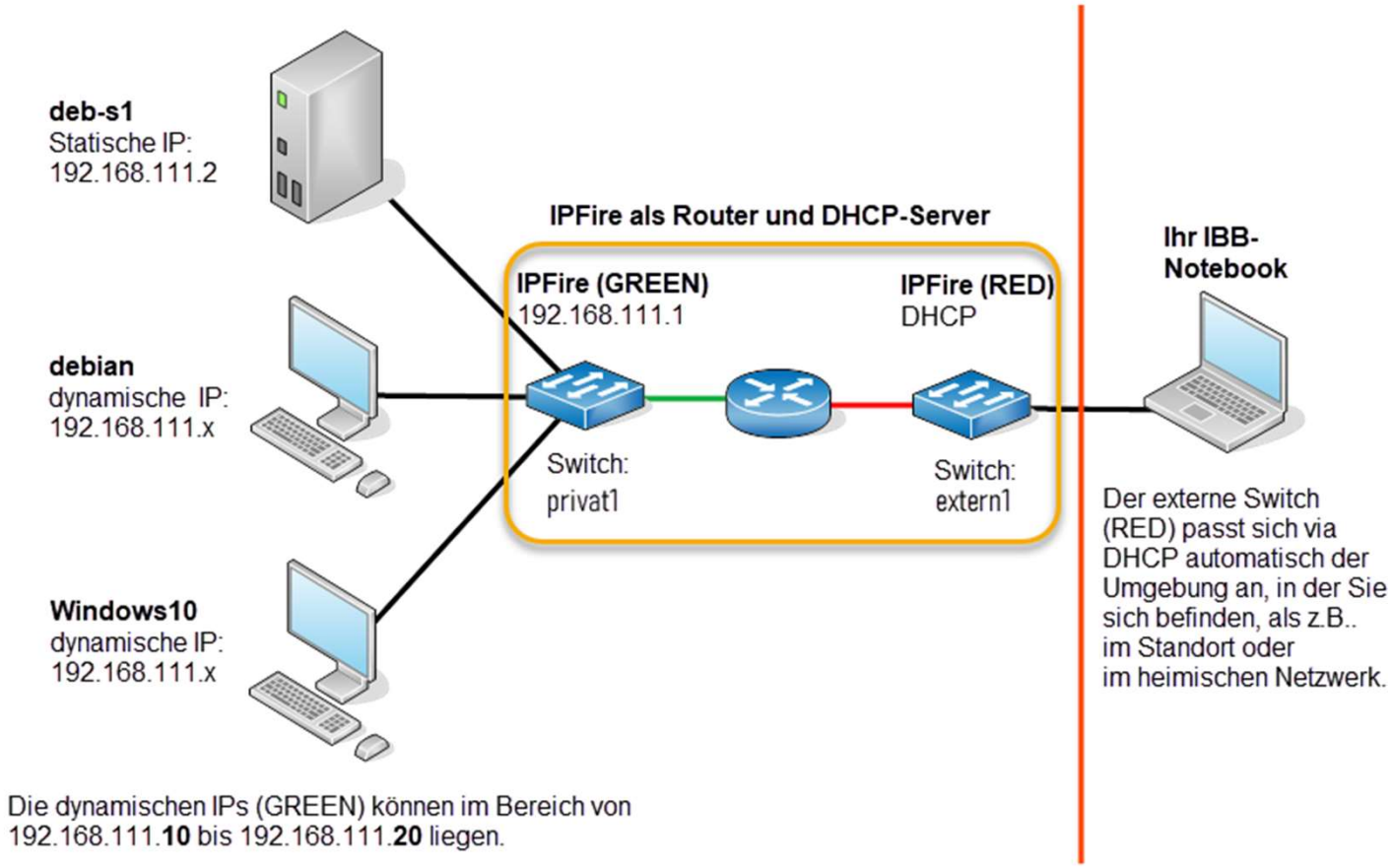
Debian als Client

- Unsere VM „debian“ (mit der GUI) soll als Client fungieren (später dann auch Ihr virtueller Windows-PC in der Hyper-V)
- Beide Linux-VM sind momentan mit dem „Default-Switch“ von Hyper-V nach außen über NAT verbunden
- Um die jeweilige IP-Adresse der VM zu erhalten, gibt es mehrere Methoden z.B.
  - `ip address show` (liefert die Netzwerkkonfiguration), Kurzform `ip a`
  - `ip route show` (liefert die Routingtabelle), Kurzform `ip r`
  - Alternativ auch `hostname –I`
  - Der früher benutzte Befehl `ifconfig` funktioniert nicht mehr, kann aber aktiviert werden

Besondere Situation bei der IBB

- Im Normalfall stehen unsere Clients und Server in einer stabilen Umgebung, so dass immer der gleiche IP-Adressraum vorliegt.
- Durch den stetigen Wechsel zwischen dem Standort und der häuslichen Umgebung wechselt die Zuordnung der IP-Adressen, was vor allen den Server deb-s1 betrifft, da er, wie alle Server, mit einer statischen IP betrieben wird.
- Das folgende Konzept findet man in der Praxis so nicht vor:
  - Zum Einsatz kommt ein virtueller Router (ipfire), der über DHCP im lokalen Netz der Umgebung steht, also im Subnetz des Standorts bzw. im heimischen Netz (externer Switch)
  - Sein anderes „Bein“ befindet sich in einem privaten Subnetz (privater Switch). In diesem Subnetz befinden sich auch die beiden Linux-VM (debian und deb-s1). Die Windows-VM muss bei Bedarf auch in das private Subnetz eingebunden werden.

Aufbau des Netzwerks



Die dynamischen IPs (GREEN) können im Bereich von 192.168.111.**10** bis 192.168.111.**20** liegen.

Vorbereitungen: privaten Switch erstellen

- Die beiden virtuellen Linux-Rechner (debian und deb-s1) und optional der virtuelle Windows-PC (Windows10 oder 11) werden in einem privaten Subnetz, über einen privaten Switch betrieben
- Erstellen Sie über den „Manager für virtuelle Switches …“ einen privaten Switch mit der Bezeichnung „privat1“
- Hierbei wird keine Schnittstellenkarte benötigt, weil in diesem Netzwerk die beiden Linux-VM und die Windows-VM untergebracht werden
- IPFire als Router, wird in beiden Netzwerken, also „privat1“ (GREEN) und „extern1“ (RED) präsent sein, daher muss auch ein externer Switch erstellt werden

Vorbereitungen: externen Switch erstellen

- Für die Verbindung zum Host-Rechner benötigt ipfire im Netz „Red“ einen externen Switch.
- Erstellen Sie im „Manager für virtuelle Switches …“ einen externen Switch mit der Bezeichnung „extern1“
- Achten Sie dabei auf die Auswahl der von Ihnen verwendeten Netzwerkkarte:
  - Wenn Sie den zusätzlichen Dockadapter für Ihr IBB-Laptop einsetzen und dort ihr Netzwerkkabel eingesteckt haben, ist in der Netzwerkadapter-Bezeichnung der Eintrag „USB“ enthalten
  - Ist Ihr LAN-Kabel direkt im Laptop eingesteckt, ist in der Netzwerkadapter-Bezeichnung der Eintrag „PCI“ enthalten
  - Falls Sie sich in einem WLAN-Netz befinden, ist in der Netzwerkadapter-Bezeichnung der Eintrag „WIFI“ enthalten

Vorbereitungen: externen Switch erstellen

- Beim Erstellen des externen Switches können Sie kurzzeitig die Netzwerkverbindung verlieren
- Wenn Sie periodisch zwischen Ihrem Standort und zuhause wechseln und zuhause WLAN nutzen, empfiehlt es sich zwei externe Switches zu erstellen.
  - z.B. im Standort „extern1“ mit dem USB-Dock und zuhause „extern2“ mit WIFI
- Je nachdem, wo sie sich befinden, tauschen Sie die externen Switches in den Einstellungen der VM „ipfire“ einfach aus und starten Sie ipfire danach
- Die Einstellung des privaten Switches werden nicht verändert.

Die VM ins private Netzwerk bringen

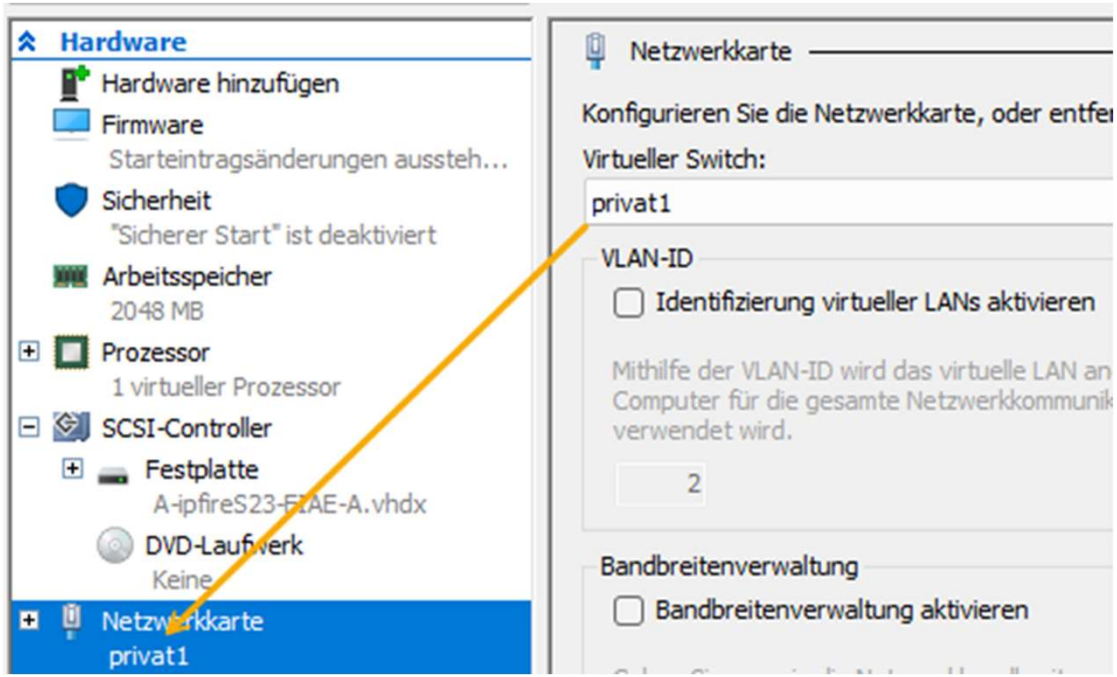
- Lassen Sie die beiden Linux-VM und die Windows-VM ausgeschaltet.
- Klicken Sie die VM „debian“ an und gehen Sie danach in den Menüpunkt „Einstellungen“
- Aktivieren Sie den Eintrag „Netzwerkkarte“ in der linken Spalte und wählen Sie unter dem Eintrag „Virtueller Switch“ den Eintrag „privat1“ aus.
- Bestätigen Sie die Auswahl mit <OK>
- Verfahren Sie auf die gleiche Art mit der VM „deb-s1“ und optional auch mit Ihrer Windows-VM.

Neue VM für IPFire erstellen

- Erstellen Sie eine neue VM mit dem
  - Namen „ipfire“,
  - Generation2,
  - Arbeitsspeicher: minimal 2048 MB,
  - Netzwerk „Default Switch“ (nur für die Erstellung der VM), Festplattengröße: 10 GB,
  - Installationsoptionen: „Betriebssystem von einer startbaren CD/DVD-ROM installieren, Abbilddatei (ISO), Klicken Sie auf die Schaltfläche <Durchsuchen> und wählen Sie die .iso-Datei von IPFire aus
  - Klicken Sie danach auf <Fertig stellen>
- Die VM mit dem Namen ipfire wird erstellt
- Starten Sie noch nicht mit der Installation von IPFire!!!
- Es müssen noch Einstellungen angepasst werden

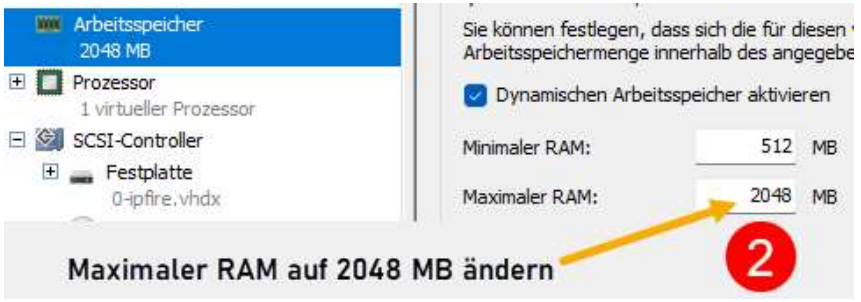
alle VM mit dem privaten Switch ausstatten

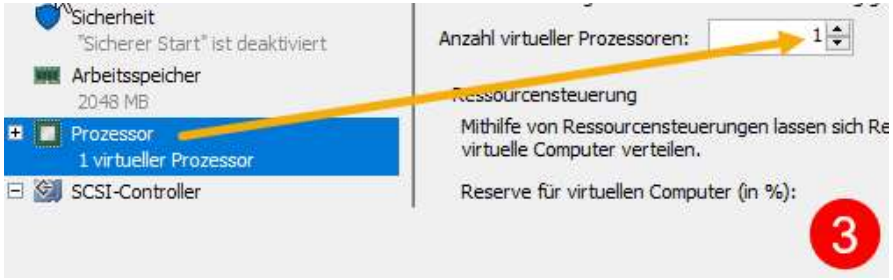
- Die abgebildete Einstellung nehmen Sie auf allen VM vor, also auf
  - deb-s1
  - debian
  - ipfire
  - und ggf. auch auf Ihrer Windows-VM



Nacharbeiten an der virtuellen Maschine

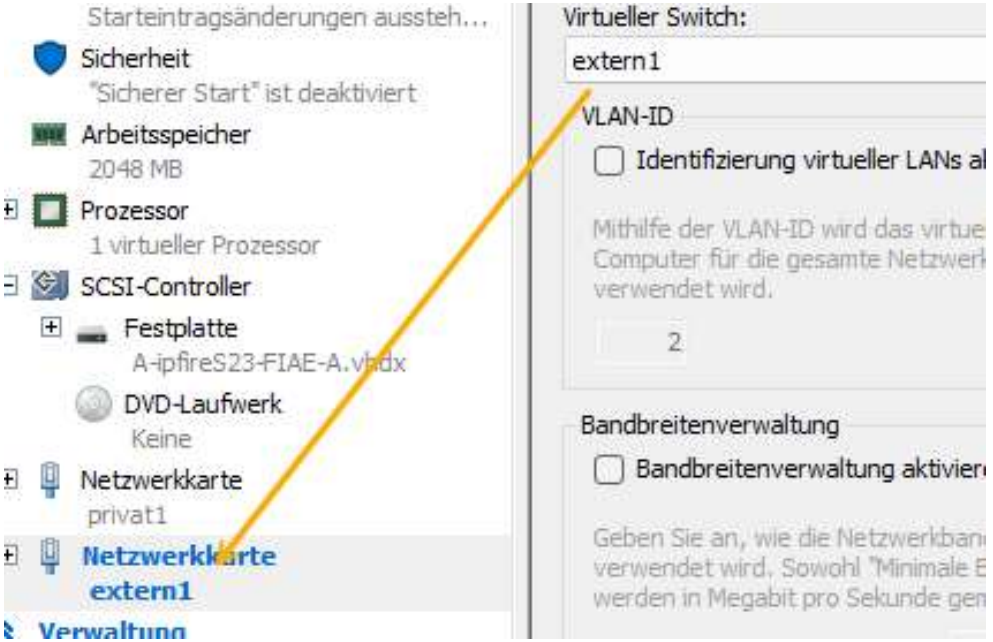
- Öffnen Sie in der Hyper-V die „Einstellungen“ der virtuellen Maschine und ändern Sie folgende Einträge:





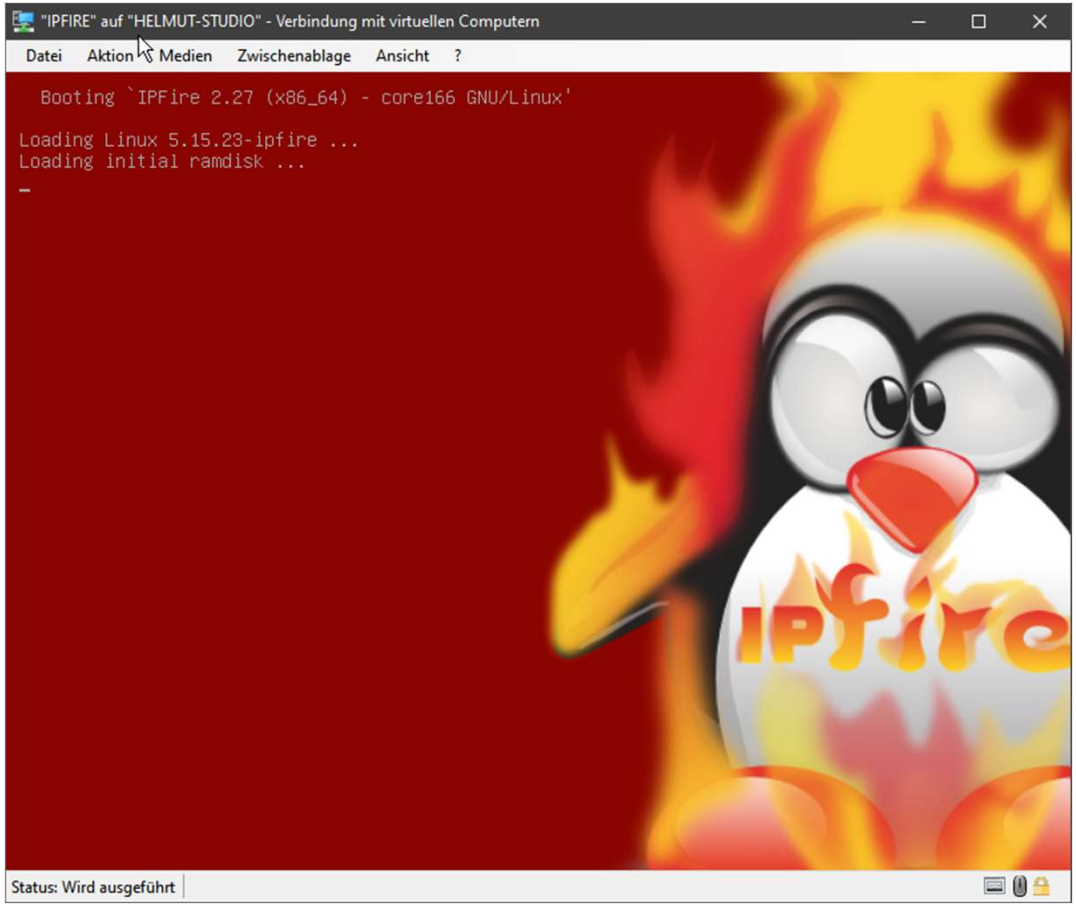
Eine zweite NIC wird benötigt

- Da die VM „ipfire“ vor allem als Router fungiert, ist es notwendig eine weitere virtuelle Netzwerkkarte (NIC) zu installieren.
  - Markieren Sie die VM „ipfire“ und klicken Sie auf „Einstellungen“, dann „Hardware inzufügen“
  - Wählen Sie „Netzwerkkarte“ aus der Liste aus und klicken Sie auf <Hinzufügen>
  - Klicken Sie die neue NIC an und übergeben Sie ihr den virtuellen Switch „extern1“
- Damit sind die Vorarbeiten für „ipfire“ abgeschlossen und die Installation kann mit Doppelklick auf „Verbinden“ beginnen



VM ipfire starten und installieren

- Starten Sie die VM „ipfire“ in Hyper-V wie üblich über „Verbinden“ und anschließend im Verbindungsfenster durch Klick auf <Starten>
- Es öffnet sich das Startfenster von IPFire



VM ipfire starten und installieren

- Wie bei jeder Installation werden die lokalen Parameter abgefragt, damit das deutsche Tastatur-Layout genutzt werden kann
- Um an den Eintrag „de“ zu gelangen muss man ziemlich weit nach oben scrollen (Taste „d“ geht schneller)
- Mit Hilfe der Tab-Taste bewegt man sich schnell zur Schaltfläche <OK>

VM ipfire starten und installieren

- Natürlich darf die Zeitzoneneinstellung auch nicht fehlen

VM ipfire starten und installieren

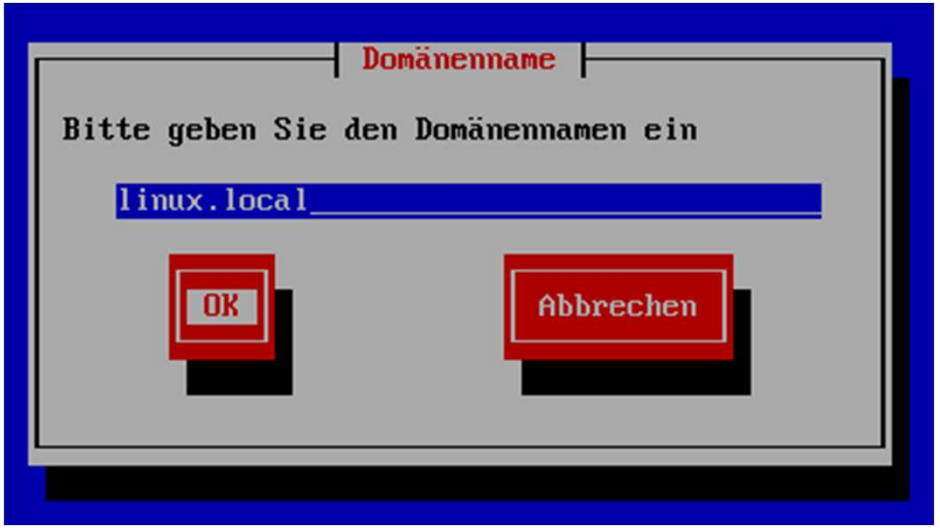
- Natürlich darf die Zeitzoneneinstellung auch nicht fehlen,
- also Europe/Berlin

VM ipfire starten und installieren

- Der Hostname kann frei gewählt werden
- Ich habe die Voreinstellung belassen

VM ipfire starten und installieren

- Auch den Domänennamen kann man frei wählen
- Ich habe mich für „linux.local“ entschieden
- Die Domäne ist für die VM‘s im privaten Netz wichtig
- Meine Empfehlung: Nutzen Sie hier den gleichen Domänennamen, um später nicht durcheinander zu kommen



VM ipfire starten und installieren

- Wie üblich benötigt der Benutzer „root“ ein Passwort
- Dieses ist 2x einzugeben. Es werden keine Zeichen angezeigt
- Zur Wiederholungszeile wechseln Sie auch mit der Tab-Taste

VM ipfire starten und installieren

- Auch der Standard-Benutzer „admin“ benötigt ein Passwort
- Denken Sie daran, dass hier einfache Passwörter möglich sind; in der produktiven Umgebung bitte nicht

VM ipfire starten und installieren

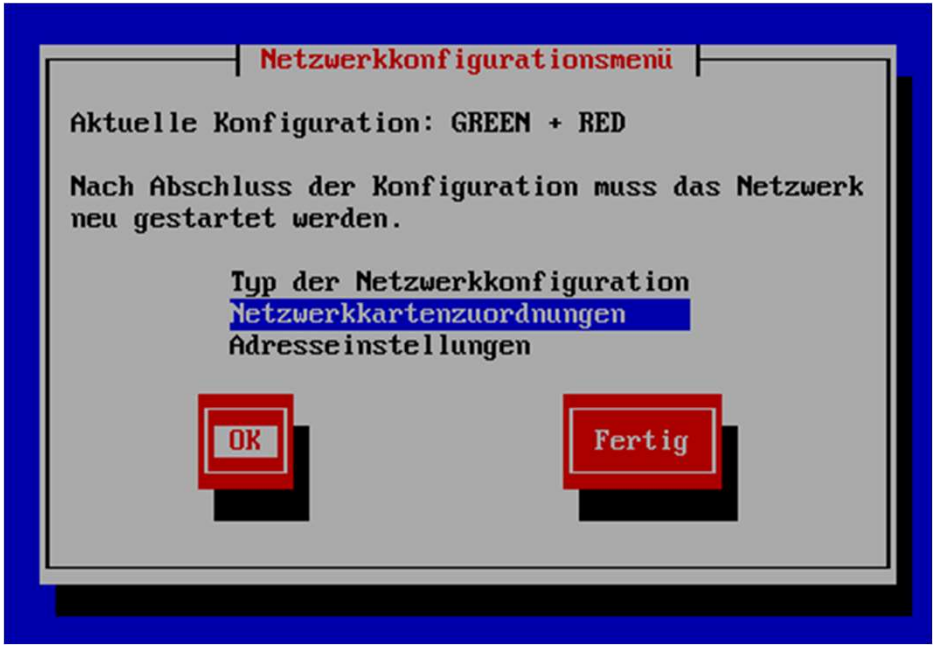
- Die Grundeinstellung ist auf eine einfache Routingfunktion aus zwei Netzen aufgebaut
  - GREEN – das private Netz
  - RED – das externe Netz
- Über die Auswahl „Typ der Netzwerkkonfiguration“ kann man die Auswahl ändern oder bestätigen.
- Wir bleiben bei GREEN + RED

VM ipfire starten und installieren

- Hier wird die Liste der Konfigurationstypen angezeigt
- Wir bestätigen GREEN + RED
- Wer später weiter mit IPFire experimentieren möchte, kann diese Einstellung jederzeit erweitern.

VM ipfire starten und installieren

- Als nächstes erfolgt die Zuordnung der Netzwerkkarten zu den beiden Netzen RED bzw. GREEN
- Sollten Sie die Netzwerkkarten falsch zuordnen, ist das kein großes Problem
- Sie brauchen dann nur in den Einstellungen der Hyper-V die Switches der jeweils anderen NIC zuweisen



VM ipfire starten und installieren

- Die Bezeichnungen der NIC‘s ist etwas ungewöhnlich
- Rechts werden die MAC-Adressen angezeigt
- Leider gibt uns die Hyper-V keine Auskunft darüber
- Die erste NIC habe ich für das Subnetz GREEN gewählt, danach mit <Auswählen> bestätigen

VM ipfire starten und installieren

- Die Zuweisung wird angezeigt und jetzt wird noch die NIC für RED zugewiesen

VM ipfire starten und installieren

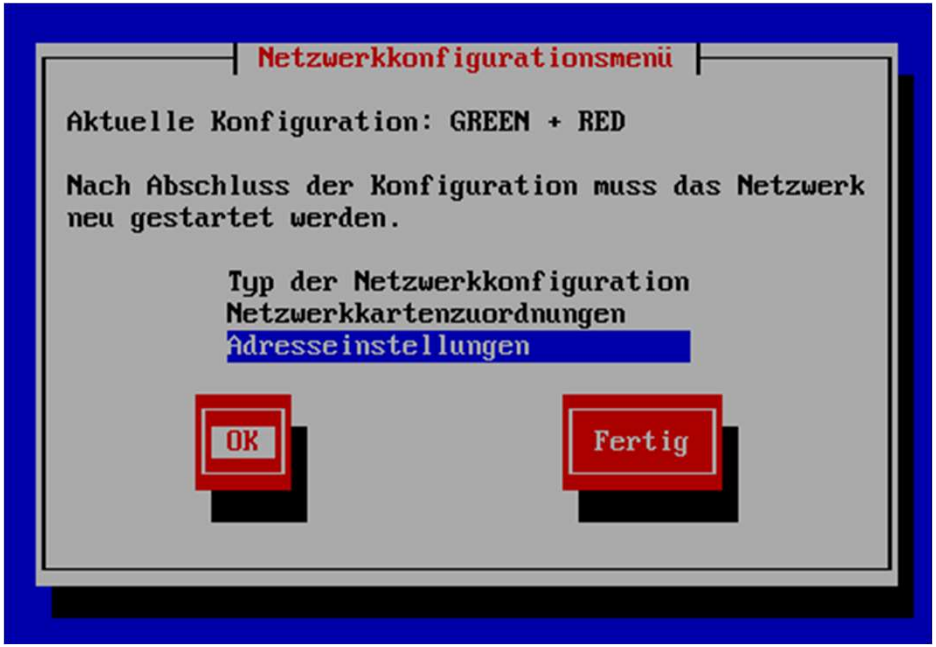
- Die verbleibende NIC wählt man nun für das Subnetz RED

VM ipfire starten und installieren

- Nach der Zuweisung erscheint wieder die Übersicht, die man dann durch die Auswahl „Fertig“ bestätigt.

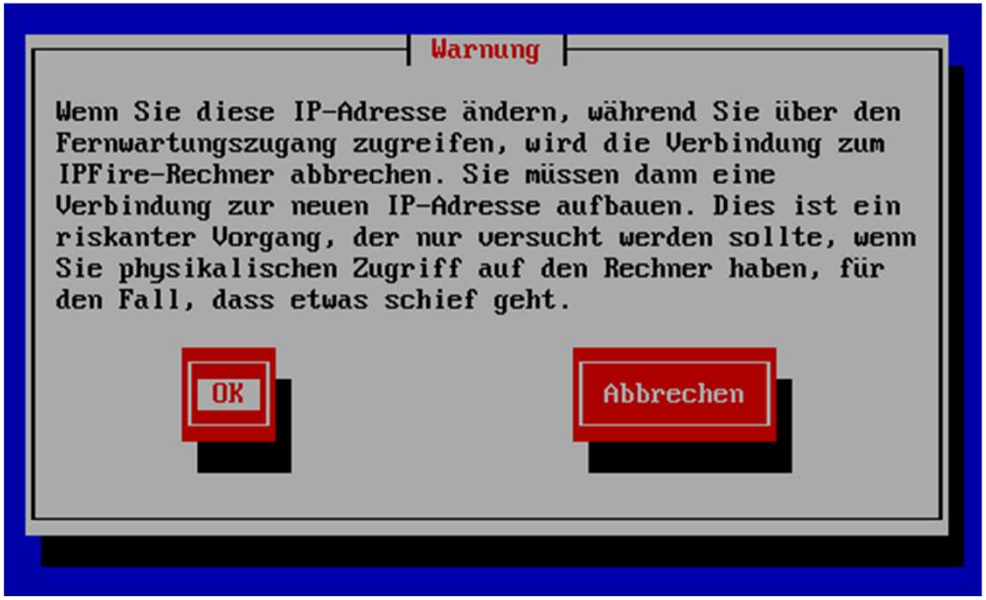
VM ipfire starten und installieren

- Jetzt erfolgt noch die „Adresseinstellung“
- Das Subnetz RED soll mittels DHCP an jedes äußere Netzwerk anpassbar sein (zuhause und am Standort)
- Das Subnetz GREEN erhält seinen eigenen Adressbereich
  - Ich habe das Subnetz 192.168.111.0 gewählt, in der Hoffnung, dass Sie weder zuhause noch im Standort den gleichen Adressbereich vorfinden



VM ipfire starten und installieren

- GREEN beginnt:
- Die Warnung können Sie ignorieren, da unser privates Netz noch keine Konfiguration enthält



VM ipfire starten und installieren

- Die Routeradresse im Subnetz GREEN stellen Sie auf 192.168.111.1 ein.
- Die Netzmaske ist für ein 8-Bit-Subnetz eingerichtet, also 255.255.255.0

VM ipfire starten und installieren

- Und jetzt das Subnetz RED:
- Hier darf kein IP-Eintrag erfolgen

VM ipfire starten und installieren

- Wechseln Sie zum Eintrag DHCP und drücken Sie die Leertaste. Damit erscheint das Sternchen vor DHCP.
- Alle anderen Einträge bleiben erhalten und Sie können mit <OK> bestätigen

VM ipfire starten und installieren

- Damit ist auch RED eingerichtet, was Sie mit <Fertig> bestätigen

VM ipfire starten und installieren

- Das Subnetz GREEN besitzt zwar die IP-Adresse des Routers, aber noch keine Adressen für die anderen Geräte (VM's), also debian, deb-1 und wenn ausgewählt, auch Ihre Windows VM.
- IPFire hat einen DHCP-Server an Bord,
  - der nur aktiviert (Leertaste) und
  - dessen IP-Bereich gewählt werden muss
  - bei mir: 192.168.111.10 bis 192.168.111.20
- Einen sekundären DNS-Eintrag können Sie optional wählen, z.B. 1.1.1.1

VM ipfire starten und installieren

- Damit ist IPFire eingerichtet und kann gestartet werden
  - Achten Sie beim Start auf die <OK>-Einträge
  - Es dauert eine längere Zeit, bis das Subnetz RED die IP-Adresse aus dem jeweiligen Netz erhält
  - Sollte RED keine IP erhalten, was mit einem Fehler quittiert wird, dann tauschen Sie, wie schon beschrieben, die Zuweisungen „privat1“ und „extern1“ der Netzwerkkarten in der Hyper-V einfach aus und starten die VM ipfire neu
  - Zum Administrieren von ipfire gibt man im Browser des debian-Clients https://ipfire:444 ein. Die Sicherheitsbedenken kann man ignorieren

Die VM‘s debian und deb-s1 starten

- Starten Sie die beiden Linux-VM (debian und deb-s1) neu
- Sie sollen dabei ihre neuen IP-Adressen aus dem Subnetz GREEN 192.168.111.0/24 erhalten und zwar aus dem zugewiesenen DHCP-Bereich 192.168.111.10 bis 192.168.111.20
- Mit Hilfe der Routingfunktion des ip-Kommandos kann man die Adressen der VM leicht abfragen (nächste Folie)
- Ihre angezeigten Adressen können natürlich anders sein, müssen aber, wie oben gezeigt, im Subnetz 192.168.111.0/24 liegen

IP-Adresse mit ip r

- Nochmal zur Erinnerung: Ihre Adressen können im 4. Oktett anders lauten, aber die ersten drei Oktette müssen auch bei Ihnen 192.168.111 sein

```
helmut@deb-s1:~$ ip r
default via 192.168.111.1 dev eth0
192.168.111.0/24 dev eth0 proto kernel scope link src 192.168.111.13
helmut@deb-s1:~$ _
```

```
helmut@debian:~$ ip r
default via 192.168.111.1 dev eth0
1.1.1.1 via 192.168.111.1 dev eth0
192.168.111.0/24 dev eth0 proto kernel scope link src 192.168.111.10
192.168.111.1 dev eth0 scope link
```

Ping mich an

- Da sich beide VM im gleichen Subnetz 192.168.111.0 befinden, muss der ping auf die jeweils andere VM gelingen
- Wählen Sie hier natürlich Ihre Adressen

```
helmut@deb-s1:~$ ping -c4 192.168.111.10
PING 192.168.111.10 (192.168.111.10) 56(84) bytes of data.
64 bytes from 192.168.111.10: icmp_seq=1 ttl=64 time=0.783 ms
64 bytes from 192.168.111.10: icmp_seq=2 ttl=64 time=0.401 ms
64 bytes from 192.168.111.10: icmp_seq=3 ttl=64 time=0.708 ms
64 bytes from 192.168.111.10: icmp_seq=4 ttl=64 time=0.501 ms

--- 192.168.111.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3080ms
rtt min/avg/max/mdev = 0.401/0.598/0.783/0.153 ms
```

```
helmut@debian:~$ ping -c4 192.168.111.13
PING 192.168.111.13 (192.168.111.13) 56(84) bytes of data.
64 bytes from 192.168.111.13: icmp_seq=1 ttl=64 time=0.555 ms
64 bytes from 192.168.111.13: icmp_seq=2 ttl=64 time=0.541 ms
64 bytes from 192.168.111.13: icmp_seq=3 ttl=64 time=0.574 ms
64 bytes from 192.168.111.13: icmp_seq=4 ttl=64 time=0.660 ms

--- 192.168.111.13 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3077ms
rtt min/avg/max/mdev = 0.541/0.582/0.660/0.046 ms
```

Troubleshooting

- Falls bei Ihnen der ping nicht zustande kommt, liegt es an den Einstellungen der folgenden Dateien, die Sie als root mit nano bearbeiten können
- Die Inhalte sind bei beiden VM identisch
  - # nano /etc/network/interfaces

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
allow-hotplug eth0
iface eth0 inet dhcp
```

nano /etc/resolv.conf

```
/etc/resolv.conf
domain linux.local
search linux.local
nameserver 192.168.111.1
nameserver 1.1.1.1
```

Falls Sie Korrekturen ausführen mussten, starten Sie die VM mit reboot (als root) neu

Statische IP für deb-s1 zuweisen

- Server, wie deb-s1, arbeiten immer mit einer statischen IP-Adresse.
  - Die VM deb-s1 erhält die statische IP-Adresse 192.168.111.2
  - Die Adresse liegt außerhalb des DHCP-Bereichs, so dass dort eine Adresse mehr nutzbar ist
  - Passen Sie die Datei /etc/network/interfaces als root wie folgt an:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
allow-hotplug eth0
iface eth0 inet static
        address 192.168.111.2/24
        network 192.168.111.0
        gateway 192.168.111.1
        broadcast 192.168.111.255
        dns-search linux.local
        dns-nameservers 192.168.111.1 1.1.1.1
```

Statische IP auf deb-s1 kontrollieren

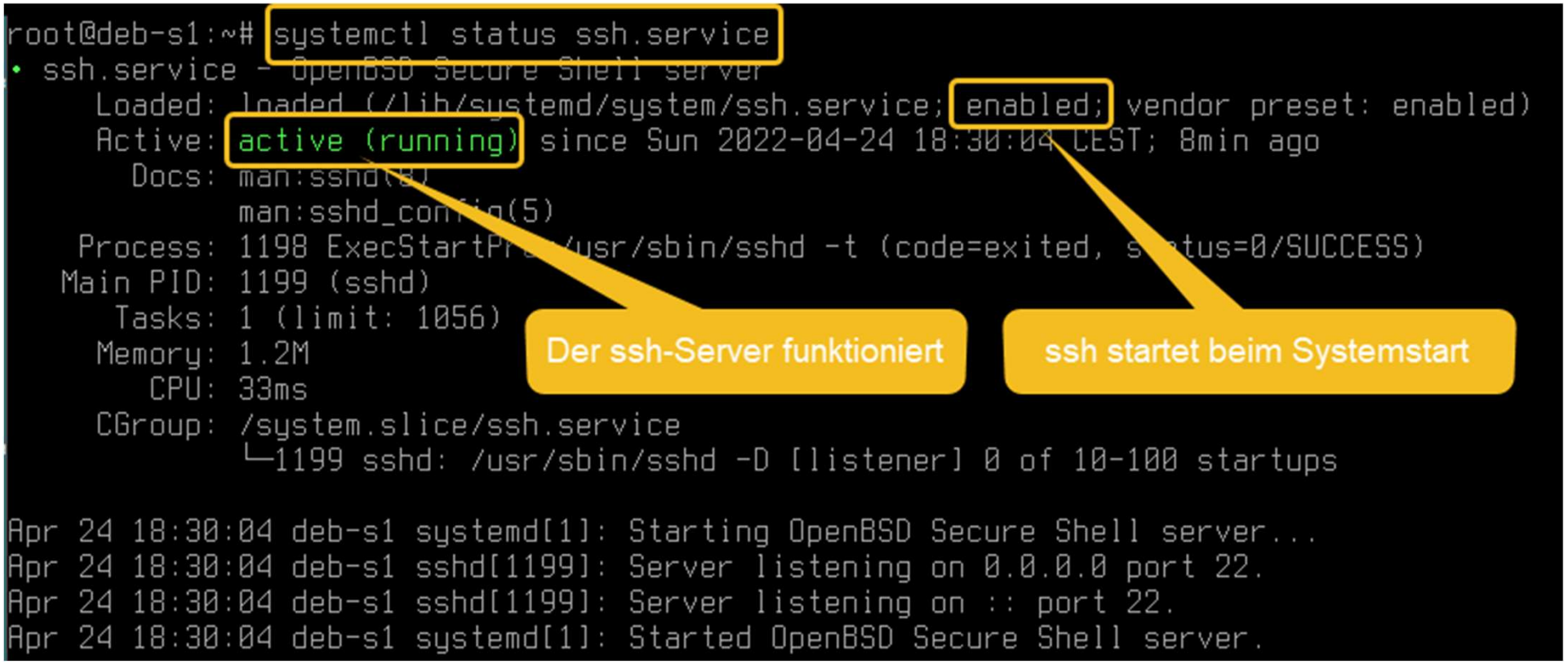
- Nach dem Speichern der neuen Einstellungen ist es am sichersten den deb-s1 neu zu starten
  - Geben Sie dazu als root das Kommando „reboot“ ein
  - Danach melden Sie sich wieder normal an und kontrollieren mit ip a den Erfolg der statischen IP

```
helmut@deb-s1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:9f:51:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.111.2/24 brd 192.168.111.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe9f:5100/64 scope link
       valid_lft forever preferred_lft forever
```

Funktioniert der SSH-Server?

- Der ssh-Server-Dienst sollte weiterhin in Betrieb sein
- Durch die statische IP ist der Server jetzt immer ansprechbar
- Bevor sich ein Client, wie unsere VM „debian“ mit dem Server über ssh verbindet, sollte man kontrollieren, ob der Serverdienst auch umfänglich funktioniert
- Seit der Einführung von systemd stehen einige neue Kommandos zur Verfügung, die Service-Dienste starten, stoppen und restarten können. Aber auch die Statusabfrage ist komfortabel möglich
- Diese geschieht für den ssh-Server als root folgendermaßen:
- # systemctl status sshd.service (.service kann weggelassen werden)

Statusabfrage von sshd auf deb-s1



Der SSH-Client

- In unserem System sollen die VM „debian“ (mit GUI) und unsere Windows-VM als Clients fungieren
- Wechseln Sie zur VM „debian“ und führen Sie die folgenden Schritte aus:
- Öffnen Sie Ihr Terminal (LX-Terminal).
- Die Einrichtung erfolgt aus Sicherheitsgründen auf Benutzerebene, also nicht als root!!!!!
- Urspüngliche allgemeine Form der Verbindung zum SSH-Server
- `$ ssh –l <Benutzer> <IP-Adresse des SSH-Servers>`
- Bei mir: `$ ssh –l helmut 192.168.111.2`

Verbindung herstellen

```
helmut@debian:~$ ssh -l helmut 192.168.111.2
The authenticity of host '192.168.111.2 (192.168.111.2)' can't be established.
ECDSA key fingerprint is SHA256:zUSuUsXPEpGLSrnyVtt3Bs/jWpBigAyV+2gmdxWyeuo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.111.2' (ECDSA) to the list of known hosts.
helmut@192.168.111.2's password: Ihr Passwort für den Server deb-s1 eingeben
Linux deb-s1 5.10.0-13-amd64 #1 SMP Debian 5.10.106-1 (2022-03-17) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Apr 24 18:21:17 2022
helmut@deb-s1:~$ Ihr Client "debian" ist mit dem Server "deb-s1" verbunden
```

Verbindung ist hergestellt

- Bis zur Abmeldung mit dem Kommando „exit“, werden jetzt alle Anweisungen auf dem Server ausgeführt
- Danach kann die Verbindung jederzeit wieder hergestellt werden
- Als Verbindungsmethode funktioniert auch: `$ ssh <Benutzer>@<IP des Servers>`
- Bei mir: `$ ssh helmut@192.168.111.2`
- Dabei wird nicht mehr nach dem Fingerprint-Schlüssel gefragt, weil dieser bereits hinterlegt ist und für die neuerliche Verbindung gewählt wird
- Die Verbindung über das Passwort ist gut, aber es geht noch besser. Dazu später mehr

Verbindung anzeigen

- Durch die Eingabe des Kommandos w bzw. who lassen sich die derzeitigen Verbindungen auf dem Server anzeigen

```
helmut@deb-s1:~$ w
 19:07:45 up 46 min,  2 users,  load average: 0,00, 0,00, 0,00
USER     TTY      VON              LOGIN@   IDLE   JCPU   PCPU WHAT
helmut   tty1     -                18:21   29:04   0.10 s  0.01 s -bash
helmut   pts/0    192.168.111.10   18:53    0.00 s  0.00 s  0.00 s w
helmut@deb-s1:~$ who
helmut   tty1         2022-04-24 18:21
helmut   pts/0        2022-04-24 18:53 (192.168.111.10)
helmut@deb-s1:~$
```

- Lassen Sie die Verbindung bestehen. Jetzt folgt noch Windows
- Geben Sie in der Kommandozeile ipconfig /all ein

Windows als Client im Subnetz GREEN

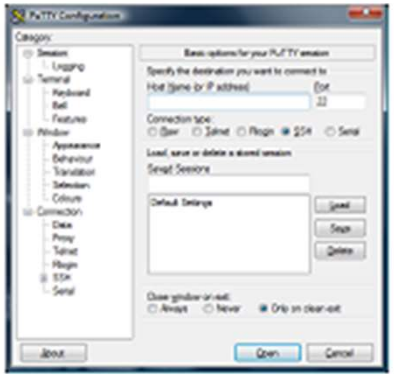
```
   Hostname  . . . . . . . . . . . . : DESKTOP-0TAKUG0
   Primäres DNS-Suffix . . . . . . . :
   Knotentyp . . . . . . . . . . . . : Hybrid
   IP-Routing aktiviert  . . . . . . : Nein
   WINS-Proxy aktiviert  . . . . . . : Nein
   DNS-Suffixsuchliste . . . . . . . : linux.local

Ethernet-Adapter Ethernet:

   Verbindungsspezifisches DNS-Suffix: linux.local
   Beschreibung. . . . . . . . . . . : Microsoft Hyper-V Network Adapter
   Physische Adresse . . . . . . . . : 00-15-5D-9F-51-05
   DHCP aktiviert. . . . . . . . . . : Ja
   Autokonfiguration aktiviert . . . : Ja
   Verbindungslokale IPv6-Adresse  . : fe80::345d:8242:f59a:18c7%6(Bevorzugt)
   IPv4-Adresse  . . . . . . . . . . : 192.168.111.14(Bevorzugt)
   Subnetzmaske  . . . . . . . . . . : 255.255.255.0
   Lease erhalten. . . . . . . . . . : Sonntag, 24. April 2022 19:23:01
   Lease läuft ab. . . . . . . . . . : Sonntag, 24. April 2022 20:23:02
   Standardgateway . . . . . . . . . : 192.168.111.1
   DHCP-Server . . . . . . . . . . . : 192.168.111.1
   DHCPv6-IAID . . . . . . . . . . . : 100668765
   DHCPv6-Client-DUID. . . . . . . . : 00-01-00-01-29-E6-D3-40-00-15-5D-9F-51-05
   DNS-Server  . . . . . . . . . . . : 192.168.111.1
                                       1.1.1.1
   NetBIOS über TCP/IP . . . . . . . : Aktiviert
```

Windows als Client

- Um Ihre Windows-VM zum Administrieren von deb-s1 zu nutzen, müssen Sie dessen NIC in der Hyper-V ins private Netz bringen (privat1)
- In Windows kann man die Powershell nutzen, trotzdem ist es sinnvoll z. B. das kostenlose Programm PuTTY von der Site putty.org zu installieren

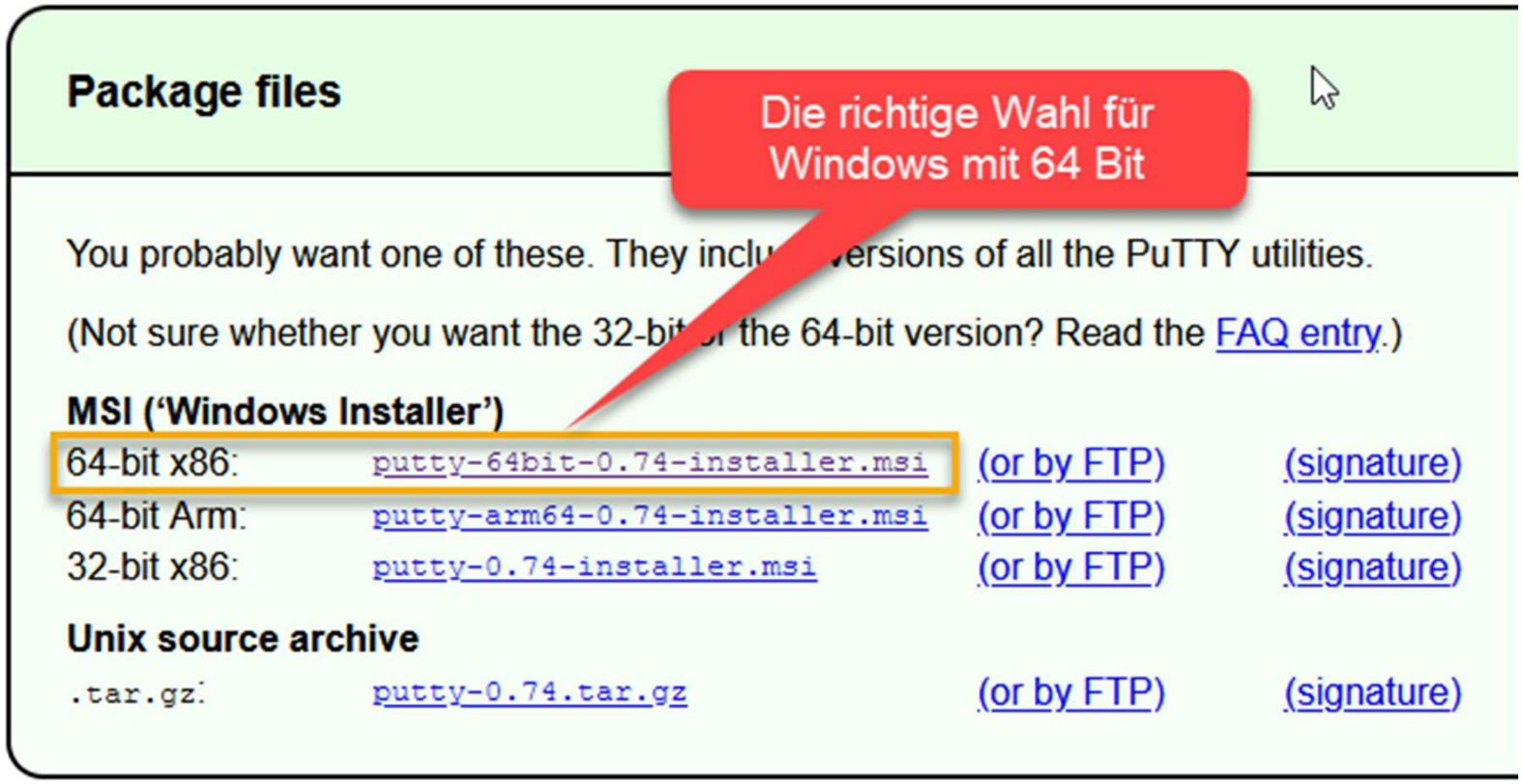


Download PuTTY

PuTTY is an SSH and telnet client, de
software that is available with source c

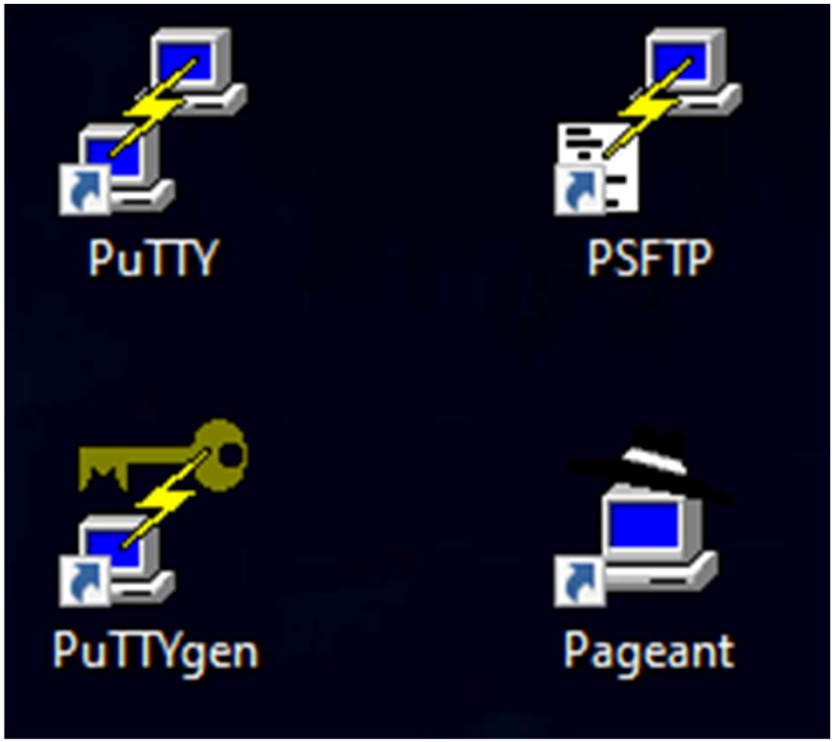
You can download PuTTY here.

PuTTY-Auswahl



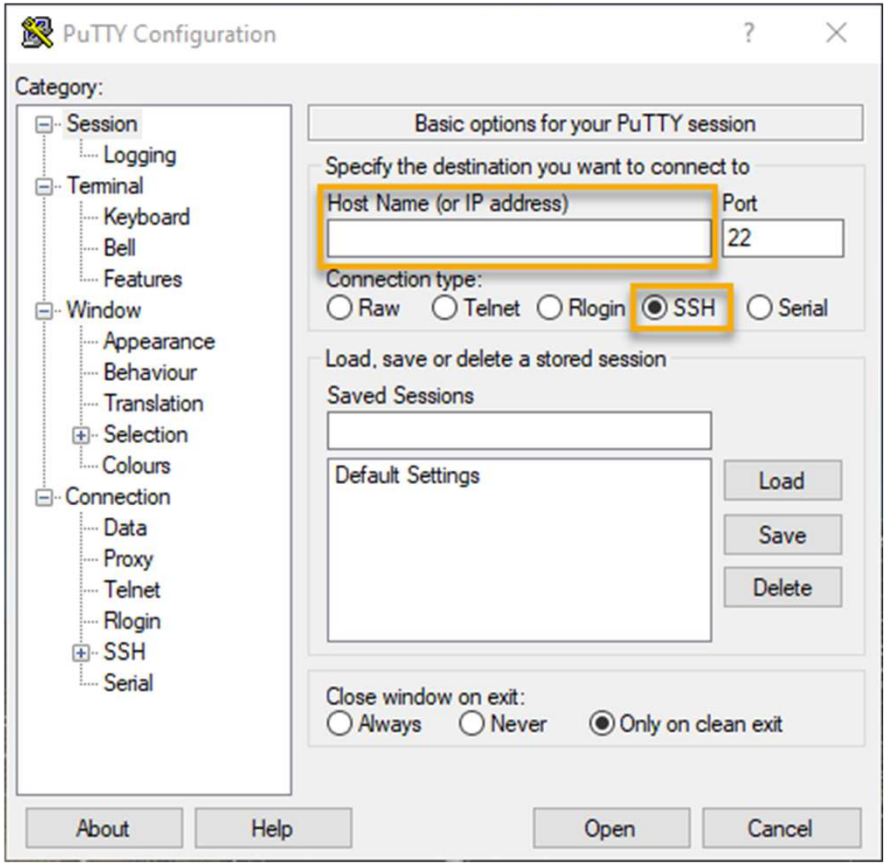
Die PuTTY-Collection

- Mit der putty.msi-Datei werden insgesamt 4 Anwendungen installiert
  - Der SSH-Client PuTTY
  - Der Key-Generator PuTTYgen
  - Der Secure-FTP-Client PSFTP
  - Der Private-Key-Verwalter Pageant
- Zunächst werden wir nur PuTTY nutzen um eine SSH-Verbindung zu deb-s1 aufzubauen
- Dazu starten Sie PuTTY (deb-s1 muss natürlich aktiv sein)

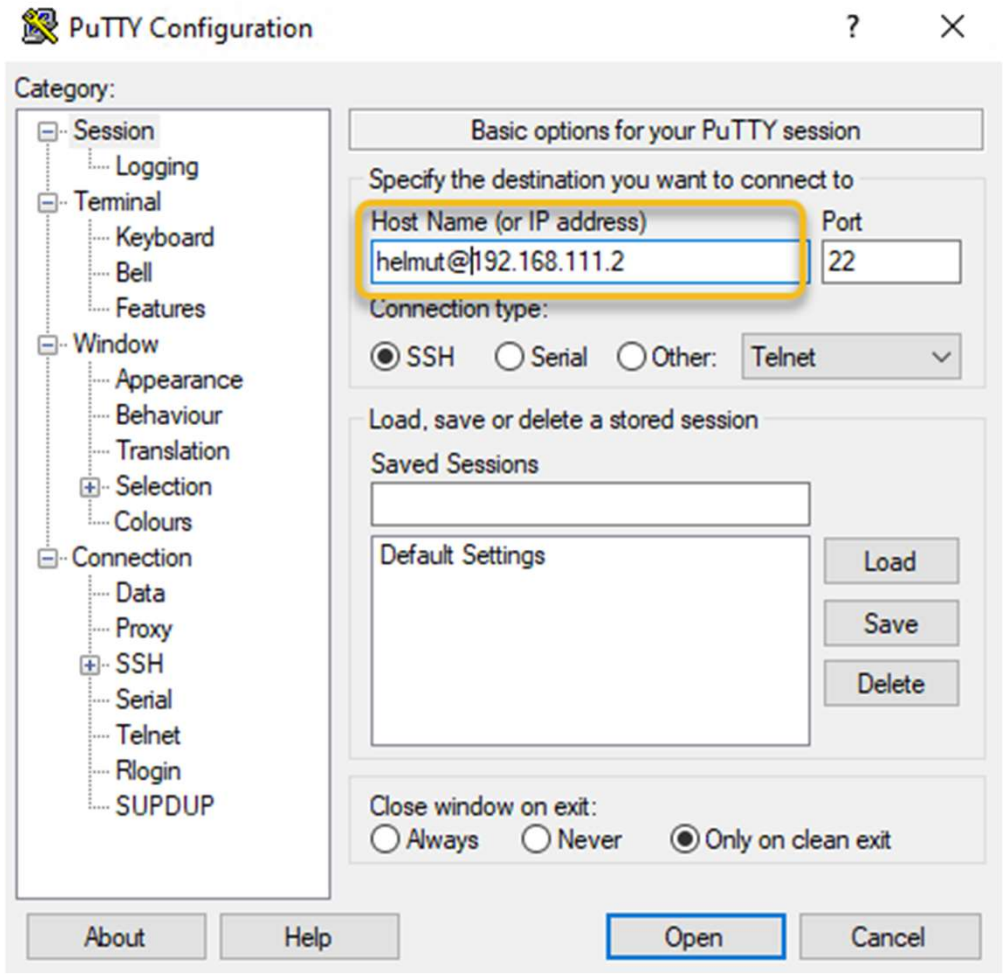


Das PuTTY-Fenster

- Nach dem Start von PuTTY zeigt sich dieses Fenster.
- Die bekannten Parameter sind ausgewählt:
  - SSH und der Standard-Port 22 den SSH benutzt
- Man muss nur noch den Hostname bzw. die IP-Adresse von deb-s1 eingeben
- Dabei kann der Benutzername vorangestellt werden
  - <Benutzer>@<Hostname bzw. IP-Adresse> bei mir
  - helmut@192.168.111.2

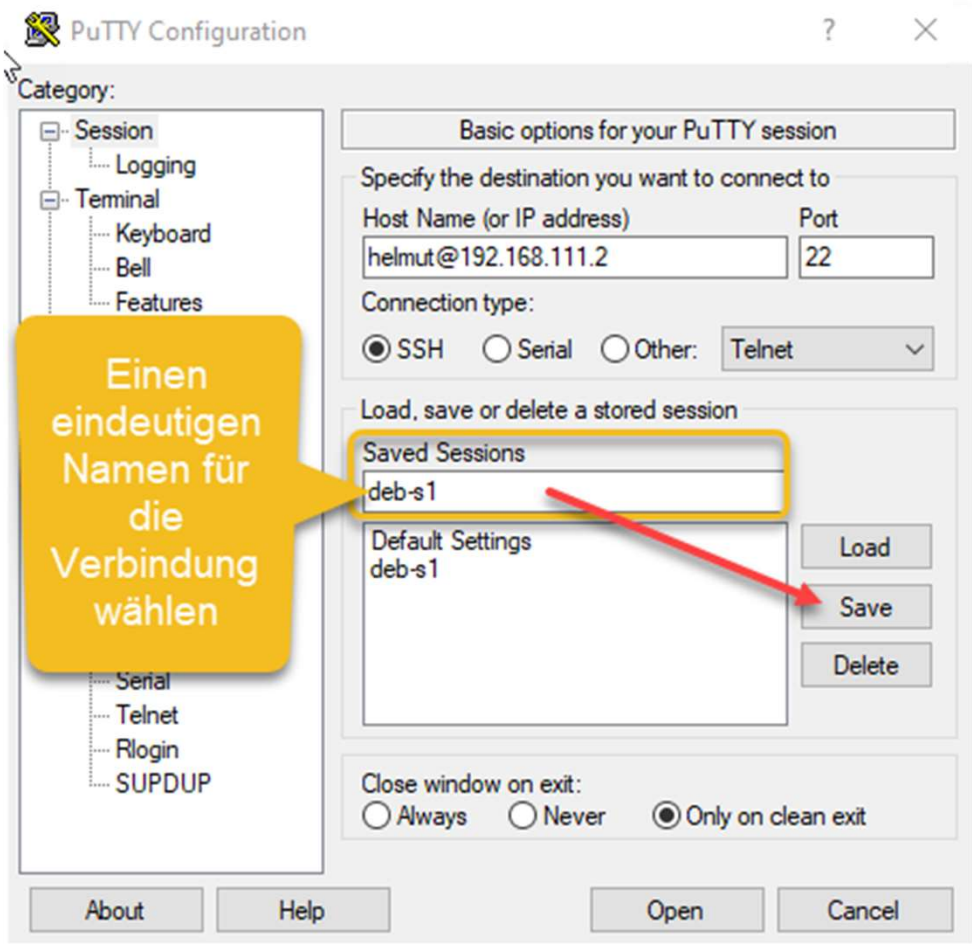


Das PuTTY-Fenster mit Einwahl

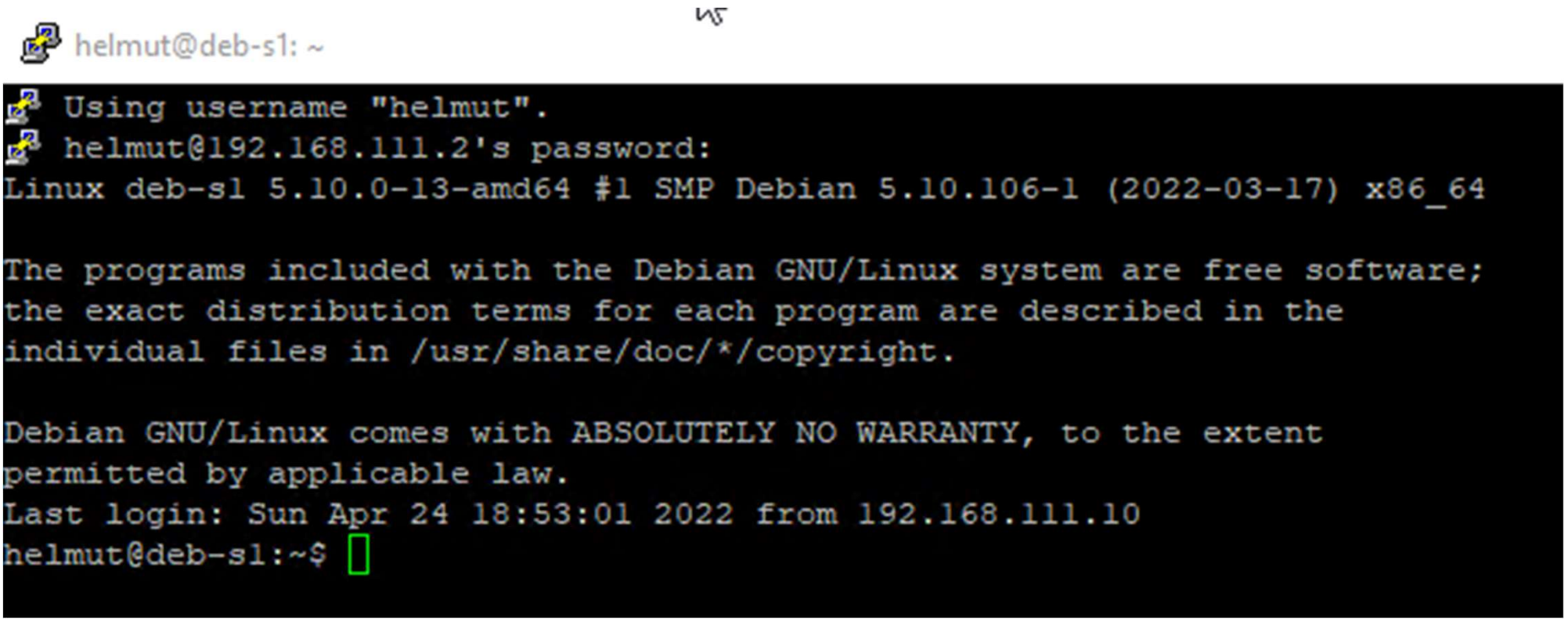


Die Verbindung in PuTTY speichern

- Die Adresseingaben müssen nicht ständig wiederholt werden, wenn man die Verbindungsdaten speichert
- Durch Klick auf Open wird die Verbindung aufgebaut
- Das Sicherheitsfenster wird nur bei der ersten Einwahl angezeigt



Die PuTTY-Konsole



2 Clients haben Zugriff auf deb-s1

- Sowohl der Linux-Client (debian) als auch der Windows-Client haben jetzt eine Verbindung zu deb-s1 und können jetzt beide darin arbeiten
- Das Kommando who zeigt es wiederum an:
  - pts: Pseudo-Terminal bei ssh tty: Terminal (teletype) Direkteinwahl auf dem Server
- Alternativ kann auch das Kommando w eingesetzt werden

```
helmut@deb-s1:~$ who
helmut   tty1         2022-04-24 20:11
helmut   pts/0        2022-04-24 19:48 (192.168.111.14)
helmut   pts/1        2022-04-24 20:10 (192.168.111.10)
helmut@deb-s1:~$
```