

Proxmox - Tools / BUG Fixes Workarounds / Tipps und Tricks

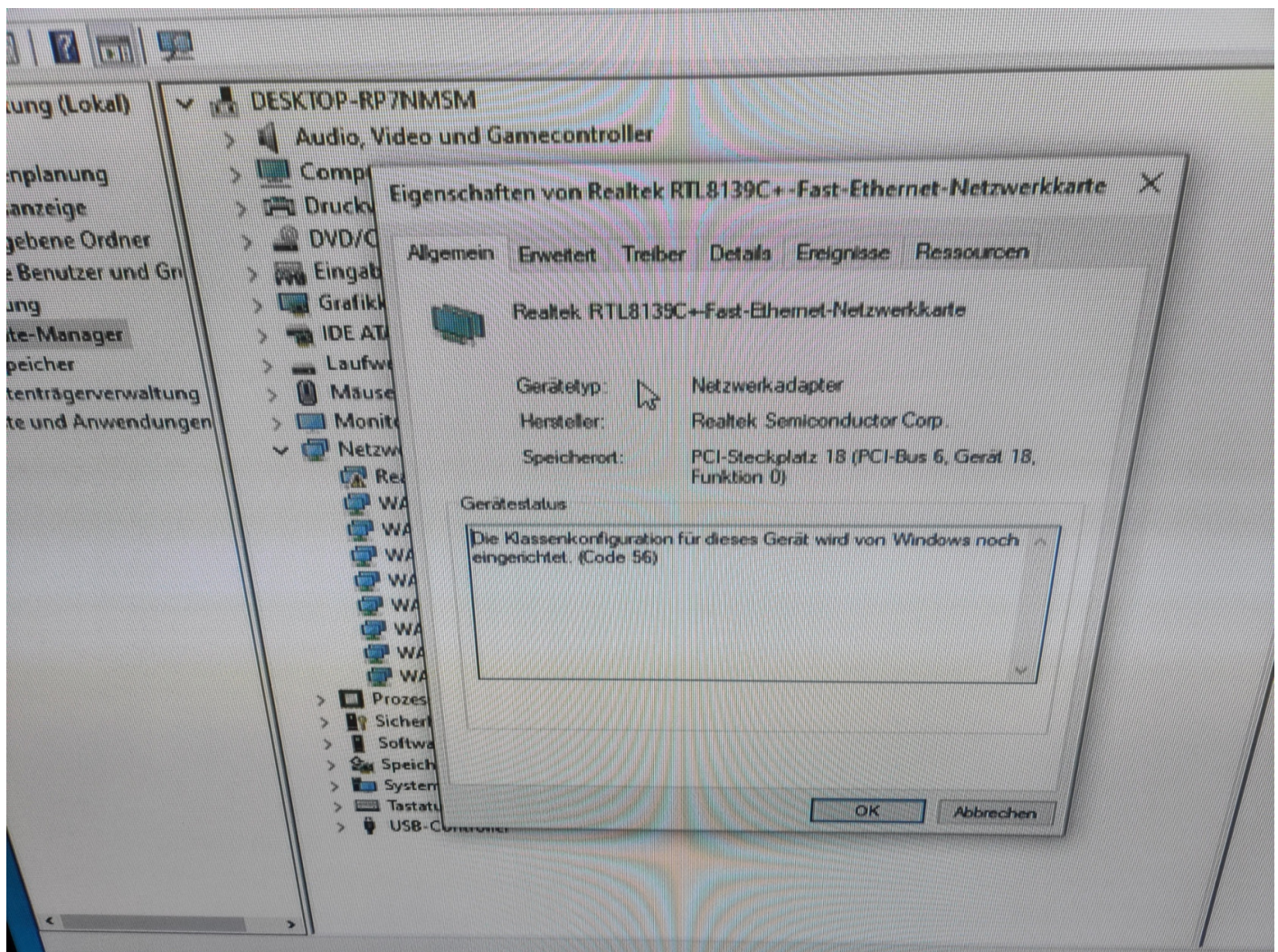
Hier kommen alle Themen Rund um Fehlerbehebungen und Workarounds rein

- Proxmox 7 - Deutsche Windows ISO nach installation keine Netzwerkkarten nur Ausrufungszeichen
- Proxmox/Debian bei älteren Monitoren - Out of Range
- Proxmox ZFS Offline Mount mit PVE Install DVD/STICK
- LXC Conatiner startet nach Installation einer Software die als Dienst startet nicht mehr. (AppArmor)
- Proxmox Wartungsmodus / Maintenance Mode - Diasable / Enable VMS at Start
- Proxmox Host aus VM herunterfahren
- Proxmox - Es lassen sich keine Snapshots anlegen
- Proxmox 7 installation hängt bei 99% make system bootable
- Anlegen / Clonen einer ARM64 VM Fehler
- LXC Conatiner - Routen beim start setzten,da post in interfaces nicht geht
- Ceph Upgrade von Pacific 17.2.7 auf reef 18.2.2 osd starten nicht
- Cpeh crash log löschen
- HDD Tray ermitteln LED blinken und/oder austauschen

Proxmox 7 - Deutsche Windows ISO nach installation keine Netzwerkkarten nur Ausrufungszeichen

Bei einem Proxmox update auf Version 7 können keine neuen Netzwerkkarten hinzugefügt werden und wenn doch Fehler 53
klassenkonfiguration wird von windows noch eingerichtet

Siehe Schreenshot.



Abhilfe q35 auf version 5.1 umstellen.

	Add	Remove	Edit	Disk Action	Revert
Summary					
> Console					
Hardware					
Cloud-Init					
Options					
Task History					
Monitor					
Backup					
Replication					
Snapshots					
Firewall					
Memory					
Processors					
BIOS					
Display					
Machine					
SCSI Controller					
CD/DVD Drive (ide0)					
CD/DVD Drive (ide2)					
Hard Disk (scsi0)					
Network Device (net0)					
EFI Disk					

Beim neustart / starten könnte es einen Bluescreen geben.

einfach nochmals starten, läuft dann.

Fertig

Proxmox/Debian bei älteren Monitoren - Out of Range

Hat man am Server einen Monitor aus den 2000er angeschlossen und der unterstützt die aktuellen modi nicht mehr.

Per SSH einloggen oder gleich nach der installation im grub nomodeset stezten.

Für GRUP legacy boot oder EFI ohne ZFS

```
nano /etc/default/grub
```

FÜR EFI Boot mit ZFS

```
nano /etc/kernel/cmdline
```

hinter quiet einfach

```
nomodeset
```

```
schreiben
```

Beispiel

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet nomodeset"
```

solltet ihr noch mehr custom paramter haben stehen die natürlich dahinter

zum Schluss für legacy oder nur efi

```
update-grub
```

für EFI mit ZFS

```
pve-efiboot-tool refresh
```

Dann den Server neustarten fertig.

Nun haben wir auch endlich Bild mit nem alten Monitor

Proxmox ZFS Offline Mount mit PVE Install DVD/STICK

Beschreibung

Wie oft hat man das Problem zum Beispiel einen Proxmoxhost wo die Grafikkarte durchgereicht ist und so die Bedienung mit Tastatur und Maus nicht möglich ist. Es war immer nur möglich den Host per ssh und Weboberfläche zu steuern.

Jetzt kommt ein ein neues Mainboard oder neue Grafikkarte rein und schon ändert sich der Name der Netzwerkkarte.

Und nun? Mit der PVE DVD Starten das ZFS Mouneten die Netzwerkkonfig ändern undfertig.

DVD Starten

Von der PVE Install DVD/Stick starten und den Debug Mode auswählen.

Proxmox VE 6.4 (iso release 1) - <https://www.proxmox.com/>



Welcome to Proxmox Virtual Environment

```
Install Proxmox VE
Install Proxmox VE (Debug mode)
Rescue Boot
Test memory (Legacy BIOS)
```

Nun sind wir in einem Debug Terminal.

in diesem können wir aber noch nicht wirkliche Befehle absetzen.

Deswegen hier mit STRG+D drücken um in das zweite Terminal zu gelangen.

```
Welcome to the Proxmox VE 6.4 installer
initial setup startup
mounting proc filesystem
mounting sys filesystem
comandline: BOOT_IMAGE=/boot/linux26 ro ramdisk_size=16777216 rw quiet splash=verbose proxdebug
loading drivers: i2c_piix4 pata_acpi floppy qemu_fw_cfg mac_hid uhci_hcd virtio_scsi input_leds serio_raw psmouse pcspkr
searching for block device containing the ISO proxmox-ve-6.4-1
with ISO ID '0156e1be-a74b-11eb-af5b-3f8908c7cde1'
testing device '/dev/sr0' for ISO
found Proxmox VE ISO
Debugging mode (type 'exit' or press CTRL-D to continue startup)
/ #
```

Hier können wir nun Befehle absetzen.

```
Welcome to the Proxmox VE 6.4 installer
initial setup startup
mounting proc filesystem
mounting sys filesystem
comandline: BOOT_IMAGE=/boot/linux26 ro ramdisk_size=16777216 rw quiet splash=ve
loading drivers: i2c_piix4 pata_acpi floppy qemu_fw_cfg mac_hid uhci_hcd virtio
searching for block device containing the ISO proxmox-ve-6.4-1
with ISO ID '0156e1be-a74b-11eb-af5b-3f8908c7cde1'
testing device '/dev/sr0' for ISO
found Proxmox VE ISO
Debugging mode (type 'exit' or press CTRL-D to continue startup)
/ # Starting Proxmox installation
Installing additional hardware drivers
[ ok ] Starting hotplug events dispatcher: systemd-udevd.
[ ok ] Synthesizing the initial hotplug events (subsystems)...done.
[ ok ] Synthesizing the initial hotplug events (devices)...done.
[ ok ] Waiting for /dev to be fully populated...done.
mount: devpts mounted on /dev/pts.
Dropping in debug shell inside chroot before starting installation
type exit or CTRL-D to start installation wizard
root@(none):/#
```

Netzwerkkarte ermitteln und Tastatursprache festlegen

Als erstes wollen wir wissen wie unsere Netzwerkkarte heißt, das hat zwar nichts mit ZFS zu tun, allerdings soll es hier ja auch einen Sinn machen.

```
ip a
```

In unserem Beispiel ens18

```
root@(none):/# ip a
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens18: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether f2:e7:c7:f1:09:e4 brd ff:ff:ff:ff:ff:ff
```

Die Tastatursprache legen wir mit

```
dpkg-reconfigure keyboard-configuration
```

fest.

Nun werden wir nach dem Tastaturmodell gefragt.

```
debconf: falling back to frontend: Readline
Configuring keyboard-configuration
-----
Please select the model of the keyboard of this machine.

 1. A4Tech KB-21
 2. A4Tech KBS-8
 3. A4Tech Wireless Desktop RFKB-23
 4. Acer AirKey V
 5. Acer C300
 6. Acer Ferrari 4000
 7. Acer laptop
 8. Advance Scorpius KI
 9. Amiga
10. Apple
11. Apple Aluminium (ANSI)
12. Apple Aluminium (ISO)
13. Apple Aluminium (JIS)
14. Apple laptop
15. Asus laptop
16. Atari TT
17. Azona RF2300 wireless Internet
18. BTC 5090
19. BTC 5113RF Multimedia
20. BTC 5126T
21. BTC 6301URF
22. BTC 9000
23. BTC 9000A
24. BTC 9001AH
25. BTC 9019U
26. BTC 9116U Mini Wireless Internet and Gaming
27. BenQ X-Touch
28. BenQ X-Touch 730
29. BenQ X-Touch 800
30. Brother Internet
31. Cherry B.UNLIMITED
32. Cherry Blue Line CyBo@rd
33. Cherry Blue Line CyBo@rd (alt.)
34. Cherry CyBo@rd USB-Hub
35. Cherry CyMotion Expert
36. Cherry CyMotion Master Linux
37. Cherry CyMotion Master XPress
38. Chicony Internet
39. Chicony KB-9885
40. Chicony KU-0108
41. Chicony KU-0420

[More]
```

Mit Enter gehts auf die nächste Seite. Ich nehme die 71 Generic 105 intl..
Solange enter drücken bis Keyboardmodell dort steht, 71 eintippen enter drücken

```
180. Truly Ergonomic Computer Key
181. Trust Direct Access
182. Trust Slimline
183. Trust Wireless Classic
184. TypeMatrix EZ-Reach 2020
185. TypeMatrix EZ-Reach 2030 PS2
186. TypeMatrix EZ-Reach 2030 USB
187. TypeMatrix EZ-Reach 2030 USB
188. TypeMatrix EZ-Reach 2030 USB
189. Unitek KB-1925
190. ViewSonic KU-306 Internet
191. Winbook Model XP5
192. Yahoo! Internet
193. eMachines m6800 laptop
```

Keyboard model: 71

Nun das Layout, hier die 11 keine toten Tasten

```
Please select the layout matching the keyboard for this machine.
```

- | | |
|--|---|
| 1. German | 11. German - German (no dead keys) |
| 2. German - German (Dvorak) | 12. German - German (with Sun dead keys) |
| 3. German - German (Macintosh) | 13. German - Lower Sorbian |
| 4. German - German (Macintosh, no dead keys) | 14. German - Lower Sorbian (QWERTZ) |
| 5. German - German (Neo 2) | 15. German - Romanian (Germany) |
| 6. German - German (QWERTY) | 16. German - Romanian (Germany, no dead keys) |
| 7. German - German (T3) | 17. German - Russian (Germany, phonetic) |
| 8. German - German (dead acute) | 18. German - Turkish (Germany) |
| 9. German - German (dead grave acute) | 19. Other |
| 10. German - German (dead tilde) | |

Keyboard layout: 11_

Nun ALT-GR Funktionstaste hier standard 1 wählen

```
With some keyboard layouts, AltGr is a modifier key used to input some characters, primarily ones that are unusual for the language of the keyboard layout, such as foreign currency symbols and accented letters. These are often printed as an extra symbol on keys.
```

- | | | | |
|--|-------------------|---------------------|--------------------|
| 1. The default for the keyboard layout | 4. Right Control | 7. Left Alt | 10. Both Logo keys |
| 2. No AltGr key | 5. Right Logo key | 8. Left Logo key | 11. Both Alt keys |
| 3. Right Alt (AltGr) | 6. Menu key | 9. Keypad Enter key | |

Key to function as AltGr: 1

Nun den Compose Key wählen (Ich habe keinen) da eure Einstellung wählen.

Also bei mir die 1

```
The Compose key (known also as Multi_key) causes the computer to interpret the next few keystrokes as a combination in order to produce a character not found on the keyboard.
```

```
On the text console the Compose key does not work in Unicode mode. If not in Unicode mode, regardless of what you choose here, you can always also use the Control+period combination as a Compose key.
```

- | | | | | | | |
|-------------------|----------------------|------------------|-------------------|-------------|------------------|--------------|
| 1. No compose key | 2. Right Alt (AltGr) | 3. Right Control | 4. Right Logo key | 5. Menu key | 6. Left Logo key | 7. Caps Lock |
|-------------------|----------------------|------------------|-------------------|-------------|------------------|--------------|

Compose key: 1_

Da wir keinen X-Server haben hier nein (no) wählen

```
By default the combination Control+Alt+Backspace does nothing. If you want it can be used to terminate the X server.  
Use Control+Alt+Backspace to terminate the X server? [yes/no] no
```

Vielleicht klappts bei euch bir mir nicht.

Legende für English

```
\ = #  
- = ß  
/ = -  
? = shift + -  
z = y  
y = z  
* = shift + 8  
@ = shift + 2  
# = shift + 3  

```

ZFS einbinden

Pools auflisten.

Mit diesem Befehl werden alle pools aufgelistet ohne diese zu importieren. Vllleicht möchte man ja nur einen bestimmten pool importieren. Wir haben hier nur einen. nämlich rpool.

```
zpool import -a
```

Hier bekommen wir nämlich einen Hinweis, das der Pool schon mal woanders gemountet war. Diese machen wir uns zu nutze um die Pools aufzulisten.

```
root@(none):/# zpool import -a  
cannot import 'rpool': pool was previously in use from another system.  
Last accessed by (none) (hostid=dc47353) at Tue Jan 3 07:09:32 2023  
The pool can be imported, use 'zpool import -f' to import the pool.  
root@(none):/# _
```

Möchte man explicit einen pool importieren gibt man dessen namen an und nicht -a

```
zpool import rpool
```

Die selbe Meldung halt nur für den Pool

```
root@none):/# zpool import rpool
cannot import 'rpool': pool was previously in use from another system.
Last accessed by (none) (hostid=dc47353) at Tue Jan  3 07:09:32 2023
The pool can be imported, use 'zpool import -f' to import the pool.
root@none):/#
```

Da wir diesen aber importieren wollfen den Paramter -f hinter her.

```
zpool import rpool -f
```

Nun schauen wir uns den Status an

```
zpool status
```

Dem Pool gehts gut

```
root@none):/# zpool status
pool: rpool
state: ONLINE
config:

    NAME                                STATE      READ  WRITE CKSUM
    rpool                                ONLINE    0     0     0
      mirror-0
        scsi-0QEMU_QEMU_HARDDISK_drive-scsi0-part3  ONLINE    0     0     0
        scsi-0QEMU_QEMU_HARDDISK_drive-scsi1-part3  ONLINE    0     0     0
errors: No known data errors
root@none):/#
```

Mit dem Befehl

```
mount
```

sehen wir, wo er den Pool gemountet hat. Diese mount points unbedingt merken.

```
root@none):/# mount
none on / type overlay (rw,relatime,lowerdir=/mnt/.pve-installer:/mnt/.pve-base,upperdir=/mnt/.workdir/upper,workdir=/mnt/.workdir/work)
/dev/sr0 on /cdrom type iso9660 (ro,relatime,nojoliet,check=s,map=n,blocksize=2048)
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
tmpfs on /run type tmpfs (rw,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,relatime,size=4053052k,nr_inodes=1013263,mode=755)
tmpfs on /dev/shm type tmpfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,gid=5,mode=620,ptmxmode=000)
/rpool/ROOT/pve-1 on / type zfs (rw,noatime,xattr,noacl)
/rpool on /rpool type zfs (rw,noatime,xattr,noacl)
/rpool/ROOT on /rpool/ROOT type zfs (rw,noatime,xattr,noacl)
/rpool/data on /rpool/data type zfs (rw,noatime,xattr,noacl)
root@none):/#
```

Das gleiche geht auch mit

```
zfs list
```

```
root@(none):/# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
rpool               1.03G 60.0G   104K   /rpool
rpool/ROOT          1.03G 60.0G    96K   /rpool/ROOT
rpool/ROOT/pve-1    1.03G 60.0G   1.03G  /
rpool/data           96K   60.0G    96K   /rpool/data
root@(none):/# _
```

Wie wir sehen ist der PVE-1 eigentlich auf / gemountet, was natürlich nicht geht, da ja schon / vom system vergeben ist.

Also müssen wir den Mountpoint umbiegen

```
zfs set mountpoint=/mnt rpool/ROOT/pve-1
```

```
root@(none):/# zfs set mountpoint=/mnt rpool/ROOT/pve-1
root@(none):/#
```

Nun wurde das Verzeichnis geremounted. Siehe mittels

mount

```
root@(none):/mnt# mount
none on / type overlay (rw,relatime,lowerdir=/mnt/.pve-installer:/mnt/.pve-base,upperdir=/mnt/.workdir/upper,workdir/work)
/dev/sr0 on /cdrom type iso9660 (ro,relatime,nojoliet,check=s,map=n,blocksize=2048)
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
tmpfs on /run type tmpfs (rw,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,relatime,size=4053052k,nr_inodes=1013263,mode=755)
tmpfs on /dev/shm type tmpfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,gid=5,mode=620,ptmxmode=000)
/mnt/rpool on /rpool type zfs (rw,noatime,xattr,noacl)
rpool/ROOT on /rpool/ROOT type zfs (rw,noatime,xattr,noacl)
rpool/data on /rpool/data type zfs (rw,noatime,xattr,noacl)
rpool/ROOT/pve-1 on /mnt type zfs (rw,noatime,xattr,noacl)
root@(none):/mnt# _
```

Nun können wir zum Beispiel die Netzwerkkarte editieren.

Das root Verzeichnis vom ZFS Pool wurde in /mnt gemountet

```
nano /mnt/etc/network/interfaces
```

Tadaaa unsere Netzwerk konfig. Diese oder noch andere sachen ändern

```
GNU nano 3.2 /mnt/etc/network/in
auto lo
iface lo inet loopback

iface ens18 inet manual

auto vmbr0
iface vmbr0 inet static
    address 192.168.178.45/24
    gateway 192.168.178.1
    bridge_ports ens18
    bridge_stp off
    bridge_fd 0
```

und zum Schluss den Mountpoint wieder zurücksetzen auf /

Dazu auf / wechseln, denn das Verzeichnis darf nicht offen sein

```
cd /
```

Nun zurücksetzen

```
zfs set mountpoint=/ rpool/ROOT/pve-1
```

Mit

```
mount
```

nochmals überprüfen.

```
root@(none):/# mount
none on / type overlay (rw,relatime,lowerdir=/mnt/.pve-installer:/mnt/.pve-base,upperdir=/mnt/ir/work)
/dev/sr0 on /cdrom type iso9660 (ro,relatime,nojoliet,check=s,map=n,blocksize=2048)
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
tmpfs on /run type tmpfs (rw,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,relatime,size=4053052k,nr_inodes=1013263,mode=755)
tmpfs on /dev/shm type tmpfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,gid=5,mode=620,ptmxmode=000)
/rpool on /rpool type zfs (rw,noatime,xattr,noacl)
/rpool/ROOT on /rpool/ROOT type zfs (rw,noatime,xattr,noacl)
/rpool/data on /rpool/data type zfs (rw,noatime,xattr,noacl)
/rpool/ROOT/pve-1 on / type zfs (rw,noatime,xattr,noacl)
root@(none):/# _
```

Passt. Fertig.

neustarten mit STRG+D

und dann abort auswählen.

changes. This EULA
maintenance, update
these terms and
update, source co

1. License. Proxmox
license to the Pro
agreement for ea
permits you to ru
obligations in so
certain binary on
license rights for
This EULA pertai
rights that super
2. Limited Warrant
without warranty
merchantability,
affiliates warrant
or that the opera
as described in tl
3. Limitation of L

Abort

Nun im Terminal wieder STRG+D drücken

Das System startet neu.

Beim Starten bekommen wir wieder den Hinweis das der pool fremdimportiert wurde.

```
Reading all physical volumes. This may take a while...org)
Machine UUID ac683117-0b93-4c17-a90f-9bc3cbe16fc4
Command: /sbin/zpool import -N 'rpool'
Message: cannot import 'rpool': pool was previously in use from another system
Last accessed by proxmox (hostid=cda56b5f) at Tue Jan 3 08:14:13 2023
The pool can be imported, use 'zpool import -f' to import the pool.
Error: 1

Failed to import pool 'rpool'.
Manually import the pool and exit.

BusyBox v1.30.1 (Debian 1:1.30.1-4) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)
```

Wie in der Beschreibung ja auch zu lesen

zpool import rpool -f

```
(initramfs) zpool import rpool -f
(initramfs)
```

Nun wieder STRG+D

Und die Kiste läuft.

```
-----
Welcome to the Proxmox Virtual Environment. Please use your web browser to
configure this server - connect to:

  https://192.168.178.45:8006/
-----
```

```
pvetest64 login:
```

Jetzt einloggen und nochmal neustarten.

```
-----
Welcome to the Proxmox Virtual Environment. Please use your web browser to
configure this server - connect to:

  https://192.168.178.45:8006/
-----
```

```
pvetest64 login: root
Password:
Last login: Tue Jan  3 08:09:17 CET 2023 on tty1
Linux pvetest64 5.4.106-1-pve #1 SMP PVE 5.4.106-1 (Fri, 19 Mar 2021 11:08:47 +0100) 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.
```

```
root@pvetest64:~#
root@pvetest64:~#
root@pvetest64:~# reboot_
```

Läuft so durch

```
-----  
Welcome to the Proxmox Virtual Environment. Please use your web browser to  
configure this server - connect to:
```

```
https://192.168.178.45:8006/  
-----
```

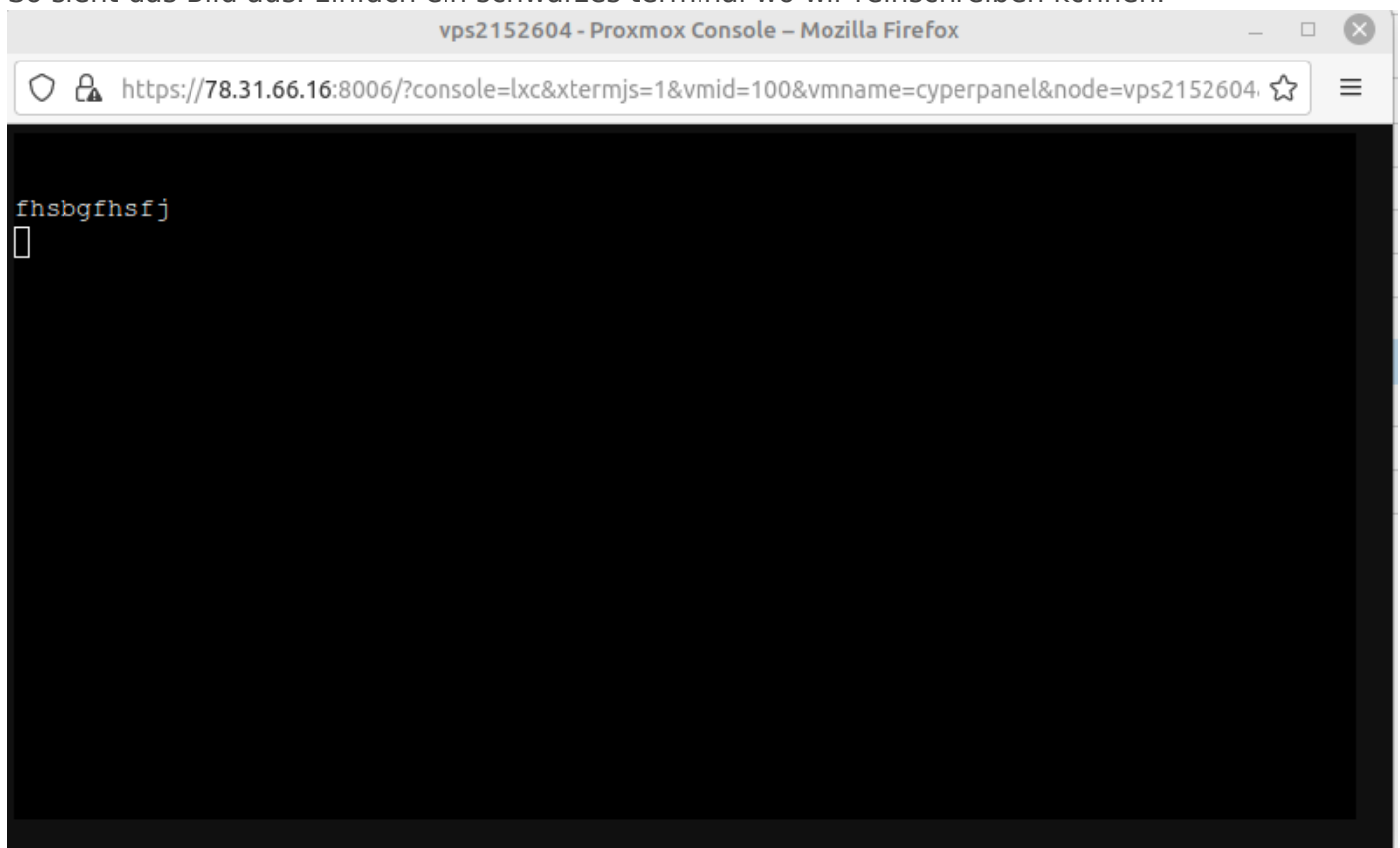
```
pvetest64 login:
```

fertig

LXC Container startet nach Installation einer Software die als Dienst startet nicht mehr. (AppArmor)

Beschreibung


LXC Container startet nach Installation einer Software die als Dienst startet nicht mehr. So sieht das Bild aus. Einfach ein schwarzes terminal wo wir reinschreiben können.


















Vorbereitung

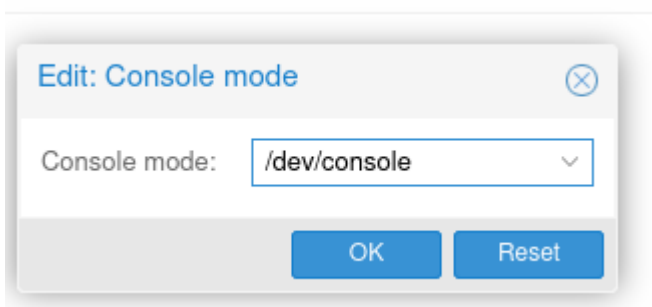
Damit wir info bekommen, warum das ding nicht startet müssen wir in den Option des Containers die Console von tty auf Console umstellen.

Dazu auf Optionen des Containers und Doppelklick auf Console

Container 100 (cyberpanel) on node 'vps2152604' No Tags 

 Summary	<input type="button" value="Edit"/> <input type="button" value="Revert"/>
>_ Console	Start at boot No
 Resources	Start/Shutdown order order=any
 Network	OS Type ubuntu
 DNS	Architecture amd64
 Options	/dev/console Enabled
 Task History	TTY count 2
 Backup	Console mode tty
 Replication	Protection No
 Snapshots	Unprivileged container No
 Firewall	Features none
 Options	
 Alias	
 IPSet	
 Log	
 Permissions	

Nun aus der Liste /dev(console auswählen und ok



Wenn wir jetzt den Container neu starten und ein neues Konsolenfenster aufmachen bekommen wir infos.

```
vps2152604 - Proxmox Console - Mozilla Firefox
https://78.31.66.16:8006/?console=lx&xtermjs=1&vmid=100&vmname=cyperpanel&node=vps2152604

[ OK ] Reached target Emergency Mode.
Starting Network Configuration...
[ OK ] Finished Flush Journal to Persistent Storage.
[ OK ] Finished Set console font and keymap.
Starting Create Volatile Files and Directories...
[ OK ] Finished Create Volatile Files and Directories.
Starting Record System Boot/Shutdown in UTMP...
[ OK ] Started Network Configuration.
[ OK ] Reached target Network.
Starting Wait for Network to be Configured...
[ OK ] Finished Record System Boot/Shutdown in UTMP.
Starting Record Runlevel Change in UTMP...
[ OK ] Finished Record Runlevel Change in UTMP.
[FAILED] Failed to start Load AppArmor profiles.
See 'systemctl status apparmor.service' for details.
[ OK ] Finished Wait for Network to be Configured.
[ OK ] Reached target Network is Online.
You are in emergency mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" or "exit"
to boot into default mode.
Give root password for maintenance
(or press Control-D to continue):
```

Nun melden wir uns mit unserem Kennwort an und sehen wir sind als root eingeloggt

```
vps2152604 - Proxmox Console - Mozilla Firefox
https://78.31.66.16:8006/?console=lx&xtermjs=1&vmid=100&vmname=cyperpanel&node=vps2152604

Starting Network Configuration...
[ OK ] Finished Flush Journal to Persistent Storage.
[ OK ] Finished Set console font and keymap.
Starting Create Volatile Files and Directories...
[ OK ] Finished Create Volatile Files and Directories.
Starting Record System Boot/Shutdown in UTMP...
[ OK ] Started Network Configuration.
[ OK ] Reached target Network.
Starting Wait for Network to be Configured...
[ OK ] Finished Record System Boot/Shutdown in UTMP.
Starting Record Runlevel Change in UTMP...
[ OK ] Finished Record Runlevel Change in UTMP.
[FAILED] Failed to start Load AppArmor profiles.
See 'systemctl status apparmor.service' for details.
[ OK ] Finished Wait for Network to be Configured.
[ OK ] Reached target Network is Online.
You are in emergency mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" or "exit"
to boot into default mode.
Give root password for maintenance
(or press Control-D to continue):
root@cyperpanel:~#
```

Nun mit journalctl -xe schauen was Sache ist. Dazu scrollen wir hoch bis zum Fehler.

```
emd[86]: /sbin/apparmor_parser: Unable to replace "lsb_release". Permission de>
emd[93]: /sbin/apparmor_parser: Unable to replace "/usr/bin/man". Permission d>
emd[92]: /sbin/apparmor_parser: Unable to replace "/usr/lib/NetworkManager/nm-d>
emd[96]: Skipping profile in /etc/apparmor.d/disable: usr.sbin.rsyslogd
emd[94]: /sbin/apparmor_parser: Unable to replace "tcpdump". Permission denied>
emd[118]: /sbin/apparmor_parser: Unable to replace "lsb_release". Permission d>
emd[121]: /sbin/apparmor_parser: Unable to replace "kmod". Permission denied; >
emd[121]: /sbin/apparmor_parser: Unable to replace "nvidia_modprobe". Permissi>
emd[133]: /sbin/apparmor_parser: Unable to replace "/usr/bin/man". Permission >
emd[131]: /sbin/apparmor_parser: Unable to replace "/usr/lib/NetworkManager/nm->
emd[139]: Skipping profile in /etc/apparmor.d/disable: usr.sbin.rsyslogd
emd[135]: /sbin/apparmor_parser: Unable to replace "tcpdump". Permission denie>
emd[77]: Error: At least one profile failed to load
pparmor.service: Main process exited, code=exited, status=1/FAILURE
```

scroll right

```
replace "lsb_release". Permission denied; attempted to load a profile while >
o replace "/usr/bin/man". Permission denied; attempted to load a profile while>
o replace "/usr/lib/NetworkManager/nm-dhcp-client.action". Permission denied; >
or.d/disable: usr.sbin.rsyslogd
o replace "tcpdump". Permission denied; attempted to load a profile while conf>
to replace "lsb_release". Permission denied; attempted to load a profile while>
to replace "kmod". Permission denied; attempted to load a profile while confin>
to replace "nvidia_modprobe". Permission denied; attempted to load a profile w>
to replace "/usr/bin/man". Permission denied; attempted to load a profile whil>
to replace "/usr/lib/NetworkManager/nm-dhcp-client.action". Permission denied;>
mor.d/disable: usr.sbin.rsyslogd
to replace "tcpdump". Permission denied; attempted to load a profile while con>
led to load
```

scroll right

```
ied; attempted to load a profile while confined?
nied; attempted to load a profile while confined?
cp-client.action". Permission denied; attempted to load a profile while confin>

attempted to load a profile while confined?
nied; attempted to load a profile while confined?
ttempted to load a profile while confined?
n denied; attempted to load a profile while confined?
```

enied; attempted to load a profile while confined?

hcp-client.action". Permission denied; attempted to load a profile while confi>

; attempted to load a profile while confined?

In dem Conatiner dann apparmor deaktivieren

```
systemctl disable apparmor
```

Dann sollte die Kiste starten.

Allerdings rate ich genau wegen solchen Dingen von Containern ab und empfehle KVM in Verbindung mit cloudinit.

Proxmox Wartungsmodus / Maintenance Mode - Disable / Enable VMS at Start

Beschreibung

Man möchte den Server mehrmals neustarten, dann nervt es doch, wenn alle VMs/CTs mit starten. Mit folgendem Script kann man das ein und ausschalten

Installation des scripte

Auf den Server einloggen (sollte es ein Cluster sein, braucht das nur auf eine Node gepackt werden)

Nun eine Neue Datei erstellen

```
nano /root/save_boot_state.sh
```

Den Inhalt einfügen

```
#!/bin/bash
# boot-save-disable.sh
# Speichert alle onboot=1 VMs/CTs und deaktiviert sie

SAVEFILE="/root/boot-state.txt"
> "$SAVEFILE"

echo "=== Speichere Boot-Status ==="

for conf in /etc/pve/qemu-server/*.conf /etc/pve/lxc/*.conf; do
    [ -f "$conf" ] || continue
    if grep -q "^onboot: 1" "$conf"; then
        id=$(basename "$conf" .conf)
```

```
type="vm"
[[ "$conf" == */lxc/* ]] && type="ct"
echo "$type $id" >> "$SAVEFILE"
sed -i 's/^onboot: 1/onboot: 0/' "$conf"
echo " Deaktiviert: $type $id"
fi
done

echo ""
echo "Gespeichert in: $SAVEFILE"
echo "$$(wc -l < "$SAVEFILE") Einträge"
```

Ausführbar machen

```
chmod +x /root/save_boot_state.sh
```

Wiederherstellung Script

```
nano /root/restore-boot-state.sh
```

Inhalt

```
#!/bin/bash
# boot-restore.sh
# Stellt onboot=1 für alle gespeicherten VMs/CTs wieder her

SAVEFILE="/root/boot-state.txt"

if [ ! -f "$SAVEFILE" ]; then
    echo "Fehler: $SAVEFILE nicht gefunden!"
    exit 1
fi

echo "=== Stelle Boot-Status wieder her ==="

while read -r type id; do
    if [ "$type" = "vm" ]; then
        conf="/etc/pve/qemu-server/${id}.conf"
    else
```

```
conf="/etc/pve/lxc/${id}.conf"
fi

if [ -f "$conf" ]; then
    if grep -q "^onboot:" "$conf"; then
        sed -i 's/^onboot: 0/onboot: 1/' "$conf"
    else
        echo "onboot: 1" >> "$conf"
    fi
    echo " Aktiviert: $type $id"
else
    echo " FEHLER: $conf nicht gefunden"
fi
done < "$SAVEFILE"

echo ""
echo "Fertig."
```

Ausführbar machen

```
chmod +x /root/restore_boot_state.sh
```

Status script

```
nano /root/status_boot_state.sh
```

Inhalt

```
#!/bin/bash
# boot-status.sh
# Zeigt aktuellen onboot-Status aller VMs/CTs

SAVEFILE="/root/boot-state.txt"

echo "=== Aktueller onboot-Status ==="
echo ""
printf "%-6s %-6s %-8s %s\n" "Typ" "ID" "onboot" "Name"
echo "-----"

for conf in /etc/pve/qemu-server/*.conf /etc/pve/lxc/*.conf; do
    [ -f "$conf" ] || continue
```

```
id=$(basename "$conf" .conf)
type="vm"; [[ "$conf" == */lxc/* ]] && type="ct"
name=$(grep "^name:" "$conf" | awk '{print $2}')
[ -z "$name" ] && name=$(grep "^hostname:" "$conf" | awk '{print $2}')
onboot=$(grep "^onboot:" "$conf" | awk '{print $2}')
[ -z "$onboot" ] && onboot="0"
printf "%-6s %-6s %-8s %s\n" "$type" "$id" "$onboot" "${name:-}"
done

echo ""
if [ -f "$SAVEFILE" ]; then
    echo "Gespeicherte Einträge in $SAVEFILE: $(wc -l < "$SAVEFILE")"
else
    echo "Kein Savefile vorhanden ($SAVEFILE)"
fi
```

Ausführbar machen

```
chmod +x /root/status_boot_state.sh
```

Proxmox Host aus VM herunterfahren

Beschreibung

Es gibt Situationen, zum Beispiel bei einer VM mit durchgereicher Grafikkarte und USB Ports, wo man den ganzen Host mit einem Doppelklick oder einem Befehl gerne herunterfahren möchte. Damit die VMs auch das Signal bekommen zum herunterfahren muss der qemu Agent installiert sein.

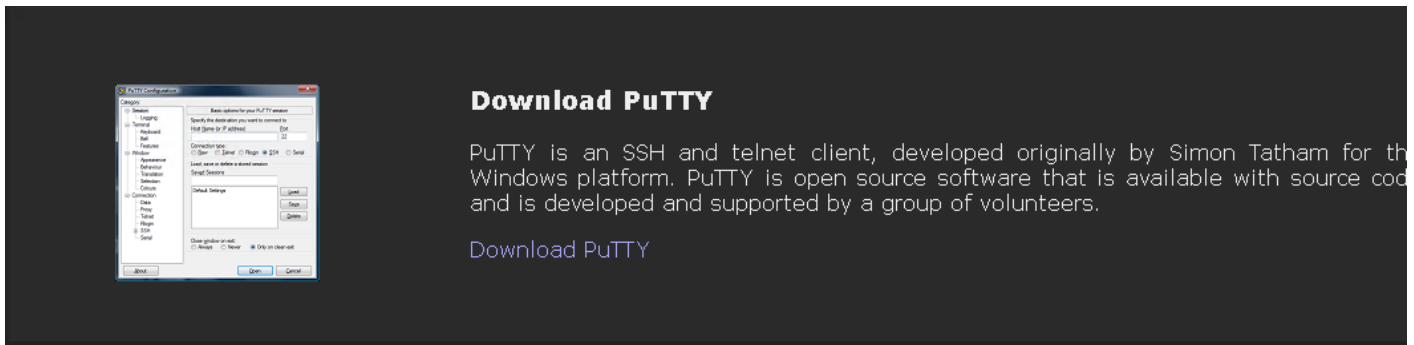
Siehe : [Proxmox - Qemu Agent installation](#)

Proxmox aus einer Windows VM herunterfahren

Als erstes downloaden wir uns plink von der putty Seite.

<https://www.putty.org>

Nun auf Download klicken

The image shows a screenshot of the PuTTY website's download page. On the left, there is a small window titled 'PuTTY Configuration' with various settings tabs like Session, Keyboard, and Connection. The main content area has the heading 'Download PuTTY' and a paragraph describing PuTTY as an SSH and telnet client. Below the text is a 'Download PuTTY' link. The background is dark.

Dort nun runterscrollen bis wir bei plink.exe ankommen.

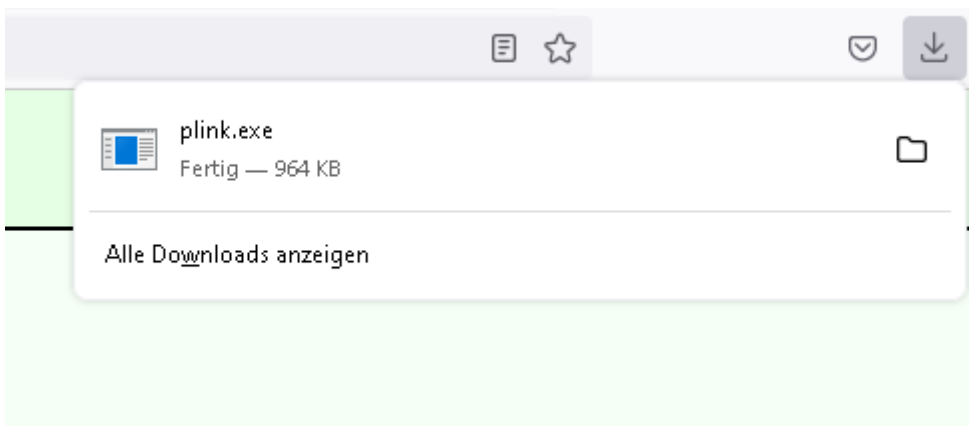
Dort für eure Architektur plink.exe anklicken

plink.exe (a command-line interface to the PuTTY back ends)

64-bit x86:	plink.exe	(signature)
64-bit Arm:	plink.exe	(signature)
32-bit x86:	plink.exe	(signature)

Da ich ein 64 Bit Windows habe, nehme ich 64bit x86 und klicke es an.

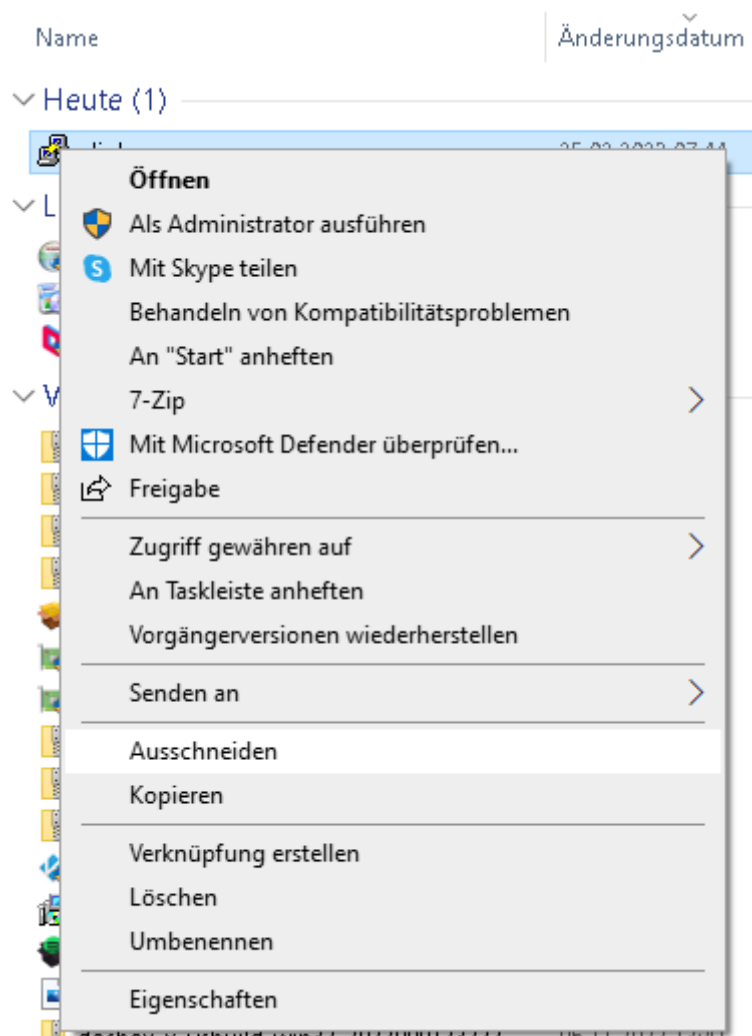
Nun klicken wir auf das Ordnersymbol neben dem download



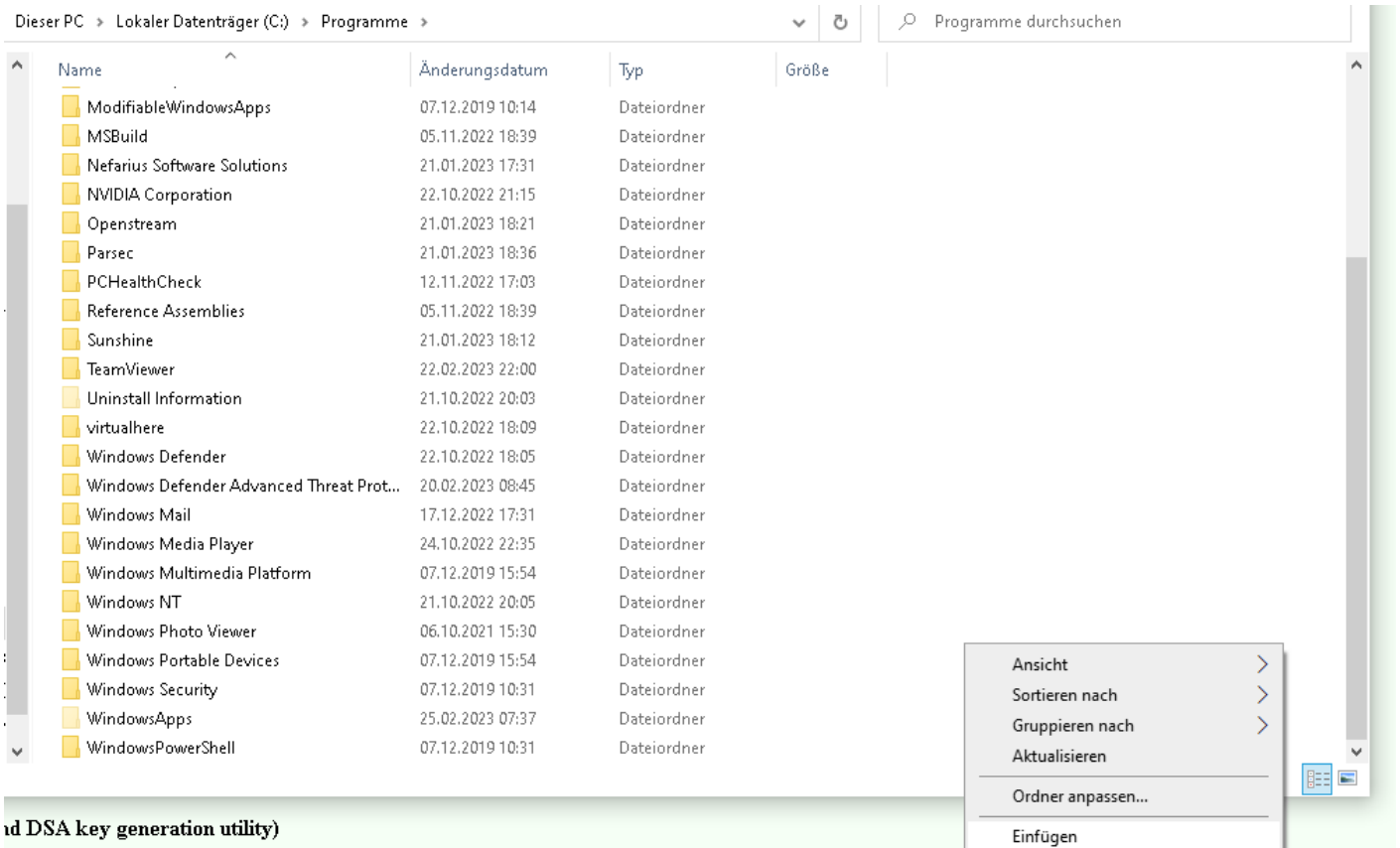
Nun öffnet sich der Download Ordner

Name	Änderungsdatum	Typ	Größe
plink.exe	25.02.2023 07:44	Anwendung	964 KB

Wir schneiden die plink exe aus

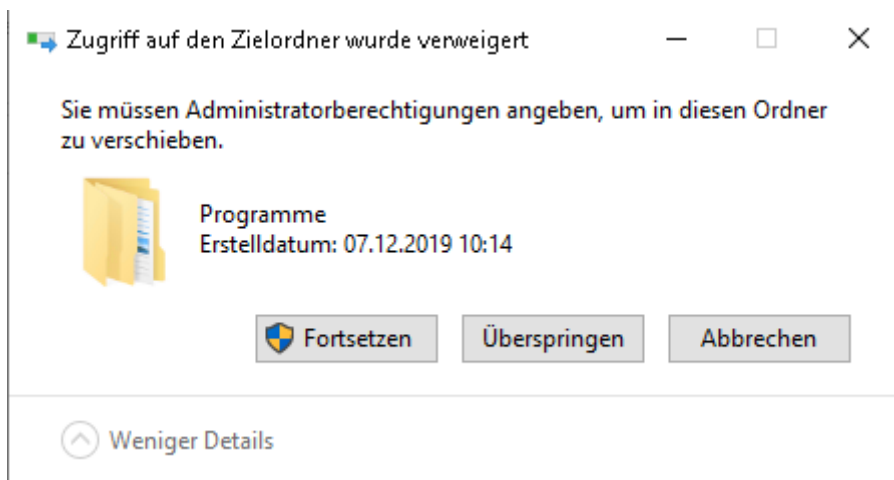


und legen sie in Program Files wieder ein



nd DSA key generation utility)

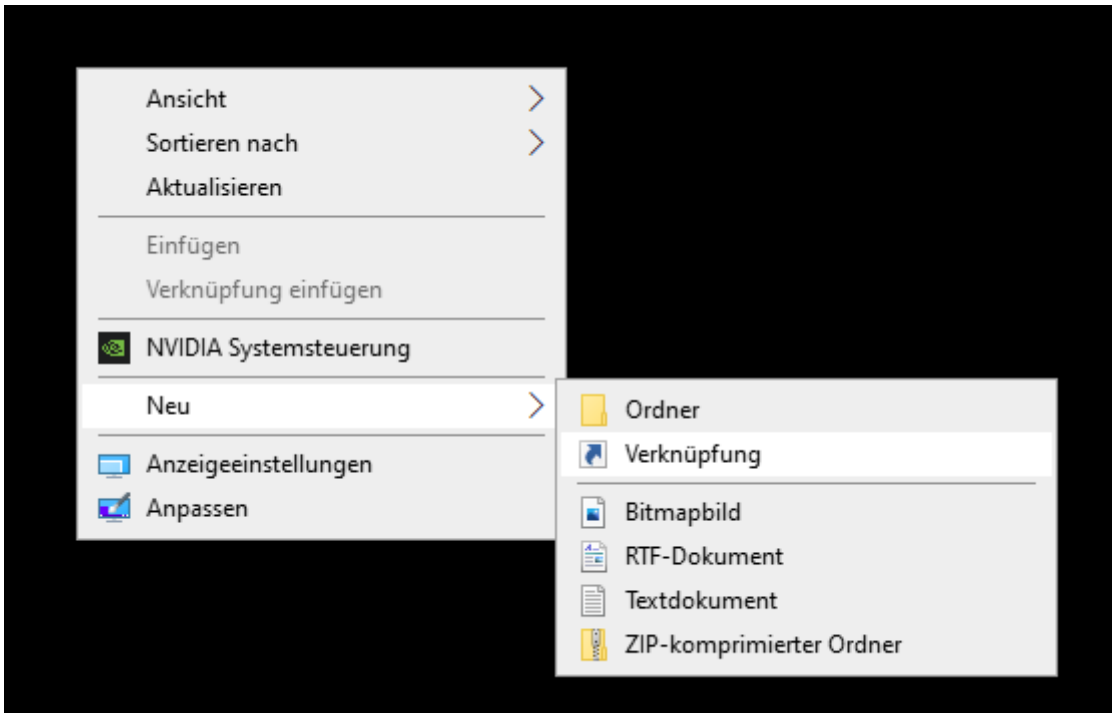
Drücken dann auf fortsetzen



Nun liegt die plink.exe im Programmverzeichnis.

Nun legen wir eine neue Verknüpfung auf dem Desktop an.

Rechtsklick auf den Desktop -> Neu -> Verknüpfung



Nun auf den Durchsuchen button klicken

- Verknüpfung erstellen

Für welche Elemente möchten Sie eine Verknüpfung erstellen?

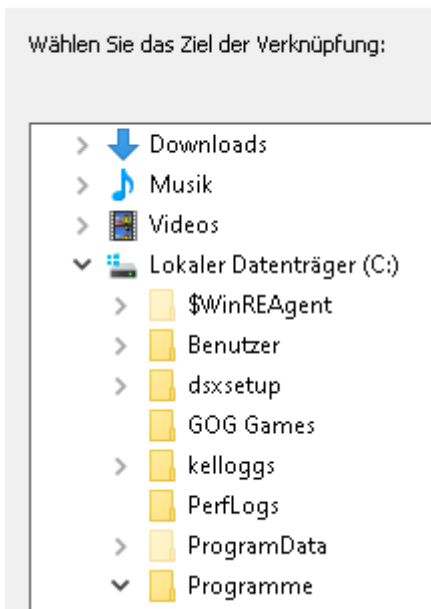
Mit diesem Assistenten können Sie Verknüpfungen mit lokalen oder vernetzten Programmen, Dateien, Ordnern, Computern oder Internetadressen erstellen.

Geben Sie den Speicherort des Elements ein:

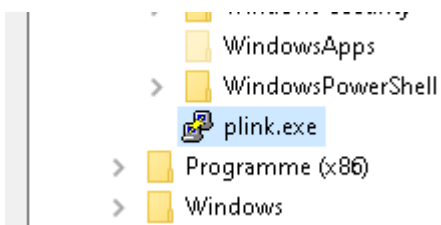
Klicken Sie auf "Weiter", um den Vorgang fortzusetzen.

Nun Auf Dieser PC -> Laufwerk C-> Proramme gehen

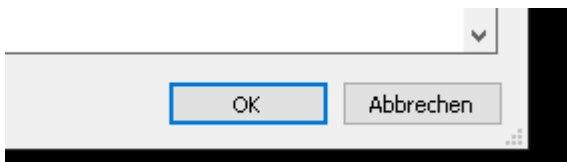
Nach Dateien oder Ordnern suchen



Nun runterscrollen bis plink.exe



Nun auf OK



Nun steht der Pfad dazu im Textfeld, hier noch NICHT auf weiter klicken, weil wir plink ein paar Paramter übergeben wollen

Für welche Elemente möchten Sie eine Verknüpfung erstellen?

Mit diesem Assistenten können Sie Verknüpfungen mit lokalen oder vernetzten Programmen, Dateien, Ordnern, Computern oder Internetadressen erstellen.

Geben Sie den Speicherort des Elements ein:

Klicken Sie auf "Weiter", um den Vorgang fortzusetzen.

Diese wären

```
"C:\Program Files\plink.exe" -ssh -pw <proxmoxpasswort> root@<proxmoxhost> poweroff
```

Beispiel: "C:\Program Files\plink.exe" -ssh -pw 12345678 root@192.168.0.100 poweroff

Dann sähe das so aus, und nun auf weiter



← Verknüpfung erstellen

Für welche Elemente möchten Sie eine Verknüpfung erstellen?

Mit diesem Assistenten können Sie Verknüpfungen mit lokalen oder vernetzten Programmen, Dateien, Ordnern, Computern oder Internetadressen erstellen.

Geben Sie den Speicherort des Elements ein:

Durchsuchen...

Klicken Sie auf "Weiter", um den Vorgang fortzusetzen.

Weiter

Abbrechen

Jetzt dem Kind noch ein Namen geben zum Beispiel Herunterfahren und dann auf Fertigstellen klicken



← Verknüpfung erstellen

Wie möchten Sie die Verknüpfung benennen?

Geben Sie den Namen für die Verknüpfung ein:

Klicken Sie auf "Fertig stellen", um die Verknüpfung zu erstellen.

Fertig stellen

Abbrechen

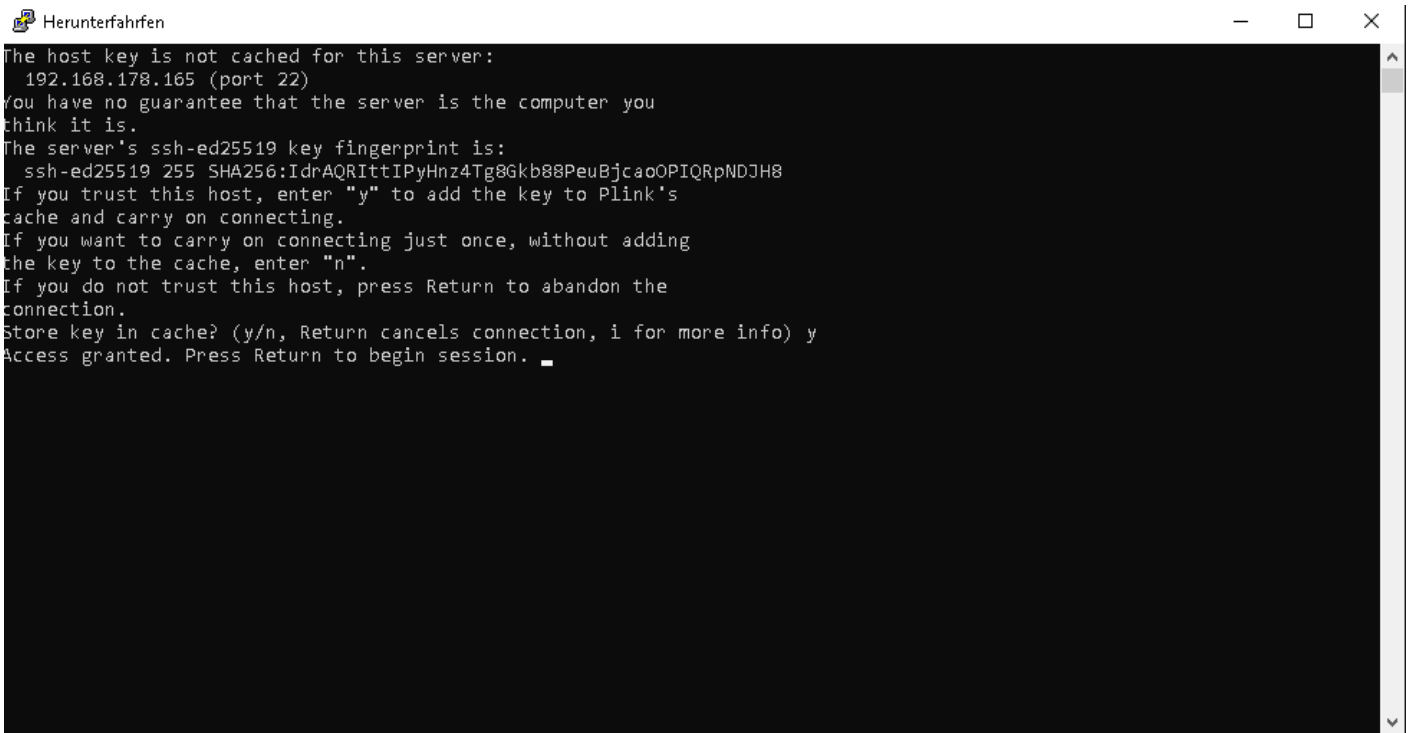
Nun haben wir eine Verknüpfung mit herunterfahren.
Doppelklick drauf die Proxmoxhost fährt runter und beendet die VMs



Nun kommt die Frage ob der Key im cache store gespeichert werden soll. Mit y bestätigen

```
Herunterfahren
The host key is not cached for this server:
 192.168.178.165 (port 22)
You have no guarantee that the server is the computer you
think it is.
The server's ssh-ed25519 key fingerprint is:
 ssh-ed25519 255 SHA256:IdrAQRIttIPyHnz4Tg8Gkb88PeuBjcaoOPIQRpMDJH8
If you trust this host, enter "y" to add the key to Plink's
cache and carry on connecting.
If you want to carry on connecting just once, without adding
the key to the cache, enter "n".
If you do not trust this host, press Return to abandon the
connection.
Store key in cache? (y/n, Return cancels connection, i for more info) _
```

Nun kommt acces granted. Einfach enter drücken



```
Herunterfahren
The host key is not cached for this server:
 192.168.178.165 (port 22)
You have no guarantee that the server is the computer you
think it is.
The server's ssh-ed25519 key fingerprint is:
 ssh-ed25519 255 SHA256:IdrAQRIttIPyHnz4Tg8Gkb88PeuBjcaoOPIQRpNDJH8
If you trust this host, enter "y" to add the key to Plink's
cache and carry on connecting.
If you want to carry on connecting just once, without adding
the key to the cache, enter "n".
If you do not trust this host, press Return to abandon the
connection.
Store key in cache? (y/n, Return cancels connection, i for more info) y
Access granted. Press Return to begin session. _
```

Nach jedem Ausführen kommt die frage ob ausgeführt werden soll. Einfach mit enter bestätigen.
nun fahren die VMs und dann der HOst herunter.

Proxmox - Es lassen sich keine Snapshots anlegen

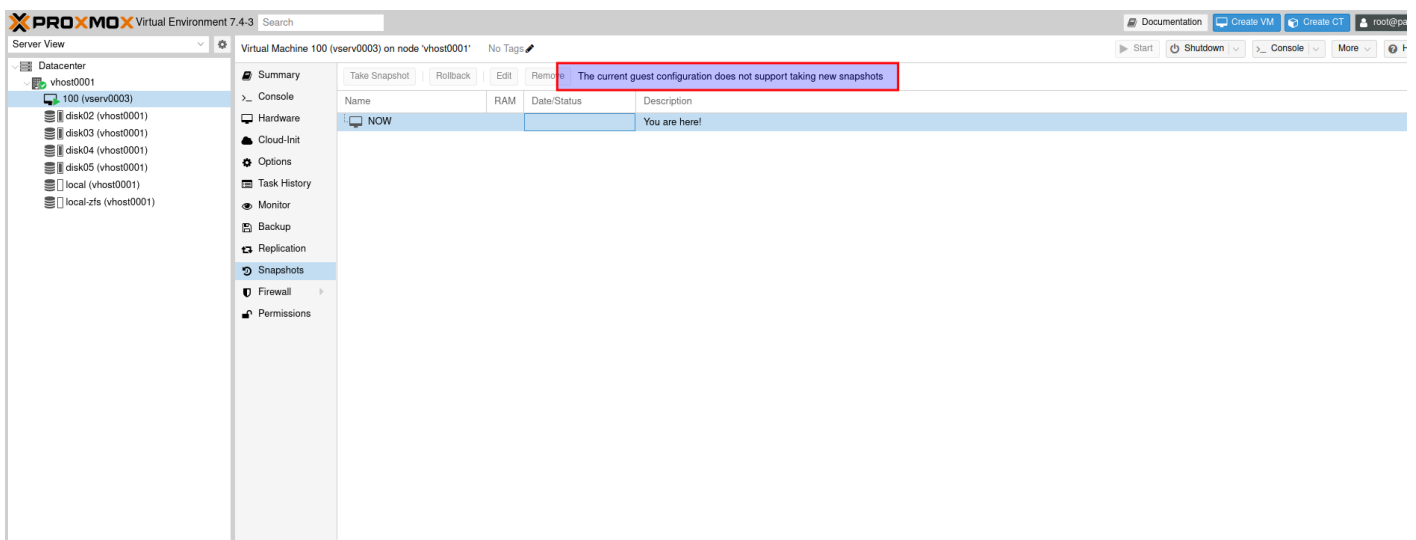
Beschreibung:

Man hat einen proxmox Host und nur den Standard local-lvm Storage.

Aber aus irgendeinem Grund kann man keine Snapshots anlegen.

Fehler :

The current guest configuration does not support taking new snapshots



Vorraussetzungen für Snap-Shots:

1. **Keine laufenden Migrationen oder Aufgaben:** Wenn eine VM migriert oder andere Aufgaben ausgeführt werden, kann es nicht möglich sein, einen Snapshot zu erstellen. Sie müssen alle laufenden Aufgaben abschließen, bevor Sie einen Snapshot erstellen können.

2. **Unterstützung auf Dateisystemebene:** Das Dateisystem der VM muss Snapshots unterstützen. Zum Beispiel werden Snapshots auf der EXT4- und XFS-Dateisystemebene nicht unterstützt.

LVM-Thin (**Nicht LVM, LVM unterstützt keine Snapshots**) und ZFS unterstützen jedoch Snapshots.

Ausnahme bei EXT4:

Nur wenn das Disk-Image der VM im QCOW2-Format (QEMU Copy-On-Write) ist.

QCOW2 ist ein von QEMU entwickeltes Dateiformat, das Snapshots auf Dateiebene unterstützt. Diese Snapshots sind unabhängig vom zugrundeliegenden Dateisystem und funktionieren daher auf Ext4. Sie sind auch wesentlich flexibler und einfacher zu

verwalten als LVM-Snapshots.

Bitte beachten Sie jedoch, dass das QCOW2-Format in einigen Situationen eine geringere Leistung aufweisen kann als RAW-Disk-Images oder LVM-basierte Speicher.

Es ist wichtig, die spezifischen Anforderungen und die Performance Ihrer Umgebung zu berücksichtigen, wenn Sie sich für ein Disk-Image-Format entscheiden.

3. **Disk-Konfiguration:** In einigen Fällen kann die Konfiguration der VM-Festplatte die Erstellung von Snapshots verhindern. Zum Beispiel unterstützen Virtio-IDE und SATA nicht das Erzeugen von Live-Snapshots (Snapshots, während die VM läuft).
4. **QEMU-Gastagent:** Der QEMU-Gastagent sollte installiert und laufend sein, wenn Sie versuchen, einen laufenden Snapshot zu erstellen (auch als Live-Snapshot bezeichnet). Der Gastagent ermöglicht eine bessere Interaktion zwischen dem Host und der VM, einschließlich der korrekten Verwaltung von Snapshots.

Wenn Ihre VM diese Anforderungen erfüllt, sollten Sie in der Lage sein, Snapshots zu erstellen. Wenn Sie trotzdem weiterhin Probleme haben, sollten Sie die Fehlermeldung und das VM-Setup genauer untersuchen.

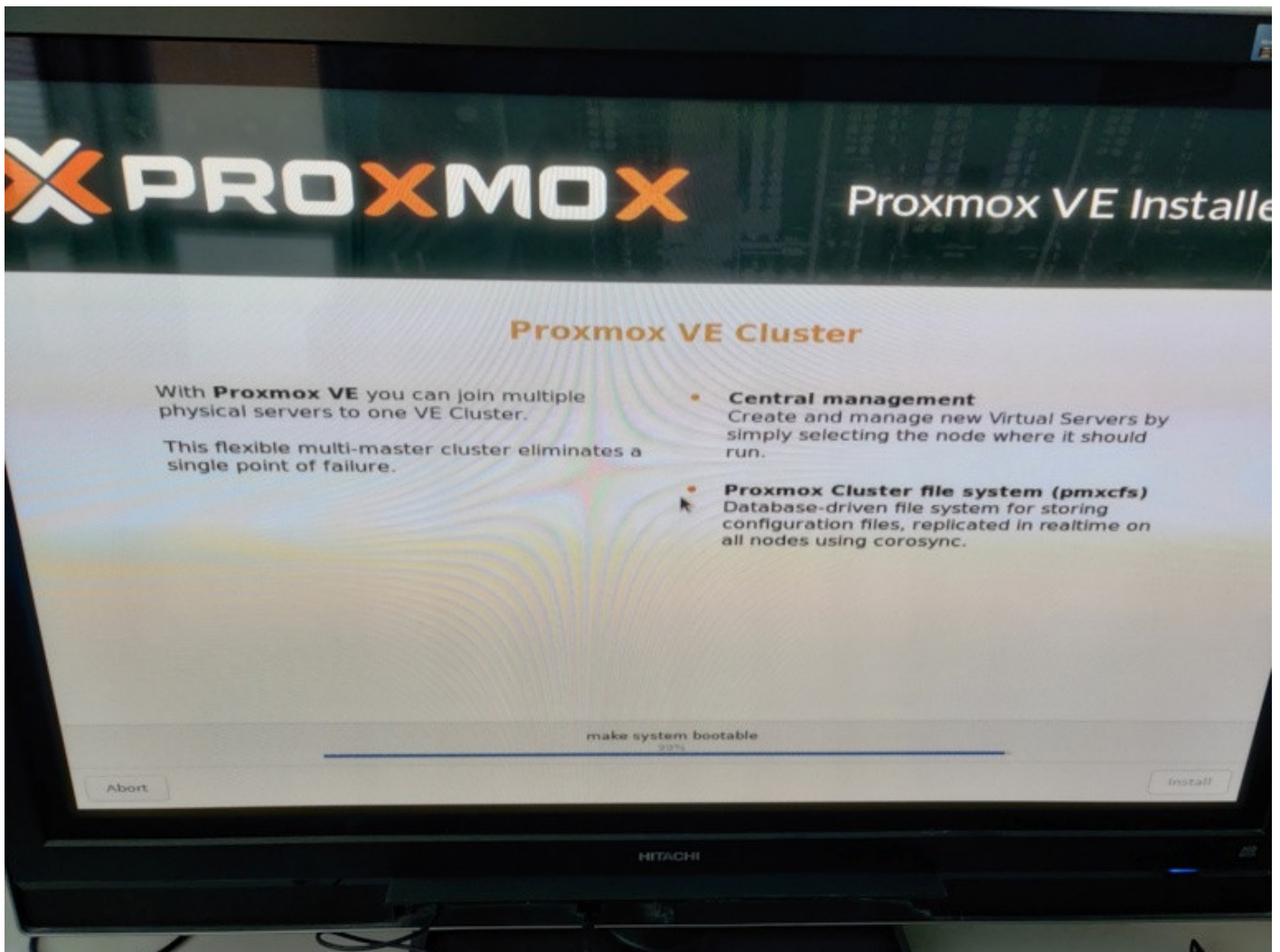
Proxmox 7 installation hängt bei 99% make system bootable

Beschreibung:

Das System bleibt stehen bei 99% make System Bootable.

Bei einem Intel Board s5520hc soll das wohl standard so sein. Da das EFI broken ist.

Kann kein NVRAM schreiben.



Was hab Ich probiert, sollte probiert werden:

Festplatte an Onboardcontroller gehängt -> kein Erfolg

Anderen Stick verwendet -> kein Erfolg

Proxmox 6.4 iso installiert, dann inplace update durchgeführt -> kein Erfolg

(falls über Ventoy gebootet werden soll, hier wird die ISO nicht gefunden. Also vom Stick der mit Schreiber für Datenträger beschrieben wurde, sonst findet er die iso nicht)

Holzhammer Methode: Festplatte in anderen Rechner hängen, dort installieren.

Wieder zurückbauen, Netzwerkkonfig ändern. -> kein Erfolg

Was brachte Erfolg:

Installieren bis Fehler 99%

Dann ausschalten, von der Proxmox ISO Starten Advanced Debug Modus auswählen.

Dann STRG+D drücken damit ein System geladen wird.

Dann nochmals STRG+D damit ein System mit richtigem Terminal geladen wird

Dann nochmals STRG+D damit Netzwerk geladen wird und der Installer.

Dann im installer auf abort klicken

Nun sind wir wieder in der shell.

apt update

apt install openssh-server

Nun die sshd config anpassen

```
nano /etc/ssh/sshd_config
```

und root zugriff erlauben

danach

```
service ssh restart
```

Nun das root Passwort neu setzen

```
passwd
```

nun mit

```
ip a
```

IP-Adresse ausgeben und per ssh verbinden.

Schlüssel akzeptieren, kennwort eingeben.

Wir sind drin ;-)

```
duffy@duffy-convert:~$ ssh root@192.168.178.70
The authenticity of host '192.168.178.70 (192.168.178.70)' can't be established.
ED25519 key fingerprint is SHA256:x4apRWrm8ushPbFMVLk1l19f8HByEidgeMvpY7a5wGU
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.178.70' (ED25519) to the list of known hosts.
root@192.168.178.70's password:
Linux proxmox 5.15.102-1-pve #1 SMP PVE 5.15.102-1 (2023-03-14T13:48Z) 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 Jun 18 07:02:24 2023
root@proxmox:/#
```

Nun den rpool mounten, wenn nichts zurück gegeben wird, hats geklappt

```
zpool import -f -R /mnt rpool
```

Nun die Verzeichnisse mounten /dev /proc /sys

```
mount --rbind /dev /mnt/rpool/ROOT/pve-1/dev
mount --rbind /proc /mnt/rpool/ROOT/pve-1/proc
mount --rbind /sys /mnt/rpool/ROOT/pve-1/sys
```

Nun müssen wir noch unsere EFI Partion herausfinden.
in unserem Beispiel ist das Laufwerk sda, bei euch anpassen

```
gdisk /dev/sda
```

Nun p drücken um die Partitionen angezeigt zu bekommen

```
isk /dev/sda: 937703088 sectors, 447.1 GiB
Model: KINGSTON SA400S3
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): E8AF2BF1-95B5-445D-89E4-E08361D101A3
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 937703054
Partitions will be aligned on 8-sector boundaries
Total free space is 0 sectors (0 bytes)
```

Number	Start (sector)	End (sector)	Size	Code	Name
1	34	2047	1007.0 KiB	EF02	
2	2048	2099199	1024.0 MiB	EF00	
3	2099200	937703054	446.1 GiB	BF01	

Die Partition mit EF00 ist unsere Efi Boot Partition
in unserem Beispiel dann /dev/sda2

Nun mounten wir die Partition in /boot/efi unser chroot umgebung

```
mount /dev/sda2 /mnt/rpool/ROOT/pve-1/boot/efi
```

Danach chrooten wir uns mit einer bash ins root Verzeichnis unserer Proxmox installtion

```
chroot /mnt/rpool/ROOT/pve-1 /bin/bash
```

Nun installieren wir endlich grub neu, aber ohne nv-ram

```
grub-install --no-nvram
```

Ausgabe:

```
root@proxmox:/# grub-install --no-nvram
Installing for x86_64-efi platform.
Installation finished. No error reported.
root@proxmox:/#
```

Grub aktualisieren

```
update-grub
```

Ausgabe, bedeutet er schaut nicht nach anderen systemen, diese werden nicht hinzugefügt

```
root@proxmox:/# update-grub
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.15.102-1-pve
Found memtest86+ image: /ROOT/pve-1@/boot/memtest86+.bin
Found memtest86+ multiboot image: /ROOT/pve-1@/boot/memtest86+_multiboot.bin
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
```

Dann exit.

Am richtigen Server wieder STRG+D drücken, dann startet der Server neu

```
aot install grub-efi-amd64
```

Nun haben wir ein System mit tools.

Dort den grubinstall neu konfigurieren

```
dpkg-reconfigure grub-efi-amd64
```

Nun herausfinden welche unsere EFI Partition ist.

```
lsblk -o +FSTYPE
```

Die wählen wo VFAT steht, also /dev/sda2

Ausgabe:

Nun den rpool mounten

```
zpool import -f -R /mnt rpool
```

Nun die Verzeichnisse mounten die wir brauchen

```
mount -o rbind /proc /mnt/rpool/ROOT/pve-1/proc
mount -o rbind /sys /mnt/rpool/ROOT/pve-1/sys
mount -o rbind /dev /mnt/rpool/ROOT/pve-1/dev
mount -o rbind /run /mnt/rpool/ROOT/pve-1/run
```

Nun in das root Verzeichnis vom pve-1 chrooten

```
chroot /mnt/rpool/ROOT/pve-1 /bin/bash
```

Jetzt den Bootloader neu installieren.

Dazu müssen wir vorher die EFI Partion mounten.
In meinem Beispiel ist das die 1GB partition

```
mount /dev/sda2 /mnt
```

Nun können wir den Befehl zum Grub installieren absetzen

Anlegen / Clonen einer ARM64 VM Fehler

Beschreibung:

Ab Version 8.1 muss das Firmware Paket für arm64 manuell nachinstalliert werden.
Sonst bekommtman diesen Fehler:

```
TASK ERROR: clone failed: EFI base image '/usr/share/pve-edk2-firmware//AAVMF_CODE.fd' not found
```

Lösung:

Folgendes Paket manuell installieren

```
apt install pve-edk2-firmware-aarch64
```

Fertig.
nun lässt sich die VM wieder erstellen / clonen

LXC Container - Routen beim start setzen, da post in interfaces nicht geht

Beschreibung:

leider unterstützen LXC keine einträge in der /etc/interfaces weil die bei jedem Start wieder gelöscht werden.

Abhilfe mit einem Systemd service

Einrichtung

Ein script erstellen was die route anlegt:

```
nano /usr/local/bin/set-routes.sh
```

Inhalt

```
route add route add <zieladresse oder netz> gw <gateway> dev <netzwerkkarte>
```

```
route add 185.26.156.118 gw 222.222.0.7 dev eth1
```

bei einem netz sähe das so aus

bei einem 28er netz

```
route add 185.26.156.0/28 gw 222.222.0.7 dev eth1
```

bei einem 24 netz

```
route add 185.26.156.24 gw 222.222.0.7 dev eth1
```

also nun das script

```
#!/bin/bash
```

```
route add route add <zieladresse oder netz> gw <gateway> dev <netzwerkkarte>
```

Script ausführbar machen

```
chmod +x /usr/local/bin/set-routes.sh
```

Nun einen Systemd Service Datei anlegen

```
nano /etc/systemd/system/set-routes.service
```

Inhalt:

```
[Unit]
Description=Set custom routes after network is up
After=network.target

[Service]
Type=oneshot
ExecStart=/usr/local/bin/set-routes.sh
RemainAfterExit=true

[Install]
WantedBy=multi-user.target
```

Nun den systemctl neuladen den Dienst aktivieren und starten

```
systemctl daemon-reload
systemctl enable set-routes.service
systemctl start set-routes.service
```

Fertig

Ceph Upgrade von Pacific 17.2.7 auf reef 18.2.2 osd starten nicht

Beschreibung:

Beim Upgrade von 17.2.7 auf 18.2.2 starten die OSD nicht.

Lösung:

Erstmal auf 18.2.1 downgraden, dann warten bis die osds gestartet sind.

```
apt install ceph=18.2.1-pve2 ceph-mon=18.2.1-pve2 ceph-mgr=18.2.1-pve2 ceph-osd=18.2.1-pve2 ceph-  
mds=18.2.1-pve2 ceph-base=18.2.1-pve2 librados2=18.2.1-pve2 ceph-mgr-modules-core=18.2.1-pve2  
libradosstriper1=18.2.1-pve2 libsqlite3-mod-ceph=18.2.1-pve2 librados2=18.2.1-pve2 librbd1=18.2.1-pve2  
ceph-common=18.2.1-pve2 libcephfs2=18.2.1-pve2 python3-cephfs=18.2.1-pve2 python3-ceph-  
argparse=18.2.1-pve2 python3-ceph-common=18.2.1-pve2 python3-rados=18.2.1-pve2 python3-rbd=18.2.1-  
pve2 python3-rgw=18.2.1-pve2 librgw2=18.2.1-pve2 ceph-fuse=18.2.1-pve2 ceph-volume=18.2.1-pve2
```

Nun die min version zurück auf quincy setzten

```
ceph osd require-osd-release quincy
```

Danach leider die node neustarten, da aber eh alle OSDs offline sind, wäre das bei der Node auch egal.

Sollte das auch nicht gehen dann downgrade auf 17.2.7 zurück

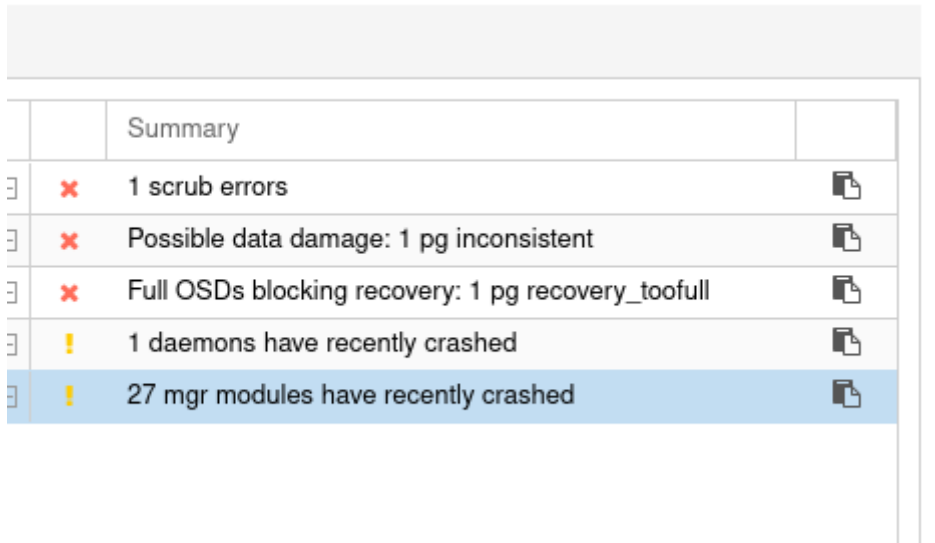
```
apt install ceph=17.2.7-pve3 ceph-mon=17.2.7-pve3 ceph-mgr=17.2.7-pve3 ceph-osd=17.2.7-pve3 ceph-  
mds=17.2.7-pve3 ceph-base=17.2.7-pve3 librados2=17.2.7-pve3 ceph-mgr-modules-core=17.2.7-pve3
```






libradosstriper1=17.2.7-pve3 libsqlite3-mod-ceph=17.2.7-pve3 librados2=17.2.7-pve3 librbd1=17.2.7-pve3
ceph-common=17.2.7-pve3 libcephfs2=17.2.7-pve3 python3-cephfs=17.2.7-pve3 python3-ceph-
argparse=17.2.7-pve3 python3-ceph-common=17.2.7-pve3 python3-rados=17.2.7-pve3 python3-rbd=17.2.7-
pve3 python3-rgw=17.2.7-pve3 librgw2=17.2.7-pve3 ceph-fuse=17.2.7-pve3 ceph-volume=17.2.7-pve3

Ceph crash log löschen

Beschreibung:

Wenn ein Daemon crasht bleibt da in der log.
um diese zu löschen folgenden Befehl ausführen



	Summary	
[-]	x 1 scrub errors	
[-]	x Possible data damage: 1 pg inconsistent	
[-]	x Full OSDs blocking recovery: 1 pg recovery_toofull	
[-]	! 1 daemons have recently crashed	
[-]	! 27 mgr modules have recently crashed	

Befehl:

To display a list of messages:

```
ceph crash ls
```

Wenn wir die Nachricht lesen wollen:

```
ceph crash info <id>
```

dann:

```
ceph crash archive <id>
```

or alle archivieren:

```
ceph crash archive-all
```


HDD Tray ermitteln LED blinken und/oder austauschen

Beschreibung:

Es gibt ein tolles Tool. Mit der man die LED zum Tray blinken lassen kann.

Noch ein kleines Script dazu wo man nur dass device angeben muss. Ein weiteres script zum Datenträger tausch.

Lässt slot blinken.

festplatte tauschen

Alte festplatte wird deregistriert und neue hinzugefügt.

LED ausgeschaltet.

BAM

Durchführung:

```
apt-get install -y lsscsi sg3-utils
```

Festplatten anzeigen lassen:

```
lsscsi -g
```

Ausgabe:

```
[1:0:0:0] cd/dvd Slimtype DVD A DS8ACSH LP5M /dev/sr0 /dev/sg17
[3:0:0:0] disk SEAGATE XS400ME70004 0004 /dev/sda /dev/sg0
[3:0:1:0] disk SEAGATE ST20000NM002D E004 /dev/sdb /dev/sg1
[3:0:2:0] disk WDC WUS721010AL5204 C980 /dev/sdc /dev/sg2
[3:0:3:0] disk SEAGATE ST20000NM002D E006 /dev/sdd /dev/sg3
[3:0:4:0] disk WDC WUS721010AL5204 C980 /dev/sde /dev/sg4
[3:0:5:0] disk SEAGATE ST20000NM002D E006 /dev/sdf /dev/sg5
[3:0:6:0] disk WDC WUS721010AL5204 C980 /dev/sdg /dev/sg6
```

```
[3:0:7:0] disk WDC WUS721010AL5204 C980 /dev/sdh /dev/sg7
[3:0:8:0] disk WDC WUS721010AL5204 C980 /dev/sdi /dev/sg8
[3:0:9:0] disk WDC WUS721010AL5204 C980 /dev/sdj /dev/sg9
[3:0:10:0] disk WDC WUS721010AL5204 C980 /dev/sdk /dev/sg10
[3:0:11:0] disk SEAGATE XS400ME70004 0004 /dev/sdl /dev/sg11
[3:0:12:0] disk WDC WUS721010AL5204 C980 /dev/sdm /dev/sg12
[3:0:13:0] disk WDC WUS721010AL5204 C980 /dev/sdn /dev/sg13
[3:0:14:0] disk WDC WUS721010AL5204 C980 /dev/sdo /dev/sg14
[3:0:15:0] disk WDC WUS721010AL5204 C980 /dev/sdp /dev/sg15
[3:0:16:0] enclosu BROADCOM VirtualSES 03 - /dev/sg16
[4:0:0:0] cd/dvd AMI Virtual CDROM0 1.00 /dev/sr1 /dev/sg18
```

Script:

```
nano /usr/local/bin/identify-disk.sh
```

Inhalt

```
#!/usr/bin/env bash
set -euo pipefail

usage() {
    cat <<USAGE
Usage:
    identify-disk [--on|--off|--blink SECONDS] [--symlink|--ses] [--all] [/dev/sdX|sdX|sdXn]

Optionen:
    --on / --off / --blink SECONDS  LED steuern
    --symlink                        nur sysfs/enclosure_device* nutzen
    --ses                            nur SES (sg_ses) nutzen
    --all                            auf alle verfügbaren Slots anwenden
    (ohne Methode)                   auto: erst symlink, dann SES-Fallback

Beispiele:
    identify-disk --on    sdi
    identify-disk --blink 10 sdg8
    identify-disk --off   --symlink sdi
    identify-disk --on    --ses    sdi
    identify-disk --off   --all     # alle LEDs aus
    identify-disk --blink 5 --all --symlink # alle via sysfs blinken
```

USAGE

}

MODE="on"

BLINK_SECS=0

METHOD="auto" # auto | symlink | ses

ALL=0

DISK=""

[[\$# -lt 1]] && { usage; exit 1; }

while [[\$# -gt 0]]; do

case "\${1:-}" in

--on) MODE="on"; shift ;;

--off) MODE="off"; shift ;;

--blink) MODE="blink"; BLINK_SECS="\${2:-}"; [[-z "\$BLINK_SECS"]] && { echo "Fehlende Sekunden bei --blink"; exit 1; }; shift 2 ;;

--symlink) METHOD="symlink"; shift ;;

--ses) METHOD="ses"; shift ;;

--all) ALL=1; shift ;;

-h|--help) usage; exit 0 ;;

*)

DISK="\$1"; shift ;;

esac

done

Helpers (norm/req/printing)

need() { command -v "\$1" >/dev/null 2>&1 || { echo "\$1 fehlt. Bitte installieren."; exit 1; }; }

normalize_disk() {

local d="\$1"

[["\$d" =~ ^sd]] && d="/dev/\$d"

local base="\$(basename "\$d")"

local base_nopart="\${base%%[0-9]*}"

echo "/dev/\$base_nopart"

}

```

# Einzel-Operationen (SYMLINK / per Disk)
# -----
do_symlink_one() {
    local disk="$1"
    local base_nopart="$(basename "$disk")"
    local p="/sys/class/block/${base_nopart}/device"
    local enc=( "$p"/enclosure_device* )
    [[ -e "${enc[0]}" ]] || { echo "Kein enclosure_device* für $disk gefunden."; return 2; }
    local LOCATE="${enc[0]}/locate"
    [[ -w "$LOCATE" ]] || { echo "'locate' nicht verfügbar/schreibbar unter ${enc[0]}."; return 3; }
    local slot="$(cat "${enc[0]}/slot" 2>/dev/null || echo "")"
    local slot_lbl=""; [[ -n "$slot" ]] && slot_lbl=" (Slot $slot)"
    case "$MODE" in
        on)  echo 1 | sudo tee "$LOCATE" >/dev/null; echo "ON via sysfs$slot_lbl - $disk";;
        off) echo 0 | sudo tee "$LOCATE" >/dev/null; echo "OFF via sysfs$slot_lbl - $disk";;
        blink) echo 1 | sudo tee "$LOCATE" >/dev/null; echo "ON via sysfs$slot_lbl - $disk"; sleep "$BLINK_SECS";
    echo 0 | sudo tee "$LOCATE" >/dev/null; echo "OFF via sysfs$slot_lbl - $disk";;
    esac
}

# -----
# Einzel-Operationen (SES / per Disk)
# -----
do_ses_one() {
    local disk="$1"
    need lsscsi; need sg_ses
    local LINE
    LINE="$(lsscsi -g | awk -v d="$disk" '{if (($NF-1)==d) {print; exit}}')"
    [[ -n "$LINE" ]] || { echo "Konnte $disk nicht in 'lsscsi -g' finden."; return 6; }

    local HOST BUS TARGET LUN
    if [[ "$LINE" =~ \[([0-9]+):([0-9]+):([0-9]+):([0-9]+)\] ]]; then
        HOST="${BASH_REMATCH[1]}"; BUS="${BASH_REMATCH[2]}"; TARGET="${BASH_REMATCH[3]}";
LUN="${BASH_REMATCH[4]}"
    else
        echo "Konnte SCSI-Adresse nicht parsen."; return 7
    fi

    local SES_LINE SES_DEV
    SES_LINE="$(lsscsi -g | awk -v h="$HOST" -v b="$BUS" '$2=="enclosu" && $1 ~ ("\\[\"h\":\"b\":") {print; exit}')"

```

```

[[ -n "$SES_LINE" ]] || { echo "Kein SES/Enclosure auf ${HOST}:${BUS} gefunden (HW-RAID? Hersteller-CLI
nötig)."; return 8; }
SES_DEV="$(echo "$SES_LINE" | awk '{print $NF}')"
[[ -e "$SES_DEV" ]] || { echo "SES-Gerät existiert nicht: $SES_DEV"; return 9; }

local SAS_PATH="/sys/class/scsi_device/${HOST}:${BUS}:${TARGET}:${LUN}/device/sas_address"
[[ -r "$SAS_PATH" ]] || { echo "SAS-Adresse nicht lesbar: $SAS_PATH"; return 10; }
local DISK_SAS; DISK_SAS="$(tr -d '\n' < "$SAS_PATH")"
[[ -n "$DISK_SAS" ]] || { echo "Leere SAS-Adresse."; return 11; }

local INDEX SLOTNUM
read INDEX SLOTNUM < <(sg_ses -p aes "$SES_DEV" 2>/dev/null \
| awk -v want="$DISK_SAS" '
/Element index:/      {idx=$3; gsub(":", "", idx); slot=""}
/device slot number:/  {slot=$5}
/SAS address:/ && $3 ~ /^0x/ {sas=$3; if (tolower(sas)==tolower(want)) {print idx, slot; exit}}
')
[[ -n "${INDEX:-}" ]] || { echo "Konnte SES-Index für $disk nicht ermitteln."; return 12; }

local slot_lbl=""; [[ -n "${SLOTNUM:-}" ]] && slot_lbl=" (Slot $SLOTNUM, idx $INDEX)" || slot_lbl=" (idx
$INDEX)"
case "$MODE" in
on)  sg_ses --index="$INDEX" --set=ident "$SES_DEV"; echo "ON via SES$slot_lbl - $disk";;
off) sg_ses --index="$INDEX" --clear=ident "$SES_DEV"; echo "OFF via SES$slot_lbl - $disk";;
blink) sg_ses --index="$INDEX" --set=ident "$SES_DEV"; echo "ON via SES$slot_lbl - $disk"; sleep
"$BLINK_SECS"; sg_ses --index="$INDEX" --clear=ident "$SES_DEV"; echo "OFF via SES$slot_lbl - $disk";;
esac
}

# -----
# ALL-Operationen (SYMLINK)
# -----
do_symlink_all() {
local disks=()
for d in /sys/class/block/sd*; do
[[ -e "$d" ]] || continue
local name="$(basename "$d")"
[[ "$name" =~ [0-9]$ ]] && continue # nur Whole Disks
comp=( "$d"/device/enclosure_device* )
[[ -e "${comp[0]}" ]] || continue

```

```

disks+=( "/dev/$name" )
done
local n="${#disks[@]}"
[[ $n -gt 0 ]] || { echo "Keine sysfs/enclosure Slots gefunden."; return 2; }

if [[ "$MODE" == "blink" ]]; then
    # Alle an, warten, alle aus
    for disk in "${disks[@}"; do
        local loc=( "/sys/class/block/${(basename "$disk")}/device/enclosure_device*/locate )
        [[ -w "${loc[0]}" ]] || continue
        echo 1 | sudo tee "${loc[0]}" >/dev/null
        local slot=""; [[ -r "${loc[0]%/*/slot}" ]] && slot="$(cat "${loc[0]%/*/slot}")"
        echo "ON via sysfs$( [[ -n "$slot" ]] && echo " (Slot $slot)" ) - $disk"
    done
    sleep "$BLINK_SECS"
    for disk in "${disks[@}"; do
        local loc=( "/sys/class/block/${(basename "$disk")}/device/enclosure_device*/locate )
        [[ -w "${loc[0]}" ]] || continue
        echo 0 | sudo tee "${loc[0]}" >/dev/null
        echo "OFF via sysfs - $disk"
    done
else
    local val=$(( [[ "$MODE" == "on" ]] && echo 1 || echo 0)
    for disk in "${disks[@}"; do
        local loc=( "/sys/class/block/${(basename "$disk")}/device/enclosure_device*/locate )
        [[ -w "${loc[0]}" ]] || continue
        echo $val | sudo tee "${loc[0]}" >/dev/null
        local slot=""; [[ -r "${loc[0]%/*/slot}" ]] && slot="$(cat "${loc[0]%/*/slot}")"
        echo "$([ [ $val -eq 1 ] ] && echo ON || echo OFF ) via sysfs$( [[ -n "$slot" ]] && echo " (Slot $slot)" ) - $disk"
    done
fi
}

# -----
# ALL-Operationen (SES)
# -----
do_ses_all() {
    need lsscsi; need sg_ses
    mapfile -t ENCS <<(lsscsi -g | awk '$2=="enclosu"{print $NF}')
    [[ ${#ENCS[@]} -gt 0 ]] || { echo "Kein SES/Enclosure gefunden."; return 3; }
}

```

```

for SES_DEV in "${ENC[@]}"; do
# Alle belegten Element-Indices sammeln (die eine SAS address haben)
mapfile -t IDX < <(sg_ses -p aes "$SES_DEV" 2>/dev/null \
| awk '
/Element index:/ {idx=$3; gsub(":", "", idx); slot=""; has_sas=0}
/device slot number:/ {slot=$5}
/SAS address:/ && $3 ~ /^0x/ {has_sas=1}
has_sas && /SAS address:/ {print idx "|" slot}
')
[[ $#IDX[@] -gt 0 ]] || { echo "Keine befüllten Slots für $SES_DEV."; continue; }

if [[ "$MODE" == "blink" ]]; then
# alle an
for pair in "${IDX[@]}"; do
IFS='|' read -r index slot <<<"$pair"
sg_ses --index="$index" --set=ident "$SES_DEV" || true
[[ -n "$slot" ]] && echo "ON via SES (Slot $slot, idx $index) - $SES_DEV" || echo "ON via SES (idx $index) -
$SES_DEV"
done
sleep "$BLINK_SECS"
# alle aus
for pair in "${IDX[@]}"; do
IFS='|' read -r index slot <<<"$pair"
sg_ses --index="$index" --clear=ident "$SES_DEV" || true
[[ -n "$slot" ]] && echo "OFF via SES (Slot $slot, idx $index) - $SES_DEV" || echo "OFF via SES (idx $index) -
$SES_DEV"
done
else
local cmd=$(( [[ "$MODE" == "on" ]] && echo "--set=ident" || echo "--clear=ident")
for pair in "${IDX[@]}"; do
IFS='|' read -r index slot <<<"$pair"
sg_ses --index="$index" $cmd "$SES_DEV" || true
local state=$(( [[ "$MODE" == "on" ]] && echo ON || echo OFF)
[[ -n "$slot" ]] && echo "$state via SES (Slot $slot, idx $index) - $SES_DEV" || echo "$state via SES (idx
$index) - $SES_DEV"
done
fi
done
}

```

```

# -----
# Main-Dispatch
# -----
if [[ $ALL -eq 1 ]]; then
  case "$METHOD" in
    symlink)
      do_symlink_all
      ;;
    ses)
      do_ses_all
      ;;
    auto)
      # erst symlink-Variante probieren; wenn nichts gefunden, SES-all
      if ! do_symlink_all; then
        echo "(Auto) Fallback auf SES (alle)..."
        do_ses_all
      fi
      ;;
    *) echo "Unbekannte Methode: $METHOD"; usage; exit 1 ;;
  esac
  exit 0
fi

# Einzel-Disk-Pfad (keine --all)
[[ -n "$DISK" ]] || { echo "Bitte /dev/sdX angeben oder --all nutzen."; exit 1; }
DISK="$(normalize_disk "$DISK")"
[[ -b "$DISK" ]] || { echo "Gerät nicht gefunden: $DISK"; exit 1; }

case "$METHOD" in
  symlink)
    do_symlink_one "$DISK" || exit $?
    ;;
  ses)
    do_ses_one "$DISK" || exit $?
    ;;
  auto)
    if do_symlink_one "$DISK"; then
      exit 0
    else

```

```
    echo "(Auto) Fallback auf SES..."
    do_ses_one "$DISK"
fi
;;
*)
    echo "Unbekannte Methode: $METHOD"; usage; exit 1 ;;
esac
```

Ausführbar machen

```
chmod +x /usr/local/bin/identify-disk.sh
```

Hilfe/Bedienung

Usage:

```
identify-disk [--on|--off|--blink SECONDS] [--symlink|--ses] [--all] [/dev/sdX|sdX|sdXn]
```

Optionen:

```
--on / --off / --blink SECONDS  LED steuern
--symlink                        nur sysfs/enclosure_device* nutzen
--ses                            nur SES (sg_ses) nutzen
--all                            auf alle verfügbaren Slots anwenden
(ohne Methode)                  auto: erst symlink, dann SES-Fallback
```

Beispiele:

```
identify-disk --on    sdi
identify-disk --blink 10 sdg8
identify-disk --off   --symlink sdi
identify-disk --on    --ses  sdi
identify-disk --off   --all   # alle LEDs aus
identify-disk --blink 5 --all --symlink # alle via sysfs blinken
```

Alte Laufwerke entfernen:

Sobald alte Laufwerke entfernt werden, bleiben Sie tot in der Liste.

Erst Laufwerk entfernen. Dann neu einschieben.

Das machen wir wieder mit einem script. Das Script wartet solange bis wir die festplatte getauscht haben.

wenn getauscht bestätigen.

```
nano /usr/local/bin/scsi-hotswap.sh
```

Inhalt

```
#!/usr/bin/env bash
set -euo pipefail

usage() {
    cat <<USAGE
Usage:
    scsi-hotswap [--method auto|symlink|ses] [--blink N|--no-led] [--no-prompt] /dev/sdX|sdX|sdXn

Optionen:
--method M    M = auto (default) | symlink | ses
--blink N     LED vor dem Tausch N Sekunden blinken (statt dauerhaft an)
--no-led      keine LED-Steuerung
--no-prompt   nicht auf ENTER warten (setze voraus, dass neue Platte schon steckt)

Beispiele:
scsi-hotswap sdi
scsi-hotswap --method symlink --blink 8 sdg8
scsi-hotswap --method ses --no-prompt sdm
USAGE
}

METHOD="auto"    # auto | symlink | ses
BLINK_SECS=0    # 0 = steady on
USE_LED=1
PROMPT=1

[[ $# -lt 1 ]] && { usage; exit 1; }
while [[ $# -gt 0 ]]; do
    case "${1:-}" in
        --method) METHOD="${2:-}"; shift 2 ;;
        --blink) BLINK_SECS="${2:-}"; [[ -z "$BLINK_SECS" ]] && { echo "Fehlende Sekunden bei --blink"; exit 1; };
        shift 2 ;;
        --no-led) USE_LED=0; shift ;;
        --no-prompt) PROMPT=0; shift ;;
        -h|--help) usage; exit 0 ;;
        *) DISK="$1"; shift ;;
    esac
done
```

```

esac
done

# --- Device normalisieren ---
[[ "$DISK" =~ ^sd ]] && DISK="/dev/$DISK"
BASE="$(basename "$DISK")"
BASE_NOPART="${BASE%%[0-9]*}"
DISK="/dev/$BASE_NOPART"
[[ -b "$DISK" ]] || { echo "Gerät nicht gefunden: $DISK"; exit 1; }

need() { command -v "$1" >/dev/null 2>&1 || { echo "$1 fehlt. Bitte installieren."; exit 1; }; }

# -----
# LED via sysfs (symlink-Weg)
# -----
# merkt sich LOCATE_PATH_CACHED+SLOT_CACHED, damit "off" nach dem Swap immer klappt
LOCATE_PATH_CACHED=""
SLOT_CACHED=""

led_symlink() {
    local mode="$1" # on|off|blink
    local LOCATE_PATH=""
    local SLOT=""

    # bereits gecachten Pfad benutzen, wenn vorhanden
    if [[ -n "$LOCATE_PATH_CACHED" && -e "$LOCATE_PATH_CACHED" ]]; then
        LOCATE_PATH="$LOCATE_PATH_CACHED"
        SLOT="$SLOT_CACHED"
    else
        local p="/sys/class/block/${BASE_NOPART}/device"
        local enc=( "$p"/enclosure_device* )
        [[ -e "${enc[0]}" ]] || return 2
        LOCATE_PATH="${enc[0]}/locate"
        [[ -w "$LOCATE_PATH" ]] || return 3
        SLOT="$(cat "${enc[0]}/slot" 2>/dev/null || echo "")"
        LOCATE_PATH_CACHED="$LOCATE_PATH"
        SLOT_CACHED="$SLOT"
    fi

    local slot_label=""

```

```

[[ -n "$SLOT" ]] && slot_label="(Slot $SLOT)"

case "$mode" in
on)
    echo 1 | sudo tee "$LOCATE_PATH" >/dev/null
    echo "LED an${slot_label} via sysfs - $DISK"
    ;;
off)
    echo 0 | sudo tee "$LOCATE_PATH" >/dev/null
    echo "LED aus${slot_label} via sysfs - $DISK"
    ;;
blink)
    echo 1 | sudo tee "$LOCATE_PATH" >/dev/null
    echo "LED an${slot_label} - blinke ${BLINK_SECS}s..."
    sleep "$BLINK_SECS"
    echo 0 | sudo tee "$LOCATE_PATH" >/dev/null
    echo "LED aus."
    ;;
esac
return 0
}

# -----
# SES-Index + Slotnummer + SES-Device ermitteln (per SAS)
# -----
get_ses_index() {
    # Outputs (echo): "<SES_DEV> <INDEX> <SLOTNUM> <HOST> <BUS> <TARGET> <LUN>"
    need lsscsi; need sg_ses

    local LINE
    LINE="$(lsscsi -g | awk -v d="$DISK" '{if ($(NF-1)==d) {print; exit}}')"
    [[ -n "$LINE" ]] || { echo "Konnte $DISK nicht in 'lsscsi -g' finden." >&2; return 1; }

    if [[ "$LINE" =~ ^[0-9]+:[0-9]+:[0-9]+:[0-9]+ ]]; then
        local HOST="${BASH_REMATCH[1]}" BUS="${BASH_REMATCH[2]}" TARGET="${BASH_REMATCH[3]}"
        LUN="${BASH_REMATCH[4]}"
    else
        echo "Konnte SCSI-Adresse nicht parsen." >&2; return 1
    fi
}

```

```

local SES_LINE SES_DEV
SES_LINE="$(ls SCSI -g | awk -v h="$HOST" -v b="$BUS" '$2=="enclosure" && $1 ~ ("\\[\"h\": \"b\":\") {print; exit}')"
[[ -n "$SES_LINE" ]] || { echo "Kein SES/Enclosure auf ${HOST}:${BUS} gefunden (HW-RAID?)." >&2; return 1;
}
SES_DEV="$(echo "$SES_LINE" | awk '{print $NF}')"
[[ -e "$SES_DEV" ]] || { echo "SES-Gerät existiert nicht: $SES_DEV" >&2; return 1; }

local SAS_PATH="/sys/class/scsi_device/${HOST}:${BUS}:${TARGET}:${LUN}/device/sas_address"
[[ -r "$SAS_PATH" ]] || { echo "SAS-Adresse nicht lesbar: $SAS_PATH" >&2; return 1; }
local DISK_SAS; DISK_SAS="$(tr -d '\n' < "$SAS_PATH")"
[[ -n "$DISK_SAS" ]] || { echo "Leere SAS-Adresse." >&2; return 1; }

# In einem Durchlauf: Index + physische Slotnummer zu *dieser* Disk finden
local INDEX SLOTNUM
read INDEX SLOTNUM < <(sg_ses -p aes "$SES_DEV" 2>/dev/null \
| awk -v want="$DISK_SAS" '
/Element index:/ {idx=$3; gsub(":", "", idx); slot=""}
/device slot number:/ {slot=$5}
/SAS address:/ && $3 ~ /^0x/ {sas=$3; if (tolower(sas)==tolower(want)) {print idx, slot; exit}}
')

# Fallback über physische Slotnummer aus sysfs (falls vorhanden)
if [[ -z "${INDEX:-}" || -z "${SLOTNUM:-}" ]]; then
local p="/sys/class/block/${BASE_NOPART}/device"
local enc=( "$p"/enclosure_device* )
if [[ -e "${enc[0]}" && -r "${enc[0]}/slot" ]]; then
local PHY_SLOT; PHY_SLOT="$(cat "${enc[0]}/slot" 2>/dev/null || true)"
if [[ -n "$PHY_SLOT" ]]; then
SLOTNUM="$PHY_SLOT"
INDEX="$(sg_ses -p aes "$SES_DEV" 2>/dev/null \
| awk -v slot="$PHY_SLOT" '
/Element index:/ {idx=$3; gsub(":", "", idx)}
/device slot number:/ {if ($5==slot) {print idx; exit}}
')"
fi
fi
fi

# Es kann vorkommen, dass SLOTNUM leer bleibt → ist ok; dann nur Index nutzen
[[ -n "$INDEX" ]] || { echo "SES-Index konnte nicht ermittelt werden." >&2; return 1; }

```

```

echo "$SES_DEV" "$INDEX" "${SLOTNUM:-}" "$HOST" "$BUS" "$TARGET" "$LUN"
}

# -----
# LED via SES (sg_ses)
# -----
led_ses() {
    local mode="$1" SES_DEV="$2" INDEX="$3" SLOTNUM="${4:-}"
    need sg_ses
    local slot_label=""
    [[ -n "$SLOTNUM" ]] && slot_label=" (Slot $SLOTNUM)"
    case "$mode" in
        on)
            sg_ses --index="$INDEX" --set=ident "$SES_DEV"
            echo "LED an${slot_label}, SES index $INDEX via $SES_DEV - $DISK"
            ;;
        off)
            sg_ses --index="$INDEX" --clear=ident "$SES_DEV"
            echo "LED aus${slot_label}, SES index $INDEX via $SES_DEV - $DISK"
            ;;
        blink)
            sg_ses --index="$INDEX" --set=ident "$SES_DEV"
            echo "LED an${slot_label} - blinke ${BLINK_SECS}s (SES index $INDEX)..."
            sleep "$BLINK_SECS"
            sg_ses --index="$INDEX" --clear=ident "$SES_DEV"
            echo "LED aus."
            ;;
    esac
}

# -----
# Methode wählen & LED einschalten (falls gew.)
# -----
SES_DEV=""; INDEX=""; SLOTNUM=""; HOST=""; BUS=""; TARGET=""; LUN=""
OLDDEV=""

if (( USE_LED )); then
    case "$METHOD" in
        symlink)
            led_symlink "${([[ $BLINK_SECS -gt 0 ]] && echo blink || echo on)" \

```

```

    || { echo "Symlink-LED nicht verfügbar."; exit 1; }
# H:B:T:L & OLDDEV für delete/scan:
need lsscsi
LINE="$(lsscsi -g | awk -v d="$DISK" '{if ($(NF-1)==d) {print; exit}})'"
[[ -n "$LINE" ]] || { echo "Konnte $DISK nicht in 'lsscsi -g' finden."; exit 1; }
OLDDEV="$(awk '{print $(NF-1)}' <<<"$LINE")"
if [[ "$LINE" =~ \[[0-9]+\]:\[[0-9]+\]:\[[0-9]+\]:\[[0-9]+\]\] ]]; then
    HOST="${BASH_REMATCH[1]}"; BUS="${BASH_REMATCH[2]}"; TARGET="${BASH_REMATCH[3]}";
LUN="${BASH_REMATCH[4]}"
else
    echo "Konnte SCSI-Adresse nicht parsen."; exit 1
fi
;;
ses)
read SES_DEV INDEX SLOTNUM HOST BUS TARGET LUN < <(get_ses_index) || exit 1
led_ses "$([[ $BLINK_SECS -gt 0 ]] && echo blink || echo on)" "$SES_DEV" "$INDEX" "$SLOTNUM"
# OLDDEV zusätzlich erfassen (nur Info)
need lsscsi
LINE="$(lsscsi -g | awk -v d="$DISK" '{if ($(NF-1)==d) {print; exit}})'"
[[ -n "$LINE" ]] && OLDDEV="$(awk '{print $(NF-1)}' <<<"$LINE")" || true
;;
auto)
if ! led_symlink "$([[ $BLINK_SECS -gt 0 ]] && echo blink || echo on)"; then
    echo "(Auto) Fallback auf SES..."
    read SES_DEV INDEX SLOTNUM HOST BUS TARGET LUN < <(get_ses_index) || exit 1
    led_ses "$([[ $BLINK_SECS -gt 0 ]] && echo blink || echo on)" "$SES_DEV" "$INDEX" "$SLOTNUM"
else
    # Symlink-Weg erfolgreich: H:B:T:L & OLDDEV besorgen
    need lsscsi
    LINE="$(lsscsi -g | awk -v d="$DISK" '{if ($(NF-1)==d) {print; exit}})'"
    [[ -n "$LINE" ]] || { echo "Konnte $DISK nicht in 'lsscsi -g' finden."; exit 1; }
    OLDDEV="$(awk '{print $(NF-1)}' <<<"$LINE")"
    if [[ "$LINE" =~ \[[0-9]+\]:\[[0-9]+\]:\[[0-9]+\]:\[[0-9]+\]\] ]]; then
        HOST="${BASH_REMATCH[1]}"; BUS="${BASH_REMATCH[2]}"; TARGET="${BASH_REMATCH[3]}";
LUN="${BASH_REMATCH[4]}"
    else
        echo "Konnte SCSI-Adresse nicht parsen."; exit 1
    fi
fi
;;

```

```

*) echo "Unbekannte Methode: $METHOD"; exit 1 ;;
esac
else
# Keine LED: wir brauchen nur H:B:T:L & OLDDEV
need lsscsi
LINE="$(lsscsi -g | awk -v d="$DISK" '{if ($(NF-1)==d) {print; exit}}')"
```

[[-n "\$LINE"]] || { echo "Konnte \$DISK nicht in 'lsscsi -g' finden."; exit 1; }

```

OLDDEV="$(awk '{print $(NF-1)}' <<<"$LINE")"
if [[ "$LINE" =~ \[[0-9]+\]:\[[0-9]+\]:\[[0-9]+\]:\[[0-9]+\]\] ]]; then
    HOST="${BASH_REMATCH[1]}"; BUS="${BASH_REMATCH[2]}"; TARGET="${BASH_REMATCH[3]}";
LUN="${BASH_REMATCH[4]}"
else
    echo "Konnte SCSI-Adresse nicht parsen."; exit 1
fi
fi

# -----
# Deregister
# -----
echo ">>> Entferne $DISK (SCSI ${HOST}:${BUS}:${TARGET}:${LUN})"
echo 1 | sudo tee "/sys/class/scsi_device/${HOST}:${BUS}:${TARGET}:${LUN}/device/delete" >/dev/null || true

# -----
# Prompt / Swap (ohne Slotnummer)
# -----
if (( PROMPT )); then
    read -p ">>> Bitte neue Festplatte einsetzen und ENTER drücken... " _
fi

# -----
# Rescan (gezielt für den Target-Slot)
# -----
echo ">>> Rescane Slot (Target ${TARGET}) über host${HOST}..."
echo "${HOST} ${BUS} ${TARGET}" | sudo tee "/sys/class/scsi_host/host${HOST}/scan" >/dev/null

# -----
# Neues Device finden (by H:B:T:L)
# -----
echo ">>> Warte auf neues Blockdevice im Slot (Target ${TARGET})..."
need lsscsi
```

```

NEWDISK=""
for i in {1..20}; do
  NEWLINE="$(lsscsi -g | awk -v h="$HOST" -v b="$BUS" -v t="$TARGET"
'$1==sprintf("[%d:%d:%d:0]",h,b,t){print; exit}')"
  if [[ -n "$NEWLINE" ]]; then
    NEWDISK="$(awk '{print $(NF-1)}' <<<"$NEWLINE")"
    echo ">>> Neues Device: $NEWDISK ($NEWLINE)"
    break
  fi
  sleep 1
done
if [[ -n "$NEWDISK" ]]; then
  if [[ -n "$OLDDEV" ]]; then
    if [[ "$NEWDISK" == "$OLDDEV" ]]; then
      echo ">>> Hinweis: Laufwerksbuchstabe unverändert geblieben ($NEWDISK) - alles ok."
    else
      echo ">>> Hinweis: Buchstabe wechselte: alt=$OLDDEV neu=$NEWDISK (normal bei Hot-Swap)."
    fi
  fi
else
  echo "Hinweis: Im Zeitfenster kein neues Device sichtbar. Entweder Treiber/Controller langsam, oder der Slot
wurde noch nicht neu gemeldet. Prüfe lsscsi/Logs."
fi

# -----
# LED aus
# -----
if (( USE_LED )); then
  case "$METHOD" in
    symlink)
      led_symlink off || true      # nutzt den gecachten locate-Pfad
      ;;
    ses)
      led_ses off "$SES_DEV" "$INDEX" "$SLOTNUM" || true
      ;;
    auto)
      if [[ -n "${SES_DEV:-}" && -n "${INDEX:-}" ]]; then
        led_ses off "$SES_DEV" "$INDEX" "$SLOTNUM" || true
      else
        led_symlink off || true
      fi
    fi
  fi
fi

```

```
fi
;;
esac
fi

echo ">>> Fertig."
```

Nun noch ausführbar machen

```
chmod +x /usr/local/bin/scsi-hotswap.sh
```

Hilfe/Benutzung:

Usage:

```
scsi-hotswap [--method auto|symlink|ses] [--blink N|--no-led] [--no-prompt] /dev/sdX|sdX|sdXn
```

Optionen:

```
--method M    M = auto (default) | symlink | ses
--blink N     LED vor dem Tausch N Sekunden blinken (statt dauerhaft an)
--no-led      keine LED-Steuerung
--no-prompt   nicht auf ENTER warten (setze voraus, dass neue Platte schon steckt)
```

Beispiele:

```
scsi-hotswap sdi
scsi-hotswap --method symlink --blink 8 sdg8
scsi-hotswap --method ses --no-prompt sdm
```

Ausgabe:

```
root@backupsrv001:~# scsi-hotswap.sh /dev/sdq
>>> Entferne /dev/sdq (SCSI 3:0:17:0)
>>> Bitte neue Festplatte in Slot 17 einsetzen...
Weiter mit ENTER, wenn neue Platte steckt.
>>> Rescanne Slot 17...
>>> Neue Device-Liste:
```