

LINUX

ADVANCED

SVA



AGENDA

- > Motivation
- > Dienste und weitere Ressourcen
- > Logging
- > Speicher
- > Systemüberwachung
- > Gängige Services

LABS

- > [Lab Do1: Dienste verwalten](#)
- > [Lab Do2: Systemumfang konfigurieren](#)
- > [Lab Do3: Units erstellen](#)
- > [Lab Do4: Timer erstellen](#)
- > [Lab Do5: Cronjobs verwenden](#)
- > [Lab Lo1: Protokolle überwachen](#)
- > [Lab Lo2: Remote Logging konfigurieren](#)
- > [Lab STo1: LVM konfigurieren](#)
- > [Lab STo2: Dateisysteme anlegen](#)

LABS

- > [Lab ST03: LVs vergrößern und verkleinern](#)
- > [Lab ST04: LVM-Snapshots benutzen](#)
- > [Lab ST05: VG erweitern](#)
- > [Lab ST06: LUKS konfigurieren](#)
- > [Lab ST07: Software-RAID konfigurieren](#)
- > [Lab ST08: Dateisysteme automatisch einhängen](#)
- > [Lab Mo1: Prozesse kontrollieren](#)
- > [Lab Mo2: Troubleshooting einer Anwendung](#)
- > [Lab So1: Apache installieren](#)

LABS

- > [Lab So2: PHP-Anwendung installieren](#)
- > [Lab So3: MariaDB installieren](#)
- > [Lab So4: MariaDB-Inhalte verwalten](#)
- > [Lab So5: Samba-Server einrichten](#)
- > [Lab So6: Samba-Client einrichten](#)
- > [Lab So7: NFS-Server einrichten](#)
- > [Lab So8: NFS-Client einrichten](#)

// MOTIVATION

MOTVATION

- > Was mit dem System anstellen, nachdem die Linux-Grundlagen erlernt wurden?
- > System-Administration bedeutet oft **kontinuierliches** Lernen
- > Weitere Grundlagen
 - > System personalisieren
 - > Erste Erfahrungen mit gängigen Diensten
 - > Troubleshooting und Monitoring
 - > Storage



// DIENSTE UND WEITERE RESSOURCEN

WIEDERHOLUNG: SYSTEMD

- > Heutige Linux-Distributionen benutzen i.d.R. **Systemd** als **Init-System**
- > **erster** Prozess während des Boots
 - > verwaltet Dienste, Sitzungen und Anmeldungen (**Units**)
 - > kann **nicht** beendet werden

WIEDERHOLUNG: SYSTEMD

- > Heutige Linux-Distributionen benutzen i.d.R. **Systemd** als **Init-System**
- > **erster** Prozess während des Boots
 - > verwaltet Dienste, Sitzungen und Anmeldungen (**Units**)
 - > kann **nicht** beendet werden
- > Systemd wurde **2010** vorgestellt und löst das ältere **SysVinit** ab
 - > bedeutend **schnellerer** Start dank aggressiver Parallelisierung
 - > modernere Code-Basis

WIEDERHOLUNG: SYSTEMD

- > Heutige Linux-Distributionen benutzen i.d.R. **Systemd** als **Init-System**
- > **erster** Prozess während des Boots
 - > verwaltet Dienste, Sitzungen und Anmeldungen (**Units**)
 - > kann **nicht** beendet werden
- > Systemd wurde **2010** vorgestellt und löst das ältere **SysVinit** ab
 - > bedeutend **schnellerer** Start dank aggressiver Parallelisierung
 - > modernere Code-Basis
- > bietet zahlreiche **Zusatzfunktionen**, u.a zentrales Logging (journald)
 - > teilweise starke Kritik in der Linux-Community
 - > Vorwurf, systemd bricht die ursprüngliche UNIX-Philosophie

WIEDERHOLUNG: SYSTEMD

Systemd steuert zahlreiche Hintergrunddienste, z.B. Web-/Datenbankserver, Netzwerk/Firewall, etc.

Kommando	Erklärung
<code>systemctl list-units --type service*</code>	Zeigt alle ausgeführten Dienste
<code>systemctl list-units --type service --all*</code>	Zeigt alle Dienste
<code>systemctl start <name>.service</code>	Startet einen Dienst
<code>systemctl restart <name>.service</code>	Startet einen Dienst neu
<code>systemctl stop <name>.service</code>	Stoppt einen Dienst
<code>systemctl enable [--now] <name>.service</code>	Aktiviert den Start beim Boot [und sofort]
<code>systemctl disable [--now] <name>.service</code>	Deaktiviert den Start beim Boot [und sofort]
<code>systemctl mark <name></code>	Sperrt eine Unit
<code>systemctl unmark <name></code>	Entsperrt eine Unit

* kann auch mit -t / -a abgekürzt werden

LAB Do1

DIENSTE VERWALTEN

SYSTEMUMFANG

- > Der Zusammenschluss verschiedener Units wird `target` genannt
 - > vergleichbar mit **Init-Runleveln**
- > Es gibt verschiedene Targets für **Systemstatus**, z.B.
 - > `basic.target` - Grundsystem
 - > `multi-user.target` - System akzeptiert mehrere eingeloggte User
 - > `network-online.target` - Netzwerkstack verfügbar
 - > `graphical.target` - Grafischer Desktop gestartet
- > Targets können sich gegenseitig referenzieren 😱

SYSTEMUMFANG

Das aktuelle Target kann wie folgt eingesehen werden:

```
# systemctl get-default  
multi-user.target
```

Es kann auch temporär geändert werden:

```
# systemctl isolate graphical.target
```

Eine dauerhafte Änderung:

```
# systemctl set-default graphical.target
```

SYSTEMUMFANG

Die Abhängigkeiten eines Target können aufgelistet werden:

```
# systemctl list-dependencies multi-user.target
multi-user.target
● └─auditd.service
● └─chronyrd.service
● └─crond.service
● └─firewalld.service
● └─httpd.service
● └─irqbalance.service
....
```

SYSTEMUMFANG

- > Systemd parallelisiert Units radikal beim Starten
 - > Server und Desktops booten so bedeutend schneller
- > Die Startzeiten werden erfasst und können eingesehen werden
 - > so lassen sich auffällige Dienste schnell identifizieren

```
22.008s doge-database.service
 7.748s dnf-makecache.service
 2.101s dev-vdb.device
 2.034s sys-subsystem-net-devices-eth1.device
 2.006s dev-ttyS2.device
```

Hier benötigt ein Dienst **22 Sekunden** zum Starten.

LAB Do2

SYSTEMUMFANG KONFIGURIEREN

WEITERE SYSTEMD-RESSOURCEN

- > Systemd unterstützt neben Diensten (.service) noch weitere Ressourcen:
 - > mount - Einhängepunkte (Mountpoint)
 - > path - Überwachen von Pfaden auf Veränderungen
 - > timer
 - > Regelmäßig ausgeführte Units
 - > Alternative zu Cron
 - > target
 - > Ressourcen für einen bestimmten Systemzustand
 - > Mit Init-Runleveln vergleichbar

WEITERE SYSTEMD-RESSOURCEN

- > Systemd-Units liegen unterhalb /usr/lib/systemd/system
 - > werden zusammen mit dem Paket installiert
 - > **nicht** verändern, werden durch Updates überschrieben
- > Anpassungen finden im Ordner /etc/systemd/system statt
 - > Eine Unit kann dorthin kopiert und editiert werden
 - > höhere **Priorität**

WEITERE SYSTEMD-RESSOURCEN

- > Systemd-Units liegen unterhalb /usr/lib/systemd/system
 - > werden zusammen mit dem Paket installiert
 - > **nicht** verändern, werden durch Updates überschrieben
- > Anpassungen finden im Ordner /etc/systemd/system statt
 - > Eine Unit kann dorthin kopiert und editiert werden
 - > höhere **Priorität**
- > Konfiguration im **INI**-Syntax
 - > einfachere Bearbeitung mit Texteditor
 - > Schlagworte beeinflussen Verhalten

WEITERE SYSTEMD-RESSOURCEN

Das `systemd-delta`-Kommando zeigt erweiterte/ersetzte Units an:

```
# systemd-delta
[EQUIVALENT] /etc/systemd/system/ctrl-alt-del.target →
/usr/lib/systemd/system/ctrl-alt-del.target
[REDIRECTED] /etc/systemd/system/default.target →
/usr/lib/systemd/system/default.target
[EXTENDED]   /usr/lib/systemd/system/httpd.service →
/usr/lib/systemd/system/httpd.service.d/php-fpm.conf

3 overridden configuration files found.
```

systemd kann erkannte Units für Debugging ausgeben:

```
# systemctl cat boot.mount
```

SERVICE-UNIT

[Unit]

```
Description=The Apache HTTP Server
Wants=httpd-init.service
After=network.target remote-fs.target nss-lookup.target httpd-init.service
```

[Service]

```
Type=notify
ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful
KillSignal=SIGHUP
KillMode=mixed
PrivateTmp=true
OOMPolicy=continue
```

[Install]

```
WantedBy=multi-user.target
```

UNIT-SEKTION

Die [Unit]-Sektion definiert Allgemeines und Abhängigkeiten:

Direktive	Erklärung
Description	Beschreibung
Documentation	URL zur Dokumentation
After	Reihenfolge: welche Units müssen vorher gestartet werden?
Requires	Womit soll diese Unit zusammen gestartet werden?
Wants	Weiche Abhängigkeit als Requires - wird auch gestartet, wenn diese nicht zur Verfügung stehen
Conflicts	Negative Abhängigkeit: wird nicht gestartet wenn diese gestartet sind

SERVICE-SEKTION

Die [Service] -Sektion regelt das Prozessverhalten:

Direktive	Erklärung
Type	Startverhalten des Prozesses
ExecStart	Befehl zum Starten des Dienstes
ExecStartPre, ExecStartPost	Optionale Pre-/Post-Skripte
ExecStop	Befehl zum Stoppen des Dienstes
ExecReload	Befehl zum Neuladen des Dienstes
KillSignal	Prozess-Signal zum Beenden des Dienstes
KillMode	Verhalten beim Beenden des Prozesses

SERVICE-SEKTION: TYPEN

Je nachdem wie sich eine Anwendung kommt einer der folgenden Typen infrage:

Typ	Erklärung
simple	Standard: Angegebener Prozess einzige Hauptprozess
forking	Der Prozess startet Kindprozesse und beendet den Elternprozess
oneshot	Wie simple, jedoch wird der Prozess rasch beendet
dbus	Wie simple, jedoch werden weitere Prozess nur gestartet, wenn D-Bus-Kommunikation stattfindet
idle	Wie simple, jedoch wird die Ausführung verzögert, bis alle Jobs abgeschlossen sind

INSTALL-SEKTION

Die [Install]-Sektion definiert, wie die Unit gestartet wird:

Direktive	Erklärung
Alias	Alternative Namen
RequiredBy	Liste von Units, die diese Unit benötigen
WantedBy	Weiche Abhängigkeit: Liste von Units, die nach dieser Unit gestartet werden möchten

MOUNT-UNIT

Hängt Geräte ein:

[Unit]

Description=Mount storage

[Mount]

What=/dev/disk/by-uuid/EC42-1337

Where=/mnt/usb

Type=ext4

Options=rw

[Install]

WantedBy=multi-user.target

MOUNT-UNIT

Direktive	Erklärung
What	Pfad zur Gerätedatei
Where	Mountpoint, in welchem das Gerät eingehängt werden soll
Type	Dateisystem-Typ, z.B. ext 4
Options	Dateisystem-Optionen, z.B. <code>ro</code> (read-only)

Moderne Linux-Distributionen generieren automatisch entsprechende Mount-Units gemäß der Einträge der /etc/fstab:

```
systemctl status boot-efi.mount
● boot-efi.mount - /boot/efi
  Loaded: loaded (/etc/fstab; generated)
  Active: active (mounted) since Thu 2025-02-13 08:04:12 CET; 10h ago
```

MOUNT-UNIT

- > Dateiname der Unit muss dem Mountpoint entsprechen 
- > Unit, die nach /mnt mountet, muss mnt .mount heißen
- > Sonderzeichen müssen escaped werden
- > Dateiname kann mit `systemd-escape` berechnet werden:

```
$ systemd-escape -p --suffix=mount "/mnt/nas/katzen-bilder"  
mnt-nas-katzen\x2dbilder.mount
```

UUIDs bei externen Speichern empfohlen:

```
# blkid  
/dev/sda4: UUID="f7b49664-5dc5-4e22-2325-b69d5be1337d" TYPE="ext4" ↵  
LABEL="Hannah Montana Linux"
```

PATH-UNIT

Diese Unit kann auf Veränderungen in Pfaden reagieren und eine weitere Unit auslösen:

[Unit]

```
Description="Monitor /etc/passwd ←  
for changes"
```

[Path]

```
PathModified=/etc/passwd  
Unit=passwd-monitor.service
```

[Install]

```
WantedBy=multi-user.target
```

[Unit]

```
Description="Send email alert"
```

[Service]

```
ExecStart=/usr/local/bin/passwd-alert.sh
```

```
#!/bin/bash  
mail -S sendwait -s "/etc/passwd was changed on $(hostname)" ←  
admin@evilcorp.lan < /etc/passwd
```

TIMER-UNIT

- > Unit, die Dienste in regelmäßigen Abständen auslöst
 - > z.B. temporäre Dateien aufräumen oder PHP-Cache leeren
- > wird gerne als Alternative zu Cron genutzt

Die letzten und nächsten Läufe lassen sich anzeigen:

```
# systemctl list-timers
```

NEXT	LEFT	LAST	PASSED
Thu 2025-02-13 17:50:00 UTC	6min left	Thu 2025-02-13 17:40:05 UTC	3min 23s ago
Thu 2025-02-13 18:09:00 UTC	25min left	Thu 2025-02-13 17:39:01 UTC	4min 27s ago

TIMER-UNIT

[Unit]

```
Description=dnf makecache --timer  
Wants=network-online.target
```

[Timer]

```
OnBootSec=10min  
OnUnitInactiveSec=1h  
RandomizedDelaySec=60m  
Unit=dnf-makecache.service
```

[Install]

```
WantedBy=timers.target
```

Stündliches aktualisieren des DNF-Paketcaches, jedoch nicht früher als 10 Minuten nach dem Booten. Erfordert, dass Netzwerk gestartet wurde.

TIMER-UNIT

Direktive	Erklärung
OnActiveSec	Start relativ zum Startpunkt der Timer-Unit
OnBootSec	Start relativ zur Bootzeit
OnStartupSec	Start relativ zum Systemd-Start
OnUnitActiveSec	Start relativ zum letzten Start der betroffenen Unit
OnUnitInactiveSec	Start relativ zum letzten Stop der betroffenen Unit
OnCalendar	Konkrete Zeitangabe

TIMER-UNIT: ZEITANGABEN

Direktive	Erklärung
Wed 15:00:00	mittwochs um 15:00
Mon..Wed *-7-9	immer ab 09.07 wenn der Tag ein Montag, Dienstag oder Mittwoch ist
2025-12-31	Um 00:00 am 31.12.2025
12/2	alle 2 Stunden ab 12:00
hourly	stündlich
daily, weekly, monthly	täglich, wöchentlich, monatlich um 00:00

LAB Do3

UNITS ERSTELLEN

LAB Do4

TIMER VERWALTEN

CRONJOBS

- > automatische Erledigung **wiederkehrender** Aufgaben
- > entweder zu bestimmten Zeiten oder Ereignissen
 - > z.B. täglich um 02:20 oder irgendwann täglich
- > moderne Linux-Systeme haben **2 Cron-Dienste**
 - > cron
 - > "klassischer" Cron (i.d.R. *Vixie Cron*) für Jobs zu festen Uhrzeiten
 - > anacron
 - > entscheidet nach vergangener Zeitspanne der letzten Ausführung
 - > u.a. für Desktops entwickelt

CRONJOBS

- > Dienst cron bzw. crond startet beide Komponenten
- > Diese haben abweichende **Konfigurationsdateien**
 - > crond: /etc/crontab
 - > anacron: /etc/anacrontab

CRONJOBS

- > Dienst cron bzw. crond startet beide Komponenten
- > Diese haben abweichende **Konfigurationsdateien**
 - > crond: /etc/crontab
 - > anacron: /etc/anacrontab
- > Insbesondere der Konfigurationssyntax von crond gilt als gewöhnungsbedürftig
- > anacron hat hier mit sprechenderen Namen die geringere Einstiegshürde

CRONJOBS: CROND

Systemweite Konfiguration (/etc/crontab):

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .---- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .--- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
```

Neben einigen Umgebungsvariablen werden vor allem Zeitangaben definiert.

CRONJOBS: CROND

Ausführen eines Programms an jedem 9.Juli um 23:52:

```
52 23 09 07 * root /bin/fork --type cstan
```

Erstellen eines Backups täglich um 12 und 18 Uhr:

```
00 12,18 * * * operator /opt/bin/online-backup
```

Skriptausführung täglich zu jeder vollen Arbeitsstunde (8 bis 18):

```
00 08-18 * * * sgiertz /bin/work-simulator /dev/mouse
```

Überwachung alle 10 Minuten

```
*/10 * * * * operator /opt/bin/check_hdd_temp
```

CRONJOBS: CROND

Später wurden einige sprechendere Direktiven implementiert, um die Konfiguration zu erleichtern:

Abkürzung	Ausführung	Erklärung
@reboot	-	Wird einmal beim Start ausgeführt
@hourly	0 * * * *	zu jeder vollen Stunde
@daily	0 0 * * *	täglich um 0:00
@weekly	0 0 * * 0	Zu jedem ersten Tag einer Woche um 0:00
@monthly	0 0 1 * *	Zu jedem ersten Tag eines Monats um 0:00
@yearly	0 0 1 1 *	Zu jedem ersten Tag eines Jahres um 0:00

CRONJOBS: CROND

- > neben systemweiten Konfigurationen können User auch eigene pflegen
 - > Feld 6 für den Username entfällt
- > diese werden unterhalb /var/spool/cron/crontabs/<name> gespeichert
- > Befehle
 - > crontab -l
 - > listet die eigene Crontab auf
 - > crontab -e
 - > editiert die Crontab
 - > crontab -r
 - > entfernt die Konfiguration

CRONJOBS: CROND

```
$ crontab -l  
# m h dom mon dow command  
0 * * * * /home/doge/hourly-meme-report.sh
```

```
$ crontab -r
```

- > beim Editieren mit `crontab -e` wird der bevorzugte Kommandozeilen-Editor gestartet
- > Konfiguration wird anschließend aktiviert
- > beim Anpassen der systemweiten Konfiguration muss die Konfiguration des Dienstes neu eingelesen werden

LAB Do5

CRONJOBS VERWENDEN

ZUSAMMENFASSUNG

- > **Systemd** ist als Init-System der erste Prozess während des Boots
 - > verwaltet u.a. Dienste und Sitzungen
 - > aggressive **Parallelisierung**
- > bietet zahlreiche Zusatzfunktionen, wie z.B. **Logging**
- > Verwaltung über das `systemctl`-Kommando
- > Neben Diensten werden auch verwaltet
 - > Mountpoints
 - > Veränderungen in Pfaden
 - > Regelmäßige auszuführende Units

ZUSAMMENFASSUNG

- > Cronjobs waren früher der einzige Weg regelmäßig Aufgaben auszuführen
 - > moderne Systeme nutzen sowohl **Vixie-Cron** als auch anacron
- > Durch reine Textkonfigurationen werden Aufgaben geplant
- > neben systemweiten Konfiguration können auch User eigene Jobs einplanen

// LOGGING

LOGS

- > Bei Fehlern schreiben Anwendungen **Protokolldateien**
 - > diese können dabei helfen, die **Fehlerursache** herauszufinden

LOGS

- > Bei Fehlern schreiben Anwendungen **Protokolldateien**
 - > diese können dabei helfen, die **Fehlerursache** herauszufinden
- > Der **Ort** des Protokolls kann **variieren**
 - > in einer Datei unterhalb /var/log
 - > in das Systemd-Journal (journalctl)
 - > in das zentrale Systemlogging rsyslog
- > Kernelmeldungen finden sich mit dmmsg

LOGS

Unterhalb von /var/log finden sich zahlreiche Logdateien:

```
$ ls -1 /var/log
cron
dnf.log
messages
secure
spooler
...
```

Inzwischen loggen die meisten Dienste in das **Systemd-Journal**

Alle Logs:

```
# journalctl
```

Spezifisches Log:

```
# systemctl list-units
# journalctl -u rsyslog.service
...
Apr 08 14:03:37 almalinux-00.lab.sva.de
Apr 08 14:03:37 almalinux-00.lab.sva.de
...
```

LOGS

dmesg zeigt Boot- und Kernel-Meldungen:

```
[    0.000000] Linux version 6.7.11-100.fc38.x86_64 (mockbuild@b3ea476685e349729)
[    0.000000] Command line: BOOT_IMAGE=(hd1,gpt2) /vmlinuz-6.7.11-100.fc38.x86_64
...
[  41.879175] wlp2s0: authenticate with xx:xx:xx:xx:xx:xx (local address=yy:yy:yy)
[  41.879187] wlp2s0: send auth to xx:xx:xx:xx:xx:xx (try 1/3)
[  41.884304] wlp2s0: authenticated
[  41.885567] wlp2s0: associate with xx:xx:xx:xx:xx:xx (try 1/3)
```

LOGS

Falls aktiviert, bleibt das Journal auch über mehrere Boots erhalten:

```
# journalctl --list-boots
```

IDX	BOOT ID	FIRST ENTRY	LAST ENTRY
-2	8e9e418f73c747f1979144868807fe19	Wed 2025-02-12 09:29:48 CET	Wed 2025-02-12 2
-1	c5bea08823ba47aea4f49755a6dc2e87	Thu 2025-02-13 09:03:57 CET	Thu 2025-02-13 2
0	2dbc74c86c5a41cdb5fb2a7894832fcc	Fri 2025-02-14 09:41:12 CET	Fri 2025-02-14 2

0 ist immer der Standard, ältere Einträge können mit -b oder --boot angegeben werden:

```
# journalctl -b -1 -u myapp.service
```

LAB Lo1

PROTOKOLLE ÜBERWACHEN

SYSTEMD–JOURNAL

- > systemd-Komponente die für das Logging zuständig ist
- > kümmert sich auch um **Log Rotation**
- > hat ein eigenes **Binärformat**
 - > schnell, bietet Schutz gegen Manipulation bei einem Angriff

SYSTEMD–JOURNAL

- > systemd-Komponente die für das Logging zuständig ist
- > kümmert sich auch um **Log Rotation**
- > hat ein eigenes **Binärformat**
 - > schnell, bietet Schutz gegen Manipulation bei einem Angriff
- > kennt zwei **Modi**
 - > **persistente** Speicherung
 - > Log unterhalb /var/log/journal, auf der Festplatte gespeichert
 - > **temporäre** Speicherung
 - > Log unterhalb /run/log/journal
 - > wird spätestens beim Reboot entfernt

SYSTEMD–JOURNAL

- > Konfigurationsdatei: /etc/systemd/journald.conf
- > für eigene abweichende Einstellungen kann auch der Ordner /etc/systemd/journald.conf.d angelegt werden
- > INI-Format

SYSTEMD-JOURNAL

Beispielhafte Konfiguration:

[Journal]

Storage=auto

Compress=yes

Seal=yes

SyncIntervalSec=5m

RateLimitIntervalSec=30s

RateLimitBurst=10000

- > Storage: Speicherart
 - > auto - persistent wenn /var/log/journal existiert, ansonsten volatile
- > Compress - komprimieren
- > Seal - Verschlüsselung aktivieren
- > Limitierung in Sekunden oder generelles Limit

RSYSLOG

- > Open Source-Implementation des syslog-Protokolls
- > Standard zahlreicher Linux-Distributionen
- > kann Nachrichten **filtern** und auf verschiedene Protokolldateien verteilen
- > **ISO-8601**-Zeitstempel; sehr genau und kann Zeitzonen unterscheiden
 - > z.B. 1990-07-09T23:52:00.001+01:00
 - > (Eine Millisekunde nach 09.07.1990 23:52)
- > Übertragung über **UDP** oder **TCP** an Remote Server
- > Unterstützung von TLS und neueren Logformaten

```
<13>Jul 9 23:52:42 mymachine sgiertz: This is a test message
```

EXKURS: SYSLOG

- > **S**ystem **L**ogging Protocol
- > Standard für Log-Meldungen zwischen Servern
- > simpel, erstmalig in den 1980ern implementiert ([RFC 3164](#))
- > **Priorität** besteht aus **Facility** und **Severity**
 - > Facility gibt betroffene Komponente an
 - > Severity definiert, wie schwerwiegend eine Nachricht ist
- > **Header** enthält Zeitstempel sowie Informationen zum Absender
 - > Zeitstempel wird vom Empfänger eingefügt

EXKURS: SYSLOG - FACILITIES

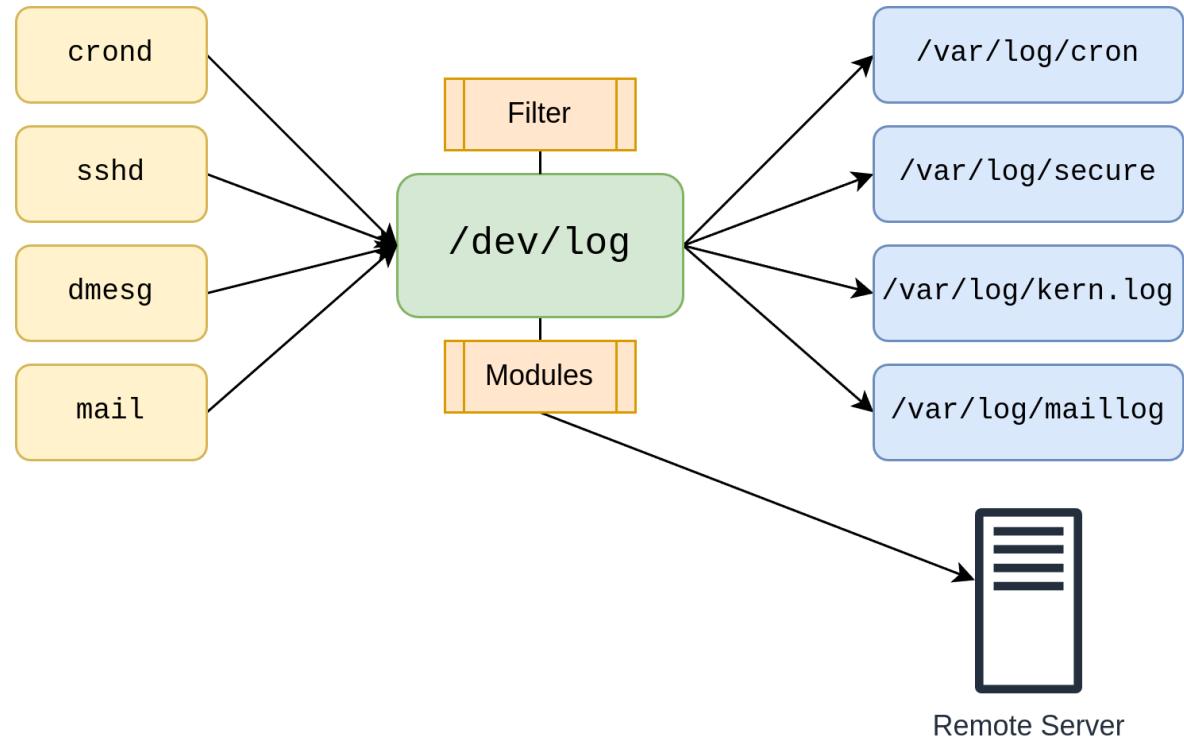
Facility	Nummer	Beschreibung
kern	0	Kernel-Meldungen
user	1	User-Meldungen
mail	2	Mailsystem
system	3	System-Meldungen und -Fehler
auth, authpriv	4, 10	Accounting
news	7	Nachrichten-System
uucp	8	U inx to U inx c opy, heute Spooler-Informationen
clock, ntp	9 und 15, 12	Uhrzeit und -Synchronisation
ftp	11	Dateiübertragung
local0 bis local7	16 bis 23	lokale, eigene Anwendungen

EXKURS: SYSLOG - SEVERITIES

Severity	Nummer	Beschreibung
emerg	0	Schwerwiegender Notfall
alert	1	Schwerer Fehler
crit	2	Kritischer Zustand
error	3	Fehler
warn	4	Warnung
notice	5	Hinweis
info	6	Optionale Information
debug	7	Debugging-Information, i.d.R. für Troubleshooting relevant

RSYSLOG

- > Hauptkonfiguration:
`/etc/rsyslog.conf`
 - > Verzeichnis `/etc/rsyslog.d` kann weitere Einstellungen enthalten
- > steuert Verhalten
 - > Standard-Berechtigungen angelegter Dateien
 - > Filter und Protokolldateien
- > Erweiterung über Module
 - > z.B. für Remote Logging



RSYSLOG

Auszug aus einer Standard-Konfiguration:

```
*.*;auth,authpriv.none          -/var/log/syslog
auth,authpriv.*                  /var/log/auth.log
cron.*                          /var/log/cron.log
kern.*                          -/var/log/kern.log
lpr.*                           -/var/log/lpr.log
mail.*                          -/var/log/mail.log
mail.err                         /var/log/mail.err
```

Alle nicht näher klassifizierten Nachrichten werden nach `/var/log/syslog` geschrieben.

Logdateien mit vorangestelltem `-` werden nicht nach jeder Änderungen sondern in periodischen Abständen geschrieben.

REMOTE LOGGING

- > systemd-journald kann Nachrichten an rsyslog weiterleiten
 - > z.B. um Logs an einen zentralen Server zu senden
 - > in großen Umgebungen sinnvoll um den Überblick zu behalten
- > rsyslog kann über Module Nachrichten empfangen
 - > imtcp (*TCP-Port 514*)
 - > umudp (*UDP-Port 514*)
- > Neben rsyslog gibt es ganze Produkte, die sich dem Thema widmen
 - > u.a. [Graylog](#), [Splunk](#), [Elasticsearch](#)
 - > diese bieten meist intelligente Suchen und Alarmfunktionen

REMOTE LOGGING: EINRICHTUNG

- > Server-Konfiguration anpassen
 - > gewünschtes Modul (TCP, UDP) aktivieren
 - > Port in der Firewall öffnen
 - > Konfiguration validieren und Dienst neustarten
- > Client-Konfiguration
 - > Facilities und Serverities weiterleiten
 - > Konfiguration validieren und Dienst neustarten

LAB Lo2

REMOTE LOGGING KONFIGURIEREN

ZUSAMMENFASSUNG

- > Anwendungen schreiben i.d.R. **Protokolldateien**
 - > diese können im **Dateisystem** oder im zentralen **Journal** stehen
- > rsyslog implementiert das syslog-Protokoll
 - > es kann Nachrichten filtern und auf Dateien verteilen
 - > auch eine Weiterleitung an externe Systeme ist möglich

// STORAGE

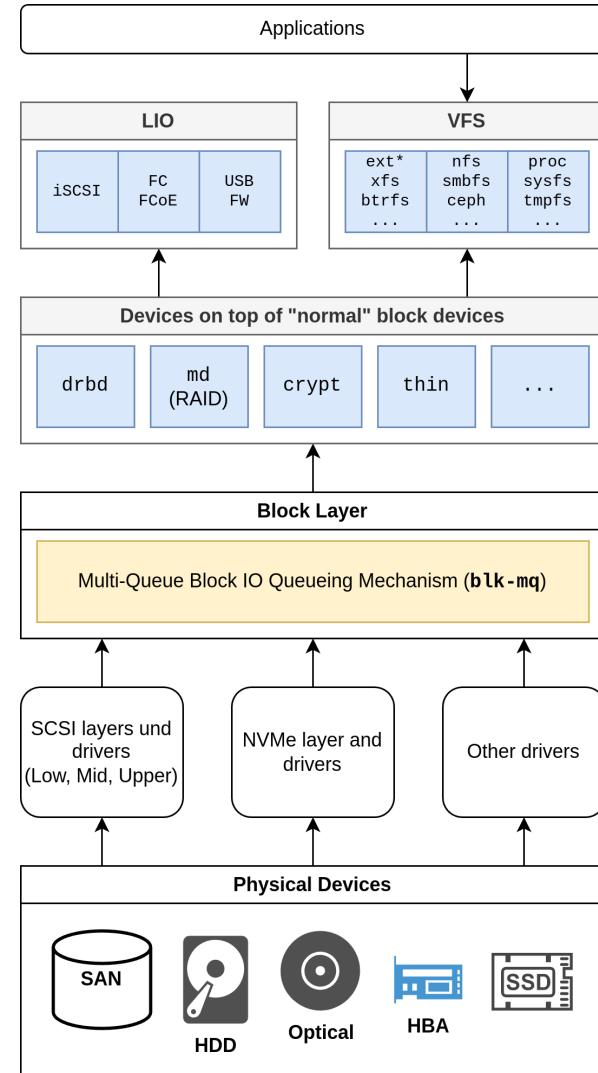
STORAGE

- > Linux unterstützt zahlreiche verschiedene Storage-Protokolle und -Medien
- > IO-Scheduler erlaubt starke **Skalierung**
 - > je nach Hardware und Einsatzzweck
- > zahlreiche Dateisysteme werden unterstützt
- > Neben Software-**RAIDs** sind auch dynamische **LVM**-Partitionierungen möglich
- > Hardware-beschleunigte Verschlüsselung mit **LUKS**

STACK

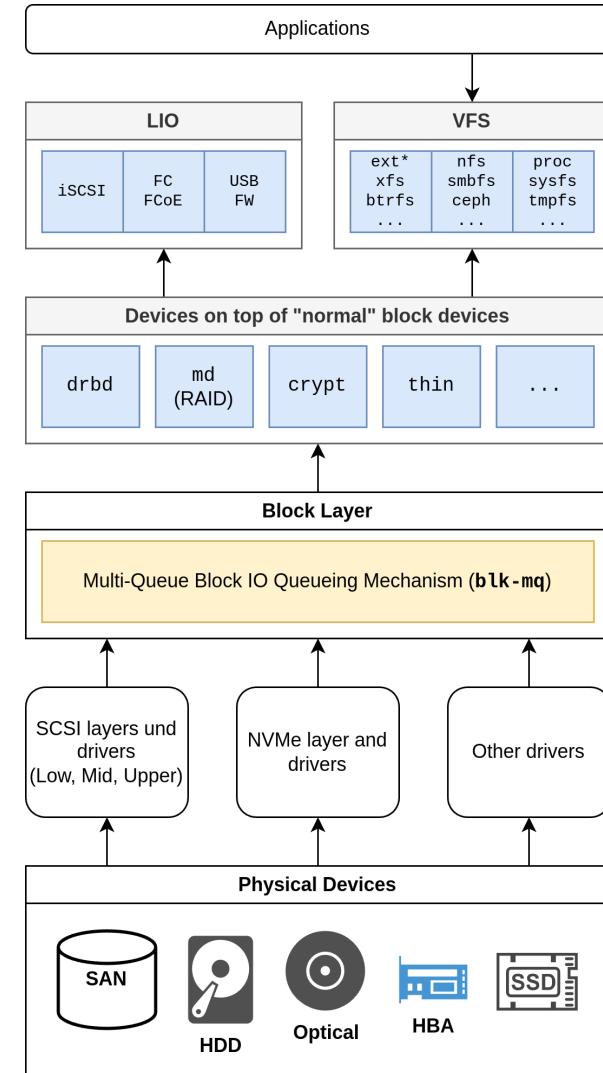
Involvierte Komponenten bei I/O durch Anwendungen:

- > **VFS** (Virtual File System)
 - > block-basierte Dateisysteme: ext*, btrfs, xfs,...
 - > Netzwerkdateisysteme: Samba, NFS, CEPH,...
 - > **FUSE** (Userspace-Dateisysteme)
 - > Pseudo-Dateisysteme: sysfs, proc,...
 - > Spezielle Dateisysteme: tmpfs,...
- > **LIO** (LinuxIO)
 - > iSCSI über verschiedene Protokolle und Geräte



STACK

- > Optionaler **Device Mapper**
- > Block Layer
 - > IO-Scheduler für verwendete Speicher
 - > seit Linux 5.0 blk-mq
 - > muss vom Treiber unterstützt werden
- > Verschiedene Schichten und Treiber je nach Protokoll
 - > SCSI, NVMe,...
- > **Physische Medien**
 - > SAN, HDD, SSD,...



IO-SCHEDULER

- > IO-Scheduler ist für die Planung und Ausführung der IO-Anfragen zuständig
- > Scheduling des Linux-Kernels ist anpassbar
 - > i.d.R. aber nicht notwendig

IO-SCHEDULER

- > IO-Scheduler ist für die Planung und Ausführung der IO-Anfragen zuständig
- > Scheduling des Linux-Kernels ist anpassbar
 - > i.d.R. aber nicht notwendig
- > mq-deadline
 - > **Standard**, für physische und virtuelle Disks
 - > garantiert faire Latenz, Leseanfragen werden bevorzugt
- > none
 - > einfacher **FIFO**-Algorithmus, **kann** bei NVMe und sehr hohen IOPS von Vorteil sein

IO-SCHEDULER

- > kyber
 - > für sehr schnelle Geräte konzipiert
 - > für geringe Latenz, kann für lesen/schreiben optimiert werden
- > bfq
 - > für Desktop-Systeme optimiert
 - > sorgt dafür, dass eine Anwendung nicht die gesamte Bandbreite nutzt
 - > Fokus liegt auf geringer Latenz statt auf maximalem Durchsatz

IO-SCHEDULER

Vom Kernel aktivierte IO-Scheduler auflisten:

```
# dmesg | grep -i 'io scheduler'  
[ 0.500980] io scheduler mq-deadline registered  
[ 0.500982] io scheduler kyber registered  
[ 0.501031] io scheduler bfq registered
```

Verwendeten IO-Scheduler eines Geräts anzeigen:

```
# cat /sys/block/sda/queue/scheduler  
[mq-deadline] kyber bfq none
```

IO-SCHEDULER

Temporäres Ändern des IO-Schedulers eines Geräts:

```
# echo 'none' > /sys/block/sda/queue/scheduler  
# cat /sys/block/sda/queue/scheduler  
mq-deadline kyber bfq [none]
```

Die Änderung kann über [TuneD](#) oder [udev](#) persistiert werden.

DEVICE MAPPER

- > **Kernel-Treiber** zur Datenspeicher-Verwaltung
- > Generischer Weg zur Ansteuerung verschiedener Speichertypen
- > erstellt **virtuelle** Geräte und überträgt logische Sektoren in physische

DEVICE MAPPER

- > **Kernel-Treiber** zur Datenspeicher-Verwaltung
- > Generischer Weg zur Ansteuerung verschiedener Speichertypen
- > erstellt **virtuelle** Geräte und überträgt logische Sektoren in physische
- > Implementiert weitere Funktionen
 - > lvm - dynamische Partitionierung
 - > multipath - Speicherzugriff über mehrere Wege (SAN)
 - > dm-raid - Software-RAID
 - > dm-crypt - LUKS-verschlüsselter Speicher
- > Zuweisungstabelle enthält Blockadressen/-längen und Parameter

DEVICE MAPPER

Verwaltete Geräte auflisten:

```
# dmsetup ls  
vg_data-lv_data (253:0)
```

Das Gerät /dev/mapper/vg_data-lv_data hat die SCSI-ID 253:0.

Zuweisungstabelle anzeigen:

```
# dmsetup table  
vg_data-lv_data: 0 2097152 linear 252:16 2048
```

Das Gerät erstreckt sich von Block 0 bis 2097152. Es handelt sich um ein linear LVM-Volume mit einem Metadaten-bedingten Offset von 2048 Byte.

DATEISYSTEME

Linux kann auf einer **breite Fülle** an Dateisystemen **installiert** werden:

- > ext-Familie
 - > ext2 - **ohne** Journal, seit 1993
 - > ext3/4 - **mit** Journal, seit 2001/2008
- > XFS
 - > stammt von Silicon Graphics (IRIX), seit 2001 für Linux
- > Btrfs, seit 2013
- > ZFS
 - > lizenzrechtlich nicht nativ im Linux-Kernel
 - > seit 2013 als **OpenZFS on Linux** für Linux

DATEISYSTEME

Darüber hinaus kann Linux weitere Dateisysteme **lesen und schreiben**:

- > FAT-Familie, **File Allocation Table**
 - > FAT12/16/32 und exFAT (*MS-DOS und Flash-Speicherkarten*)
- > JFS, **Journaled File System** (*IBM AIX und OS/2*)
- > NTFS, **New Technology File System** (*Microsoft Windows NT*)
- > HFS und HFS+ (*Apple*)
- > UFS, **Unix File System**
- > ISO 9660 (*CD*) und UDF (*DVD*)

DATEISYSTEME

- > Red Hat-artige Distributionen setzen i.d.R. auf **XFS**
 - > SUSE-artige Distributionen setzen standardmäßig auf **Btrfs***
 - > allerdings werden nicht alle Features unterstützt, z.B. **RAID** und **Komprimierung**
 - > Bei Debian ist XFS und ext4 üblich
 - > Ubuntu bietet zusätzlich optional **ZFS** an
- * nur für das Betriebssystem; **nicht** für Anwendungsdaten, wie z.B. Datenbanken

DATEISYSTEME

- > **Journaling**-Dateisysteme schreiben Änderungen vor dem eigentlichen Schreiben in einen **Zwischenspeicher**
 - > erschwert Datenverlust, z.B. bei Stromausfällen
 - > gerade bei großen Partitionsgrößen sinnvoll
 - > seit Mitte der 1990er Standard gängiger Dateisysteme

DATEISYSTEME

- > **Journaling**-Dateisysteme schreiben Änderungen vor dem eigentlichen Schreiben in einen **Zwischenspeicher**
 - > erschwert Datenverlust, z.B. bei Stromausfällen
 - > gerade bei großen Partitionsgrößen sinnvoll
 - > seit Mitte der 1990er Standard gängiger Dateisysteme
- > **Btrfs** vereint **weitere Funktionen** in einem Dateisystem
 - > seit 2013 im Linux-Kernel, jedoch stagnierende Entwicklung
 - > **Snapshots** und Prüfsummen
 - > Subvolumes und Prüfsummen
 - > Datenkompression und **RAID**

DATEISYSTEME

- > **ZFS**
 - > leistungsfähiges und zukunftssicheres Dateisystem von **SUN**, heute **Oracle**
 - > bietet **RAID** und **Prüfsummen** um Datenverluste zu vermeiden
 - > aufgrund 128-bit Design nahezu unerschöpfliche Größe (16 Exabyte*)

DATEISYSTEME

- > **ZFS**
 - > leistungsfähiges und zukunftssicheres Dateisystem von **SUN**, heute **Oracle**
 - > bietet **RAID** und **Prüfsummen** um Datenverluste zu vermeiden
 - > aufgrund 128-bit Design nahezu unerschöpfliche Größe (16 Exabyte*)
 - > diente als Vorlage für **Btrfs**
 - > unter der **CDDL** lizenziert, aber leider nicht mit der GPL kompatibel
 - > daher nicht im Linux-Kernel enthalten, aber als **3rd Party-Paket** installierbar

* = ~999.999 TB

DATEISYSTEME

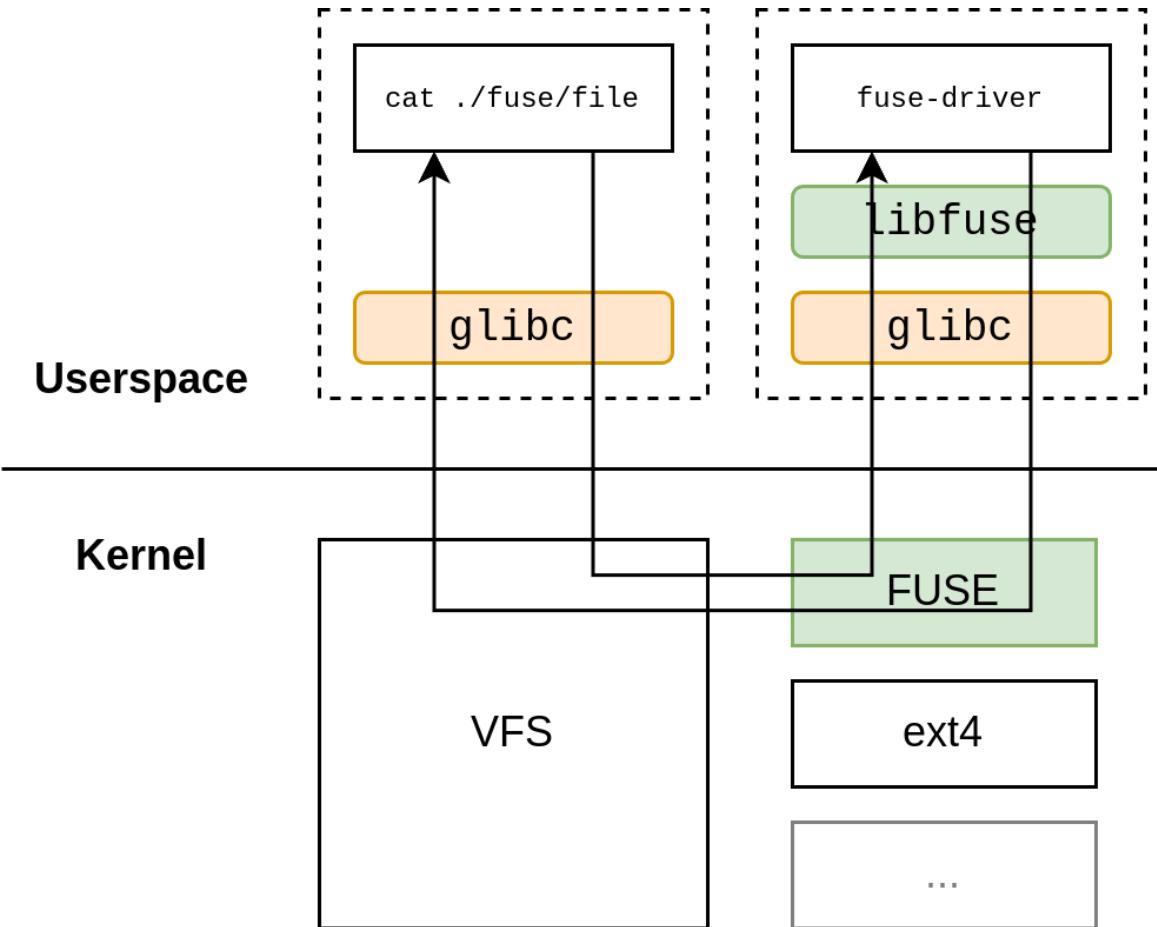
- > Btrfs und ZFS sind **Copy-on-write**-Dateisysteme (*CoW*)
 - > Ansatz zur Optimierung von Schreibzugriffen
 - > wenn eine Datei mehrfach angelegt wird, wird sie nur einmal physisch angelegt
 - > erst wenn eine der Kopien **geändert** wird, wird die Änderung physisch **gespeichert**
 - > Einsparung unnötiger Kopien (**Deduplikation**)

EXKURS: INODES

- > essentielles Element zur Datenverwaltung in unixartigen Dateisystemen
- > eindeutig identifizierbar (**Inode-Nummer**)
- > enthalten **Metadaten** einer Datei
 - > Dateiart (*Datei, Verzeichnis, symbolischer Link, Gerätedatei, Socket*)
 - > Größe
 - > Berechtigungen sowie Owner/Group
 - > Zugriffszeitstempel
 - > Verweis auf die tatsächlichen Inhalte
- > Jeder Namenseintrag verweist auf **exakt** einen Inode

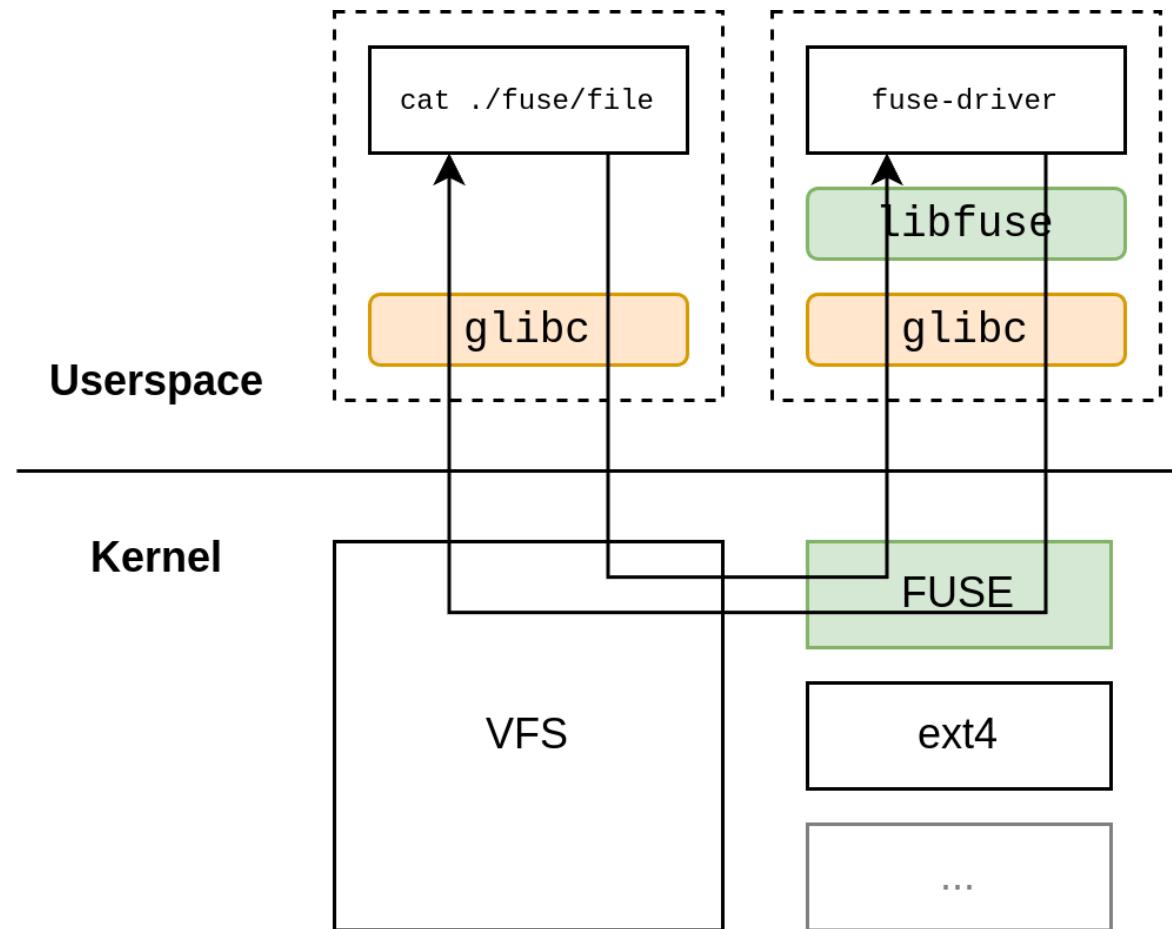
FUSE

- > Filesystem in Userspace
- > Verlagerung von Dateisystem-Treibern vom Kernel- in den Usermodus
 - > so können auch unprivilegierte User Dateisysteme einhängen
 - > vereinfacht die Entwicklung von Dateisystemen (`libfuse`)
- > Über 50 Dateisysteme entstanden
 - > exFAT, NTFS-3G, [httpdirfs](#), [CurlFtpFS](#), [rclone](#), Borg...



FUSE

- > seit Linux 2.6.14 enthalten
 - > Kernelmodul fuse muss geladen sein
 - > manche Distributionen erfordern User-Mitgliedschaft in der fuse-Gruppe
- > zwei libfuse-Versionen
 - > FUSE 2 - alte Version [von 2019](#).
 - > FUSE 3 - aktuelle Version, wird aktiv weiter entwickelt



FUSE

Anlegen eines Mountpoints sowie Einhängen eines Webspiegels via [httpdirfs](#):

```
$ mkdir centos  
$ httpdirfs -o ro https://mirror.stream.centos.org/ centos
```

Überprüfen, ob der Mount zustande gekommen ist:

```
$ grep httpdirfs /proc/mounts  
httpdirfs /home/sgiertz/centos fuse.httpdirfs ro,nosuid,nodev,relatime 0 0
```

Auflisten von Inhalten:

```
$ ls centos  
10-stream 9-stream fullfiletimelist SIGs TIME timestamp.txt
```

SPEZIELLE DATEISYSTEME

- > **Pseudodateisysteme**
 - > beinhalten keine tatsächlichen Dateien
 - > machen Informationen über **virtuelle Dateien** zugänglich
- > /proc
 - > beinhaltet hierarisch geordnete Prozessinformationen
- > /sys
 - > dient zur Kontrolle und Steuerung von Hardware und Treibern
- > tmpfs
 - > temporäres Dateisystem im Arbeitsspeicher

/PROC

Beinhaltet zahlreiche Informationen pro Prozess:

```
$ echo $PPID
```

```
133722
```

```
$ ls /proc/133722
```

```
arch_status
```

```
fdinfo
```

```
ns
```

```
smaps_rollup
```

```
attr
```

```
gid_map
```

```
numa_maps
```

```
stack
```

```
autogroup
```

```
io
```

```
oom_adj
```

```
stat
```

```
...
```

- > cmdline, exe - Pfad/symbolischer Link der ausgeführten Datei
- > cwd - symbolischer Link zum aktuellen Arbeitsverzeichnis
- > limits - Limitierungen des Prozesses
- > maps - verwendete Bibliotheken und deren Adressbereiche
- > status - detaillierte Statistiken

/PROC

Weitere prozessunabhängige Dateien:

- > /proc/cmdline - gebootetes Kernel-Image inkl. Parameter
- > /proc/cpuinfo - CPU-Informationen
- > /proc/filesystems - unterstützte Dateisysteme
- > /proc/mdstat - Software-RAID Status
- > /proc/meminfo - Arbeitsspeicher-Informationen
- > /proc/modules - geladene Kernelmodule
- > /proc/partitions - erkannte Partitionen
- > /proc/swaps - benutzter Auslagerungsspeicher
- > /proc/version - Details zum verwendeten Kernel
- > /proc/vmstat - Details zum virtuellen Speicher

/SYS

- > macht Kernel-Subsysteme und Gerätetreiber über Dateien und Ordner zugänglich
- > kann benutzt werden, um Stati auszulesen und Aktionen auszulösen
- > Ordnerstruktur unterhalb /sys:
 - > block - Blockgeräte (Festplatten, SSDs)
 - > bus - Bussysteme (GPIO, Sound, SCSI, PCI,...)
 - > class, devices - verschiedene Geräte-Laufzeitinformationen
 - > fs - Dateisystem-Details
 - > kernel, module - Details zum Kernel und dazugehörigen Modulen

/SYS: BEISPIELE

Temperatur des ersten Sensors (48 C°) anzeigen:

```
$ cat /sys/class/thermal/thermal_zone0/temp  
48000
```

CPU-Verwundbarkeiten auflisten:

```
$ cat /sys/devices/system/cpu/vulnerabilities/*  
Not affected  
Mitigation: Safe RET  
...
```

/SYS: BEISPIELE

Ladestand eines Akkus anzeigen:

```
$ cat /sys/class/power_supply/BAT0/capacity  
100
```

Hersteller einer SCSI-Festplatte (3:0:0:0) auslesen und Geometrie nach Vergrößerung neu einlesen:

```
# cat /sys/class/scsi_disk/3\:0\:0\:0/device/vendor  
VMware  
# echo 1 > /sys/class/scsi_disk/3\:0\:0\:0/device/rescan
```

TMPFS

- > Temporäres Dateisystem, welches sich komplett im Arbeitsverzeichnis befindet
- > auch **RAM-Disk** genannt
- > keinerlei Persistenz, aber hohe Performance
- > nutzt virtuellen Speicher, kann weniger benutzte Blöcke in **Swap** auslagern

```
# mount -t tmpfs -o size=1G tmpfs /tmp/test
# df -h /tmp/test
Dateisystem      Größe Benutzt Verf. Verw% Eingehängt auf
tmpfs            1,0G          0   1,0G    0% /tmp/test
```

ISCSI*

- > **Block-basiertes** Speicherprotokoll
 - > Software-Implementation des SCSI-Protokolls über konventionelle IP-Netze
 - > Komponenten
 - > **Target** stellt Speicher (*Festplatten, Bandlaufwerke, etc.*) zur Verfügung
 - > **Initiator** stellen Verbindungen zu *Targets* her
 - > Verwendete Ports
 - > TCP **860** und TCP/UDP **3260**
 - > **kostengünstige** und pragmatische Alternative zu Fibre Channel
- * Internet **S**mall **C**omputer **S**ystem **I**nterface

ISCSI

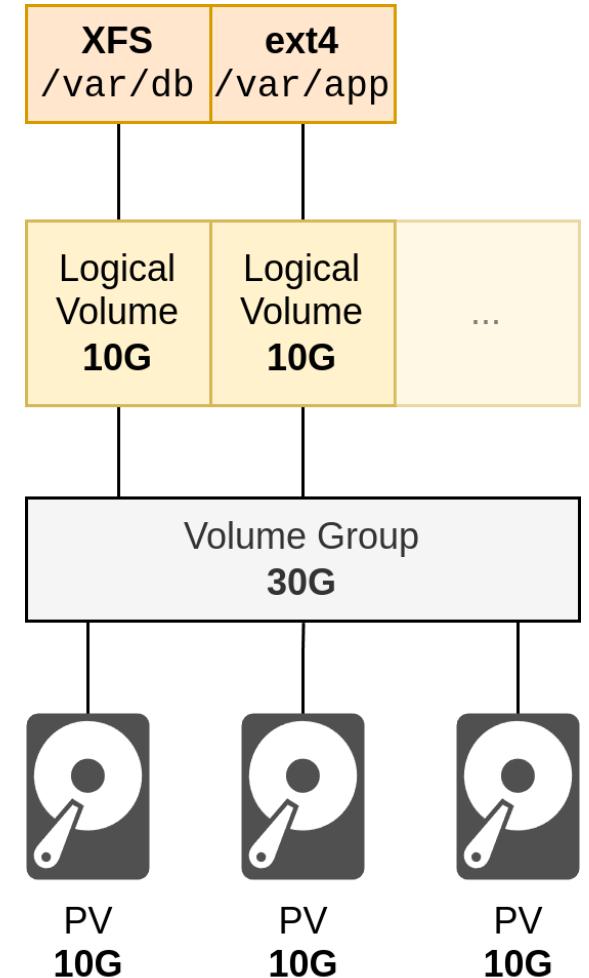
- > Authentifizierung via Klartext-Passwort oder **CHAP***
- > Linux unterstützt iSCSI-Initiator seit 2.6.12 (2005)
 - > seit 3.1 auch iSCSI-Target (**LIO Unified Target**, 2011)
- > Benötigte Anwendungen und Dienste
 - > open-iscsi als Initiator
 - > Konfiguration via iscsiadm
 - > targetcli (Red Hat) bzw. tgt (Debian) für Targets

* Challenge Handshake Authentication Protocol

LVM*

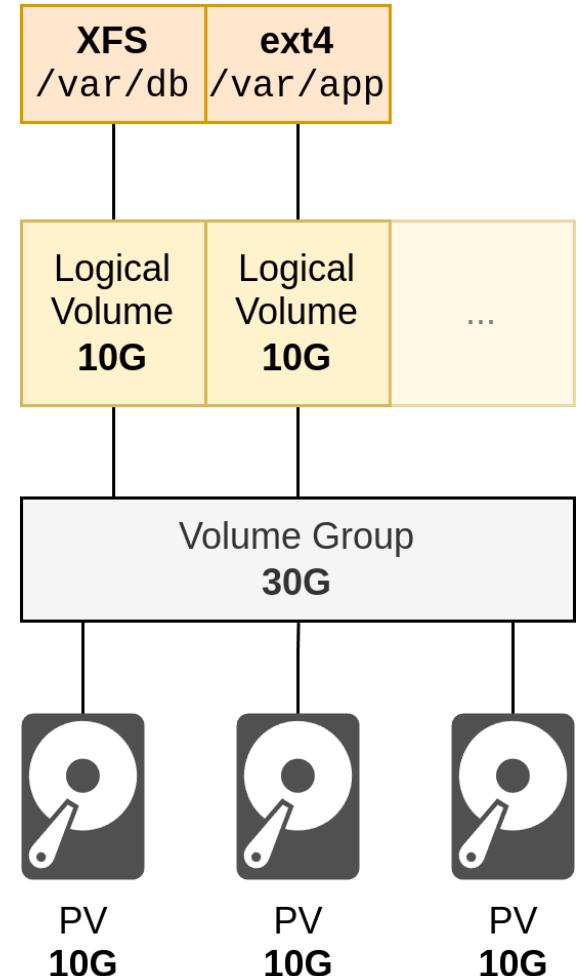
- > Partitionsverwaltung über eine oder mehrere Festplatten
- > weitere **Schicht** zwischen Festplatte und Dateisystem
- > kann **dynamisch** verändert werden
 - > online vergrößern, verkleinern
 - > verschieben auf andere Festplatte
- > Versionierung von Inhalten durch **Snapshots**
- > Stammt ursprünglich von IBM AIX
 - > Linux-Implementation ist an HP-UX angelehnt

* Logical Volume Manager



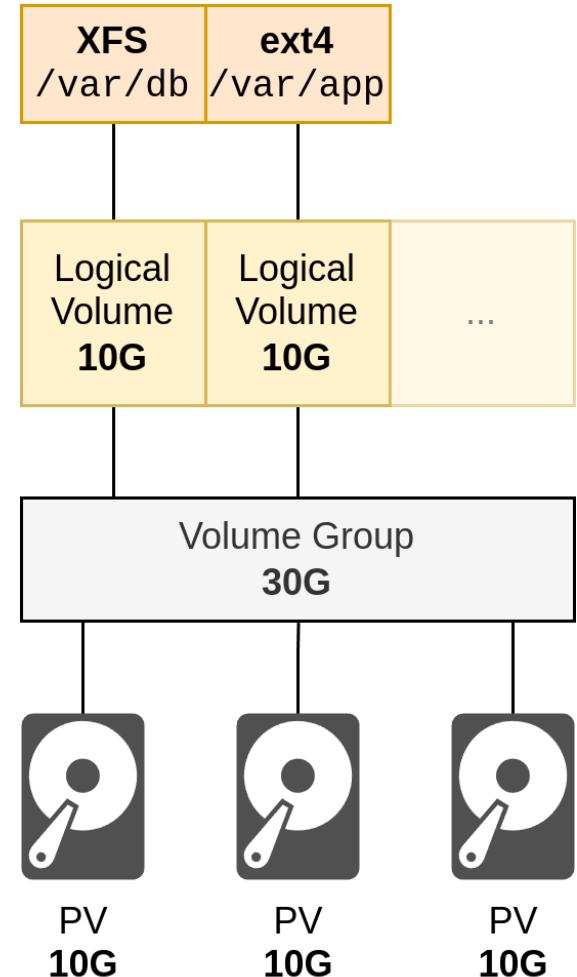
LVM

- > Begriffe
 - > benutzte Laufwerke lauten **Physical Volumes (PV)**
 - > PVs werden in **Volume Groups (VG)** zusammengefasst
 - > VGs beinhalten ein/mehrere **Logical Volumes (LV)**
- > LVs werden mit einem **Dateisystem** versehen
 - > LVs könnten auch weiter partitioniert werden, keine Best Practice



LVM

- > Kommandos beginnen mit `pv`, `vg` oder `lv`:
 - > `pvcreate`, `vgdisplay`, `lvremove`,...
- > Reihenfolge bei der Erstellung
 - > physischen Speicher als **PV** anlegen
 - > dazugehörige **VG** mit PVs erstellen
 - > **LVs** für gewünschte Inhalte anlegen
 - > **Dateisystem** auf LVs erstellen
 - > Dateisysteme einhängen



LVM: BEISPIEL

```
# pvcreate /dev/sdb
# vgcreate vg_data /dev/sdb
# lvcreate vg_data --name lv_db --size 1G

# mkfs.xfs /dev/mapper/vg_data-lv_db1

# mkdir -p /var/data/db1
# mount /dev/mapper/vg_data-lv_db1 /var/data/db1
```

- > Anlegen eines PV sowie einer VG
- > Erstellen eines LV mit einer Größe von **1 GB**
- > Anlegen eines Dateisystems
- > Einhängen des Speichers

LVM

Kommando	Beispiel	Beschreibung
pvcreate	pvcreate /dev/sdb	PV erstellen
pvremove	pvremove /dev/sdb	PV entfernen
pvdisplay	pvdisplay /dev/sdb	PV-Informationen anzeigen
pvresize	pvresize /dev/sdb	Größe eines PV ändern
pvs	pvs	Alle PVs anzeigen
pvscan	pvscan	Nach PVs suchen
pvmove	pvmove /dev/sdb /dev/sdc	Extents zwischen PVs verschieben

LVM

Kommando	Beispiel	Beschreibung
vgcreate	vgcreate --name vg_data /dev/sdb	VG erstellen
vgremove	vgremove vg_data	VG entfernen
vgdisplay	vgdisplay vg_data	VG-Informationen anzeigen
vgextend	vgextend vg_data /dev/sdc	VG erweitern
vgreduce	vgreduce vg_data /dev/sdc	VG verkleinern
vgs	vgs	Alle VGs anzeigen
vgscan	vgscan	Nach VGs suchen

LVM

Kommando	Beispiel	Beschreibung
lvcreate	lvcreate vg_data --name lv_app --size 10G	LV erstellen
lvremove	lvremove vg_data/lv_app	LV entfernen
lvdisplay	lvdisplay vg_data/lv_app	LV-Informationen anzeigen
lvresize	lvresize vg_data/lv_app --size +5G	Größe eines LV ändern
lvs	lvs	Alle LVs anzeigen
lvscan	lvscan	Nach LVs suchen

EXKURS: LSBLK

- > `lsblk` **listet Blockgeräte auf**
- > nützlich beim Erstellen von LVM-Konstrukten und Dateisystemen

```
# lsblk
NAME           MAJ:MIN   RM   SIZE   RO   TYPE   MOUNTPOINTS
sda            252:0     0  19.5G   0   disk
└─sda1          252:1     0      1M   0   part
└─sda2          252:2     0   200M   0   part   /boot/efi
└─sda3          252:3     0      1G   0   part   /boot
└─sda4          252:4     0  18.3G   0   part   /
sdb            252:16    0    10G   0   disk
└─vg_training-lv_test 253:0     0      1G   0   lvm
sdc            252:32    0    10G   0   disk
```

LAB ST01

LVM KONFIGURIEREN

LAB STO2

DATEISYSTEME ANLEGEN

LVM: VERGRÖSSERN/VERKLEINERN

- > LVM kann online die Größe von LVs ändern
- > muss jedoch auch vom **Dateisystem** unterstützt werden
 - > andernfalls wird es bei der Aktion **beschädigt**
- > i.d.R. unterstützen Dateisysteme online das **Vergrößern**, können aber nur verkleinert werden, wenn sie nicht eingehängt sind
 - > **XFS** unterstützt generell kein Verkleinern
 - > Die ext-Familie erfordert einen Konsistenzcheck vor der Verkleinerung
- > Nach dem Vergrößern des LVs muss auch das Dateisystem vergrößert werden
 - > jeweiliges Dateisystem-Tool übernimmt diese Aufgabe
 - > alternativ `lvresize --resize` benutzen

LVM: VERGRÖSSERN/VERKLEINERN - BEISPIEL

Vergrößern eines LVs um 1 GB:

```
# lvresize --size +1G vg_training/lv_data2
```

Vergrößern des XFS-Dateisystems:

```
# xfs_growfs /dev/mapper/vg_training-lv_data2
```

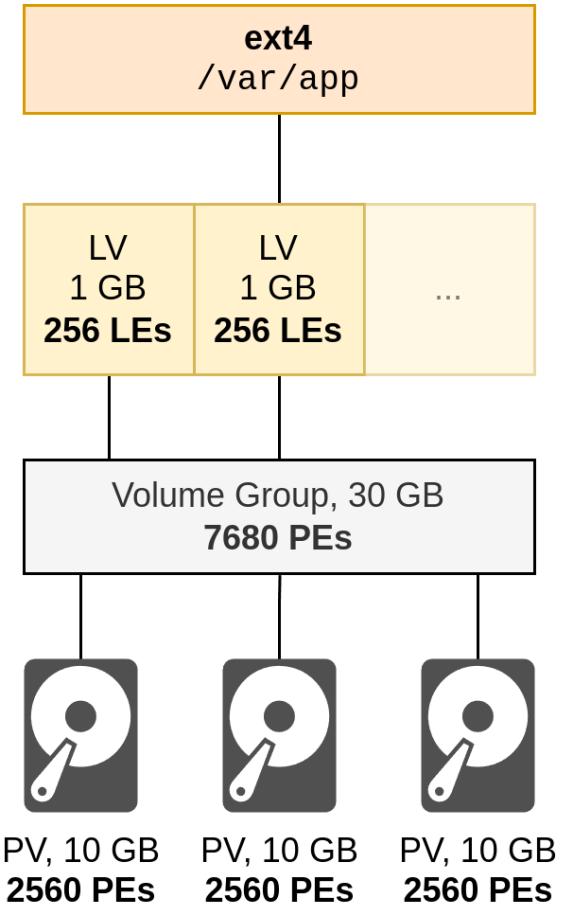
Überprüfen der Dateisystemgröße:

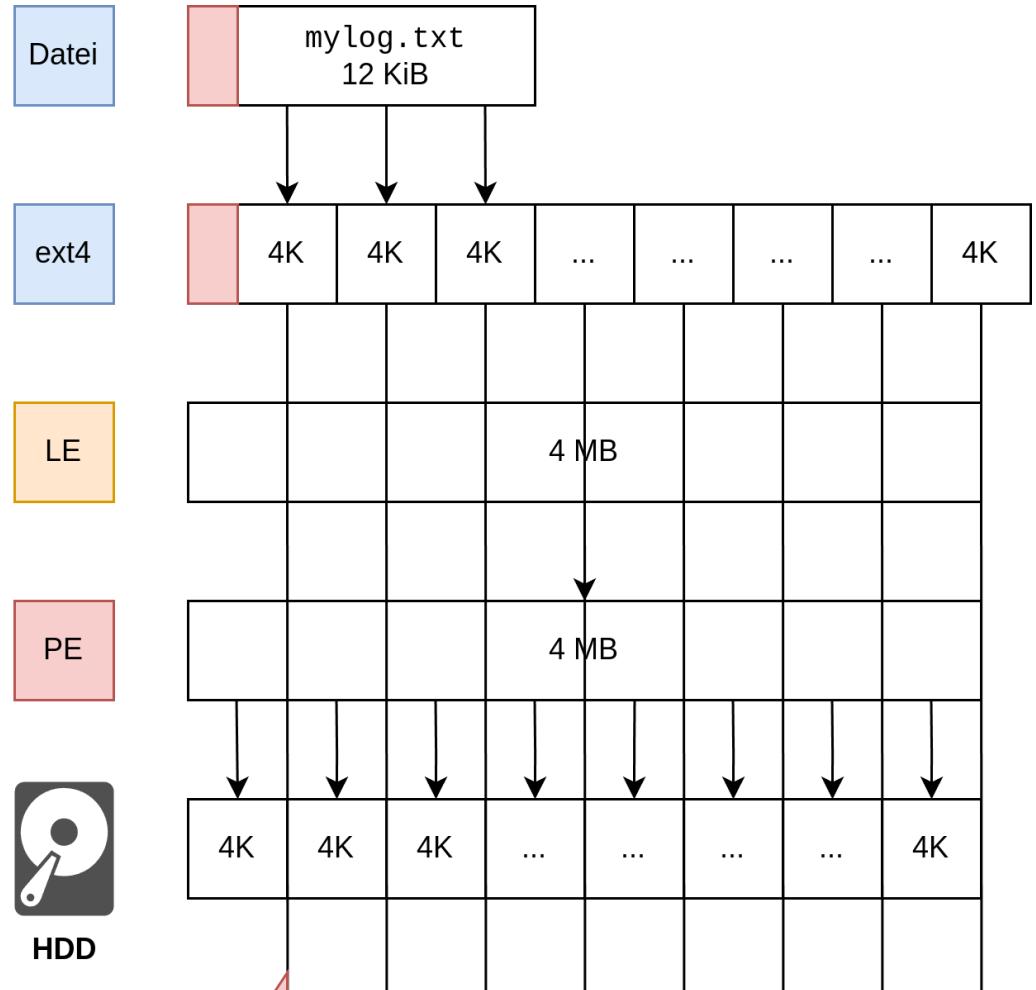
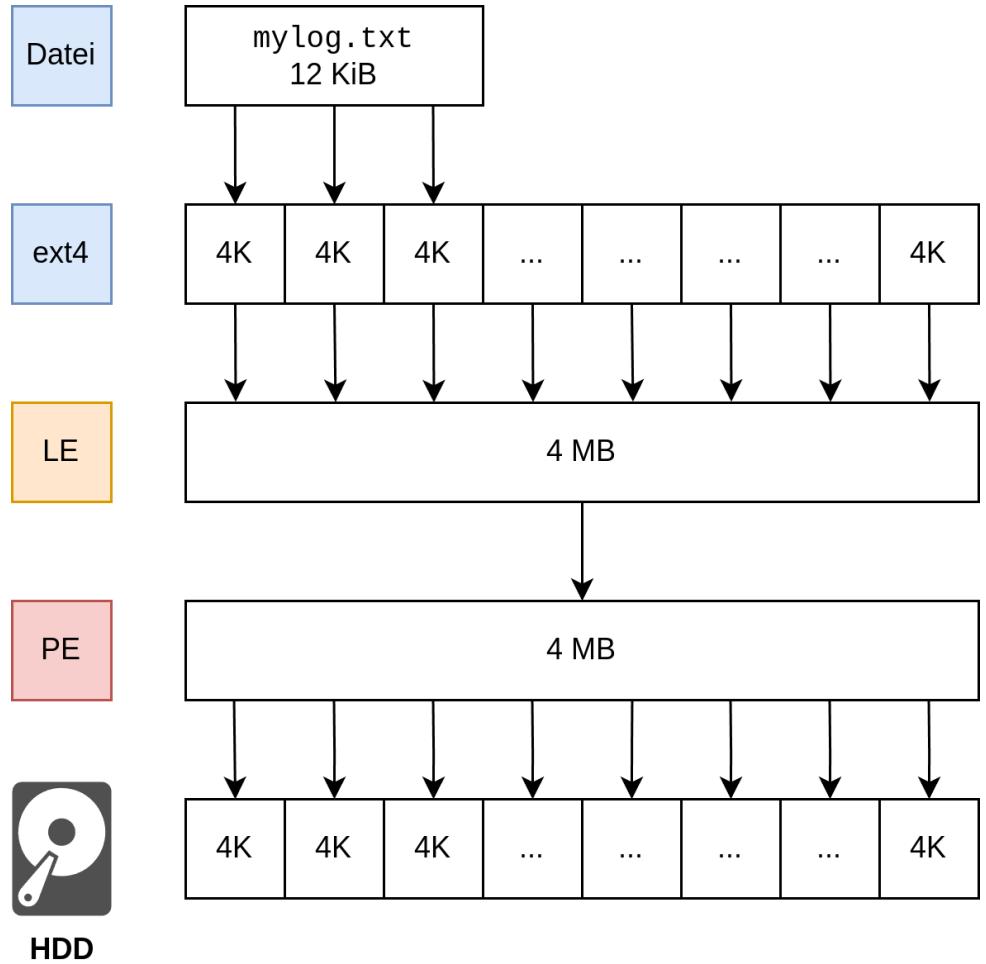
```
# df -h /data2
/dev/mapper/vg_training-lv_data2  3.0G  566M  2.4G  19%  /data2
```

LAB ST03

LVS VERGRÖSSERN UND VERKLEINERN

- > LVM unterteilt PVs in **Physical Extents (PE)**
 - > kleinste Speichereinheit
 - > optimale Größe wird automatisch berechnet
 - > i.d.R. 4 MB
- > LVs werden in **Logical Extents (LE)** aufgeteilt
 - > Größe deckt sich mit PE der PVs
- > VGs übernimmt die Zuordnung der LEs zu PEs
- > Von der Anpassung der PE-/LE-Größen wird abgeraten
 - > Inkorrekt Alignement kann I/O drastisch verlangsamen



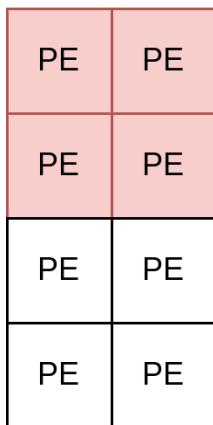
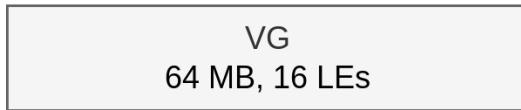


Drei unterschiedliche Modi geben an, wie Daten geschrieben werden:

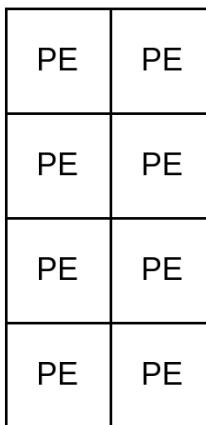
- > **Linear**
 - > verteilt Blöcke sukzessive auf PVs
- > **Mirrored**
 - > verteilt Blöcke gleichzeitig auf mehrere PVs, beugt Datenschutz vor
- > **Striped**
 - > verteilt Blöcke im Versatz auf unterschiedliche PVs
 - > kann in **manchen** Szenarien Datendurchsatz erhöhen

LVM ist **kein RAID-Ersatz**, verwendeter Speicher ist für Ausfallsicherheit zuständig!

linear

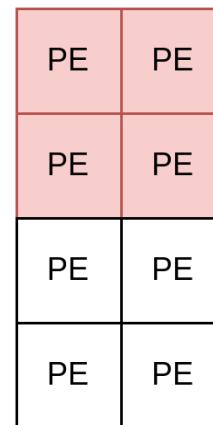
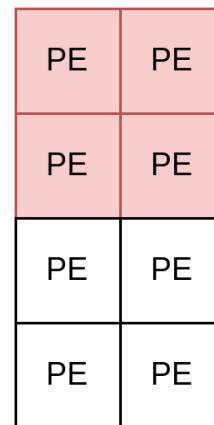
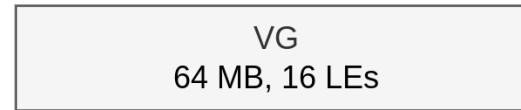
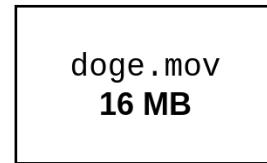


PV



PV

mirrored

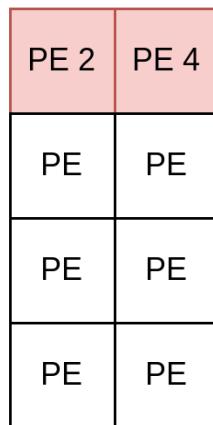
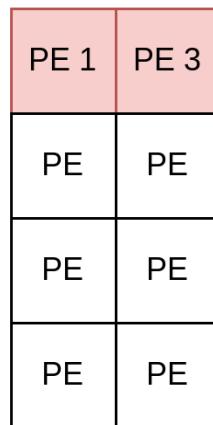
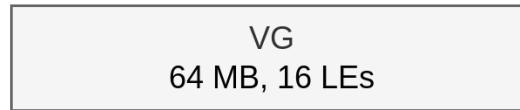


PV



PV

striped



PV



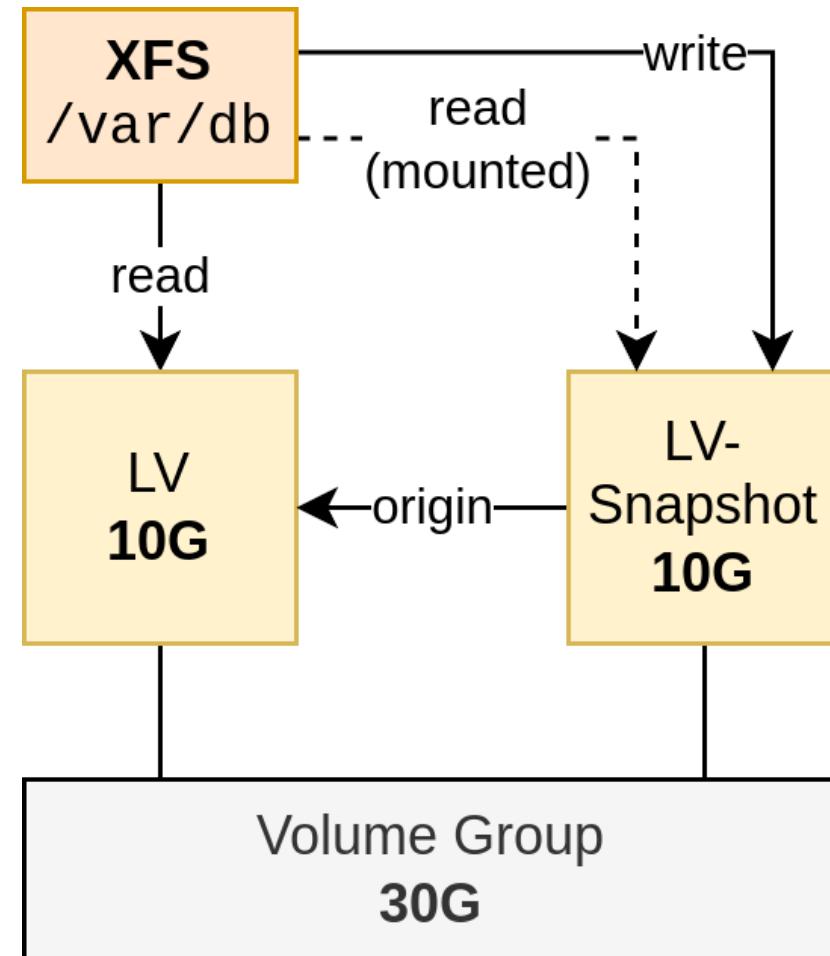
PV

LVM: SNAPSHOTS

- > erlauben **Point-in-time** Kopien von LVs
- > nutzen **Copy-on-write**-Technologie
 - > nur tatsächliche Änderungen werden in den Snapshot geschrieben
 - > Kopien belegen keinen weiteren Blöcke, solange sie unverändert sind
- > werden als LV mit Verknüpfung zum eigentlichen LV angelegt
- > haben eine **feste Größe**, die **überwacht** werden muss
 - > wenn der VG-Speicher erschöpft ist, können Änderungen nicht mehr protokolliert werden
- > sind **kein** Backup und sollten nicht dauerhaft benutzt werden
 - > können jedoch unterstützen, da Dienste während der Sicherung online bleiben können

LVM: SNAPSHTOS

- > Um Änderungen zu verwerfen, muss der Snapshot **zusammengeführt werden**
- > erfordert Aushängen des Speichers



LVM: SNAPSHOTS

Anlegen eines Snapshots:

```
# lvcreate --size 1G -s -n lv_data_snap vg_data/lv_data
Logical volume "lv_data_snap" created.
```

Anzeigen der LVs, der Snapshot hat eine Verknüpfung zu lv_data:

```
# lvs
  LV        VG        Attr      LSize Pool Origin  Data%
  lv_data   vg_data  owi-aos--- 1.00g
  lv_data_snap  vg_data swi-a-s--- 1.00g          lv_data  0.01
```

LVM: SNAPSHOTS

Der Snapshot lässt sich im laufenden Betrieb einhängen und zeigt vorherige Daten:

```
# mount /dev/vg_data/lv_data_snap /mnt/data_snap  
  
# ls /mnt/data /mnt/data_snap  
/mnt/data:  
bin.img bin2.img  
  
/mnt/data_snap:  
bin.img
```

LVM: SNAPSHTOS

Änderungen **verwerfen** und vorherigen Zustand wiederherstellen:

```
# lvconvert --merge vg_data/lv_data_snap
Delaying merge since origin is open.
Merging of snapshot vg_data/lv_data_snap will occur on next activation
of vg_data/lv_data.
# umount /mnt/test /mnt/test_snap
# lvchange -an vg_data/lv_data
# lvchange -ay vg_data/lv_data
```

Einhängen der Daten:

```
# mount /dev/vg_data/lv_data /mnt/data
# ls /mnt/data
bin.img
```

LVM: SNAPSHOTS

Zur Übernahme der Daten muss lediglich der Snapshot gelöscht werden:

```
# lvremove vg_data/lv_data
```

Ein Aushängen des Speichers wird nicht benötigt.

LAB STo4

LVM-SNAPSHOTS BENUTZEN

LAB ST05

VG ERWEITERN

LUKS*

- > **Spezifikation** zur Festplatten-Verschlüsselung
- > Platformunabhängiger Standard für Linux
- > verschlüsselt Dateisystem-unabhängig **Blockspeicher**
- > existiert seit 2004, inzwischen gibt es **zwei** Versionen
- > Ein Hauptschlüssel und mehrere weitere Schlüssel definierbar
- > für einzelne Partitionen oder Gesamtsystem (**Full-Disk Encryption**) geeignet
 - > kleiner unverschlüsselter Header am Anfang einer Partition
 - > die erste Stufe des Bootloaders lässt sich nicht verschlüsseln

* Linux Unified Key Setup

LUKS

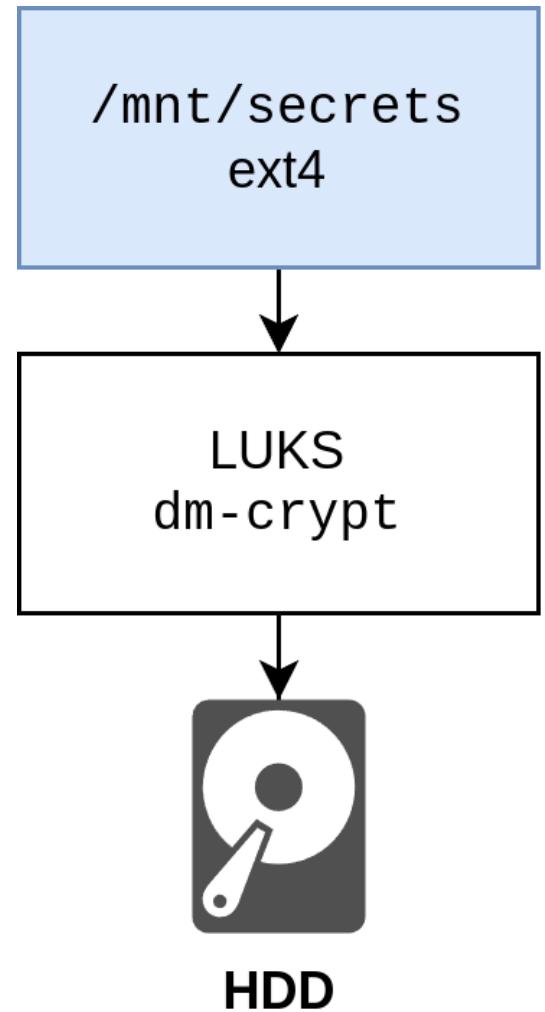
	LUKS1	LUKS2
Erschienen	2004	2017
Anzahl Header	1	Backup-Header
Header-Algorithmus	PBKDF2	Argon2i/id
Max. Schlüssel	8	32
Bootfähig	Ja	Ja (seit GRUB 2.06, 06/2020)

- > Gerät wird mit **Master-Key** verschlüsselt
 - > wird mit weiteren optionalen **User Keys** verschlüsselt
- > Entschlüsselung auch via Smartcard, TPM oder FIDO2 möglich

LUKS

Reihenfolge der Implementation:

1. Formatierung mit cryptsetup
2. Volume öffnen
3. Anlegen eines Dateisystems
4. Einhängen und Verwenden des Dateisystems



LUKS: BEISPIEL

Anlegen und Öffnen eines LUKS-Volumes:

```
# cryptsetup luksFormat /dev/sdb
WARNING!
=====
This will overwrite data on /dev/sdb irrevocably.

Are you sure? (Type 'yes' in capital letters): YES
Enter passphrase for /dev/sdb: LUKS1337
Verify passphrase: LUKS1337

# cryptsetup open /dev/sdb secret_data
Enter passphrase for /dev/sdb: LUKS1337
```

LUKS: BEISPIEL

Anlegen und Einhängen eines Dateisystems:

```
# mkfs.ext4 /dev/mapper/secret_data  
# mount /dev/mapper/secret_data /mnt/secret_data
```

Weiteren Schlüssel hinzufügen:

```
# cryptsetup luksAddKey /dev/sdb  
Enter any existing passphrase: LUKS1337  
Enter new passphrase for key slot: Linux1337  
Verify passphrase: Linux1337
```

Speicher aushängen und LUKS-Volume stoppen:

```
# umount /mnt/secret_data  
# cryptsetup close /dev/sdb
```

LUKS: BEISPIEL

Anzeigen von Header-Informationen:

```
# cryptsetup luksDump /dev/sdb
LUKS header information
Version:          2
Epoch:            4
Metadata area:   16384 [bytes]
Keyslots area:   16744448 [bytes]
UUID:             ea15183b-ec73-4b75-9947-7648ad112c29
Label:            (no label)
...
Keyslots:
  0: luks2
    Key:      512 bits
    ...
  1: luks2
    Key:      512 bits
    ...
```

LUKS: WEITERE KOMMANDOS

Kommando	Beschreibung
<code>cryptsetup luksChangeKey <device></code>	Hauptschlüssel ändern
<code>cryptsetup luksRemoveKey <device> <slot></code>	Ein bekanntes Passwort entfernen
<code>cryptsetup luksKillSlot <device> <slot></code>	Einen bestimmten Schlüssel entfernen
<code>cryptsetup luksHeaderBackup <device> --header-backup-file <name></code>	Header-Backup erstellen
<code>cryptsetup luksHeaderRestore <device> --header-backup-file <name></code>	Header-Backup wiederherstellen

LAB STo6

LUKS KONFIGURIEREN

LUKS: AUTOMATISCHE ENTSCHLÜSSELUNG

- > Eingabe von Passphrase z.B. bei Servern ungünstig
 - > Prompt muss **beim Boot** ausgefüllt werden
- > Eine Option ist die Erstellung einer **Schlüsseldatei** oder eines -skripts
 - > diese beinhaltet entweder das Klartext-Passwort oder generiert es
 - > sicherheitstechnisch **problematisch**

LUKS: AUTOMATISCHE ENTSCHLÜSSELUNG

- > Eingabe von Passphrase z.B. bei Servern ungünstig
 - > Prompt muss **beim Boot** ausgefüllt werden
- > Eine Option ist die Erstellung einer **Schlüsseldatei** oder eines -skripts
 - > diese beinhaltet entweder das Klartext-Passwort oder generiert es
 - > sicherheitstechnisch **problematisch**
- > Bessere aber komplexe Alternative: **NBDE***
 - > LUKS-Erweiterung für serverbasierte Entschlüsselung
 - > Passphrases müssen nicht mehr eingegeben werden

* **Network-bound Disk Encryption**

SOFTWARE-RAID

- > in Software implementierter Zusammenschluss mehrerer Festplatten
- > Erhöhung von Datendurchsatz und/oder Verfügbarkeit
- > kein Ersatz für funktionale Backups

SOFTWARE-RAID

- > in Software implementierter Zusammenschluss mehrerer Festplatten
- > Erhöhung von Datendurchsatz und/oder Verfügbarkeit
- > kein Ersatz für funktionale Backups
- > **Vorteile** gegenüber Hardware-RAID
 - > günstiger/einfacher zu implementieren
 - > Verbund kann in anderem Linux-Rechner weiterbenutzt werden
- > **Nachteile** gegenüber Hardware-RAID
 - > Host-CPU wird geringfügig höher belastet
 - > aufgrund fehlender Pufferbatterie kein Schutz vor Stromausfall

SOFTWARE-RAID

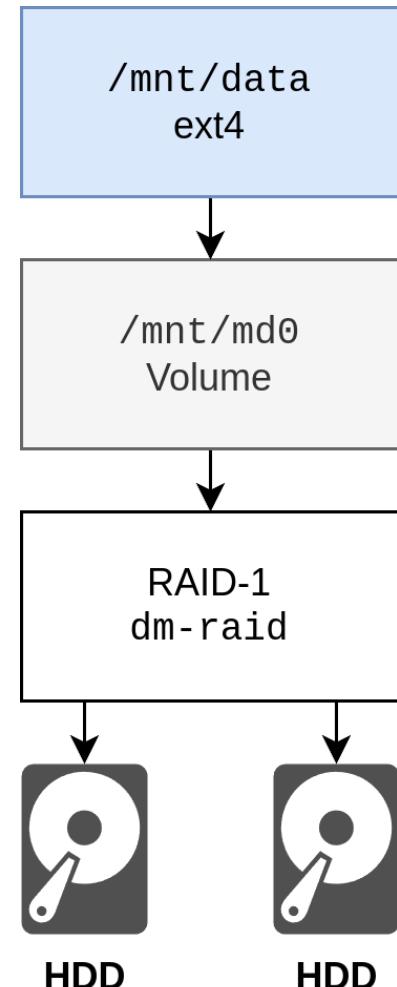
Linux unterstützt die üblichen Level:

Level	HDDs	Kapazität	Reserve	Beschreibung
RAID-0	mind. 2	$x * n$	keine	Striping, erhöhte Geschwindigkeit/Kapazität
RAID-1	mind. 2	x	bis auf eine	Mirroring, Spiegelung, erhöhte Sicherheit und Lesegeschwindigkeit
RAID-5	mind. 3	$x * (n - 1)$	eine darf ausfallen	Kompromiss aus Sicherheit, Kapazität und Lesegeschwindigkeit
RAID-6	mind. 4	$x * (n - 2)$	zwei dürfen ausfallen	Ausfallsicherheit und Lesegeschwindigkeit
RAID-10	mind. 4			Sicherheit und gesteigerte Lese-/Schreibgeschwindigkeit, RAID-0 über mehrere RAID-1

SOFTWARE-RAID

Vorgehensweise:

1. Partitionieren der Festplatten
2. Erstellen des RAIDs mittels mdadm
3. Warten, bis Initialisierung abgeschlossen ist
4. Dateisystem erstellen



SOFTWARE-RAID: BEISPIEL

Festplatten partitionieren:

```
# parted /dev/sdb mklabel gpt
# parted -a optimal -- /dev/sdb mkpart primary 0% 100%
# parted /dev/sdb set 1 raid on
...
```

RAID o-Volume erstellen:

```
# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb1 /dev/sdc1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

SOFTWARE-RAID

- > Ein RAID weist einen **Zustand** auf:
 - > **Clean** - Normzustand, intakt
 - > **Degraded** - Ausfall liegt vor, je nach RAID-Level noch funktionsfähig
 - > **Resync** - Sicherungsinformationen werden geprüft und ggf. korrigiert - ist u.a. nach dem Anlegen der Fall
 - > **Rebuild** - Erholung nach einem Ausfall, verlorene Paritäten werden wiederhergestellt
- > Synchronisationsdauer richtet sich nach Größe und RAID-Level
 - > mehrere Stunden bis Tage

SOFTWARE-RAID

- > RAIDs können auch **konvertiert** werden
 - > RAID-5 zu RAID-0/4/5/10
 - > RAID-6 zu RAID-5
 - > RAID-10 zu RAID-0
- > auch **Hotspares** werden unterstützt
 - > Festplatten, die im Fehlerfall sofort als Ersatz dienen können

SOFTWARE-RAID: MDADM

Befehl	Beschreibung
mdadm --create <name> --level=0 --raid-devices=2 <device> <device>	Legt ein RAID-0 mit 2 Laufwerken an
mdadm --detail <name>	Zeigt Details eines RAIDs an
mdadm --examine <name>	Zeigt Festplatten-Details an
mdadm --stop <name>	Stoppt ein RAID
mdadm --zero-superblock <device>	Löscht Superblock eines RAID-Mitglieds
mdadm --add <name> <device>	Fügt eine Festplatte zum RAID hinzu, z.B. nach einem Ausfall
mdadm --remove <name> <device>	Entfernt eine Festplatte vom RAID
mdadm --manage --set-faulty <name> <device>	Markiert eine Festplatte als fehlerhaft

LAB ST07

SOFTWARE-RAID KONFIGURIEREN

SOFTWARE-RAID: FEHLERBEHEBUNG

Wenn in einem RAID-1/5/6-Verbund eine Festplatte defekt ist, kann diese gewechselt werden. Falls noch nicht geschehen, diese aus dem Verbund entfernen:

```
# mdadm /dev/md0 --remove /dev/sdc
```

Neue Festplatte partitionieren und aufnehmen:

```
# parted /dev/sdd
# mdadm /dev/md0 --add /dev/sdc
```

Status überwachen:

```
# mdadm --detail /dev/md0
```

SOFTWARE-RAID: ÜBERWACHUNG

- > Software-RAIDs brauchen unter Linux prinzipiell keine **Konfigurationsdatei**
- > beim Boot werden alle RAIDs mit validen Superblöcken erkannt und gestartet

SOFTWARE-RAID: ÜBERWACHUNG

- > Software-RAIDs brauchen unter Linux prinzipiell keine **Konfigurationsdatei**
- > beim Boot werden alle RAIDs mit validen Superblöcken erkannt und gestartet
- > Die Datei `/etc/mdadm/mdadm.conf` kann jedoch generiert werden
 - > z.B. um eine **Mail-Benachrichtigungen** zu konfigurieren
 - > Beispiel: MAILADDR sgiertz@robo.ts

```
debian# /usr/share/mdadm/mkconf > /etc/mdadm/mdadm.conf  
redhat# mdadm --detail --scan >> /etc/mdadm.conf
```

- > gängige Monitoringsysteme haben hierfür ebenfalls Plugins

LAB STo8

DATEISYSTEME AUTOMATISCH EINHÄNGEN UND TUNEN

ZUSAMMENFASSUNG

- > Linux unterstützt zahlreiche verschiedene Storage-Protokolle und -Medien
- > **skaliert** von Kleinstrechnern über Notebooks und Server bis hin zu Mainframes
- > Der IO-Scheduler blk-mq wurde schon mit **über 15. Mio.** IOPS getestet

ZUSAMMENFASSUNG

- > Linux unterstützt zahlreiche verschiedene Storage-Protokolle und -Medien
- > **skaliert** von Kleinstrechnern über Notebooks und Server bis hin zu Mainframes
- > Der IO-Scheduler `blk-mq` wurde schon mit **über 15. Mio.** IOPS getestet
- > zahlreiche **Dateisysteme** werden zur Installation und Benutzung unterstützt
- > Mit **LVM** können Partitionen dynamisch verändert werden
- > Durch den Einsatz von **LUKS** können Speicher verschlüsselt werden
- > Mehrere Festplatten können im **RAID**-Verbund genutzt werden, um Durchsatz, Kapazität oder Ausfallsicherheit zu erhöhen

// SYSTEMÜBERWACHUNG

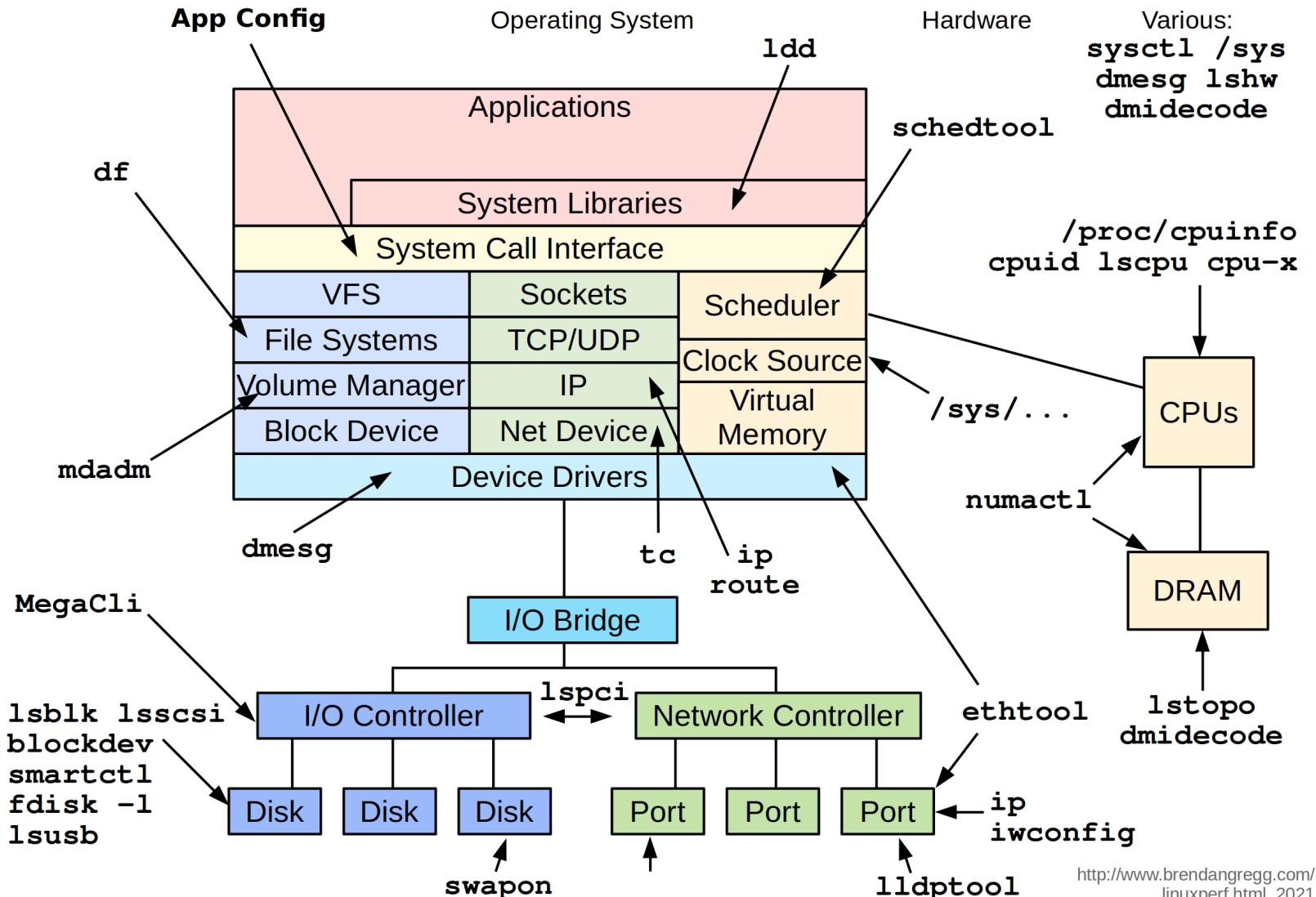
SYSTEMÜBERWACHUNG

Linux bietet zahlreiche Tools zur Systemüberwachung, u.a.:*

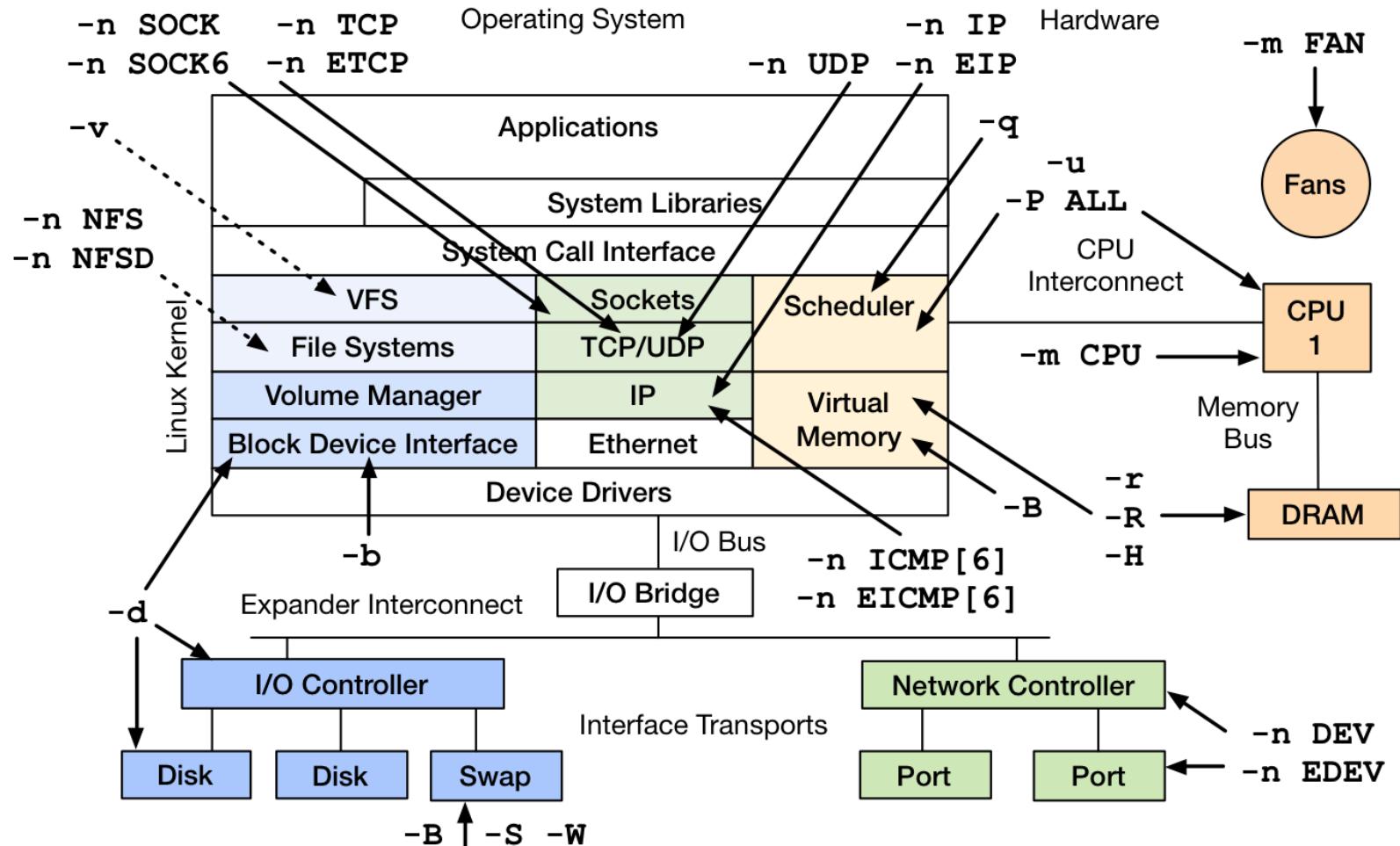
- > Allgemein: top, htop, nmon, sar
- > Prozesse: ps, top, htop
- > Dateien: lsof
- > Storage: iotop, iostat
- > Speicher: free, top, htop, vmstat, smem, mpstat, swapon
- > Netzwerk: netstat, ss, iptraf

* die Liste lässt sich beliebig fortführen

Linux Static Performance Tools



Linux Performance Observability: sar



PS

ps listet laufende Prozesse auf:

Parameter	Erklärung
-e / -A	alle Prozesse anzeigen
-f	mehr Details anzeigen
-C	Nach Programm suchen
-u / -U / --user	Programme eines bestimmten Users anzeigen

```
$ ps
  PID TTY          TIME CMD
 93439 pts/0        00:00:00 bash
238881 pts/0        00:00:00 ps
```

PS

Auflisten **eigener** Prozesse:

```
$ ps -fU cstankow
UID          PID      PPID    C  STIME   TTY                  TIME CMD
cstankow     3540      3197    2  08:23 ?                00:15:13 /usr/bin/gnome-shell
cstankow     17621     14224    0  08:29 ?                00:00:12 /usr/lib64/firefox/firefox
...
```

Auflisten **aller** Prozesse:

```
$ ps -ef
UID          PID      PPID    C  STIME   TTY                  TIME CMD
root         1          0    0  08:23 ?                00:00:46 /usr/lib/systemd/systemd --sw
root         2          0    0  08:23 ?                00:00:00 [kthreadd]
root         3          2    0  08:23 ?                00:00:00 [pool_workqueue_release]
...
```

TOP

Parameter	Erklärung
-i	Schlafende Prozesse ignorieren
-u / -U	Nur Prozesse eines bestimmten Users anzeigen

- > top agiert interaktiv in der Shell
- > **Tasten** lösen Aktionen aus
- > **Leertaste** aktualisiert die Ansicht
- > in jeder Distribution enthalten

Taste	Erklärung
h	Hilfe anzeigen
z	Farbe aktivieren
P	Nach CPU-Last sortieren
M	Nach RAM-Verbrauch sortieren
T	Nach Zeit sortieren
U	Nach User filtern
k	Prozess mit PID beenden
m	RAM-Darstellung ändern
1	Einzelne CPUs anzeigen
q	Beenden

HTOP

Parameter	Erklärung
-t / --tree	Prozesse als Bäume darstellen
-u / --user	Nur Prozesse eines bestimmten Users anzeigen

- > wesentlich **einfacher** zu benutzen
- > bessere **farbliche** Hervorhebung
- > nicht in allen Distributionen enthalten

Taste	Erklärung
Pfeiltasten	Navigieren
F1, ?, h	Hilfe anzeigen
u	Nach User filtern
F2	Konfiguration anpassen
F3	Nach Prozess suchen
F4	Nach Prozessnamen filtern
F5	Baumansicht de-/aktivieren
+ / -	Baum aus-/einklappen
P / M	Nach CPU-/RAM-Last sortieren
T	Nach Zeit sortieren
q / STRG+C	Beenden

TOP VS. HTOP

```
cstankow@TPCSTANKOW:~ --top
top - 18:20:27 up 9:21, 2 users, load average: 1,63, 1,61, 1,48
Tasks: 528 total, 1 running, 527 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,6 us, 0,8 sy, 0,0 ni, 98,3 id, 0,0 wa, 0,2 hi, 0,1 si, 0,0 st
MiB Mem : 30838,0 total, 2718,1 free, 8970,4 used, 19149,4 buff/cache
MiB Swap: 8192,0 total, 8191,5 free, 0,5 used. 20830,8 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
374226 root 0 -20 0 0 I 0,0 0,0 0:00.00 kworker+
374225 root 0 -20 0 0 I 0,0 0,0 0:00.00 kworker+
374155 cstankow 20 0 225516 4096 2944 R 1,0 0,0 0:00.20 top
374116 root 0 -20 0 0 I 0,0 0,0 0:00.00 kworker+
374091 root 20 0 0 0 I 0,0 0,0 0:00.00 kworker+
374074 root 20 0 15280 6656 5888 S 0,0 0,0 0:00.00 systemd+
373962 root 20 0 0 0 I 0,0 0,0 0:00.00 kworker+
373873 cstankow 20 0 1183444 92564 75264 S 0,0 0,3 0:00.14 gnome-c+
373671 cstankow 20 0 2598208 61348 48036 S 0,0 0,2 0:00.13 Web Con+
373633 root 20 0 0 0 I 0,0 0,0 0:00.00 kworker+
373615 root 0 -20 0 0 I 0,0 0,0 0:00.00 kworker+
373551 cstankow 20 0 2598208 62276 48964 S 0,0 0,2 0:00.15 Web Con+
373538 cstankow 20 0 2598208 62748 49436 S 0,0 0,2 0:00.16 Web Con+
373520 root 20 0 0 0 I 0,0 0,0 0:00.00 kworker+
373472 root 20 0 0 0 I 0,0 0,0 0:00.00 kworker+
373441 root 20 0 0 0 I 0,0 0,0 0:00.00 kworker+
372679 cstankow 20 0 2710952 126772 87320 S 0,0 0,4 0:00.84 Isolate+
```

```
cstankow@TPCSTANKOW:~ --htop
0[|||] 3.1% 4[|||] 6.7% 8[|||] 1.8% 12[|||] 4.2%
1[|||] 5.4% 5[|||] 5.5% 9[|||] 3.1% 13[|||] 1.2%
2[|||] 5.0% 6[|||] 6.7% 10[|||] 6.1% 14[|||] 3.1%
3[|||] 4.9% 7[|||] 2.4% 11[|||] 0.6% 15[|||||] 16.6%
Mem[|||||||||] 9.37G/30.1G Tasks: 242, 2133 thr, 281 kthr; 0 runn
Swp[|||] 512K/8.00G Load average: 1.46 1.58 1.47
Uptime: 09:21:12

Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
374301 cstankow 20 0 222M 8320 4224 R 11.7 0.0 0:00.72 htop
3217 cstankow 20 0 5986M 373M 145M S 6.8 1.2 19:56.46 /usr/bin/gnom
2867 cstankow 20 0 1235M 209M 136M S 4.9 0.7 17:57.32 /usr/libexec/
44671 cstankow 20 0 1134G 302M 101M S 4.3 1.0 11:29.28 /usr/share/co
2065 root 20 0 2333M 524M 284M S 3.7 1.7 0:50.32 s1-agent
15683 cstankow 20 0 1140G 754M 123M S 3.7 2.4 22:38.53 /app/chromium
4736 root 20 0 2333M 524M 284M S 3.1 1.7 0:57.43 s1-agent
2960 cstankow 20 0 1235M 209M 136M S 1.8 0.7 0:54.31 /usr/libexec/
44573 cstankow 20 0 1124G 192M 130M S 1.8 0.6 2:08.12 /usr/share/co
86473 cstankow 20 0 1729M 175M 104M S 1.8 0.6 9:30.62 /usr/share/of
350643 cstankow 20 0 814M 50076 38760 S 1.8 0.2 0:07.41 /usr/libexec/
374303 cstankow 39 19 669M 30592 24064 S 1.8 0.1 0:00.03 /usr/libexec/
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
```

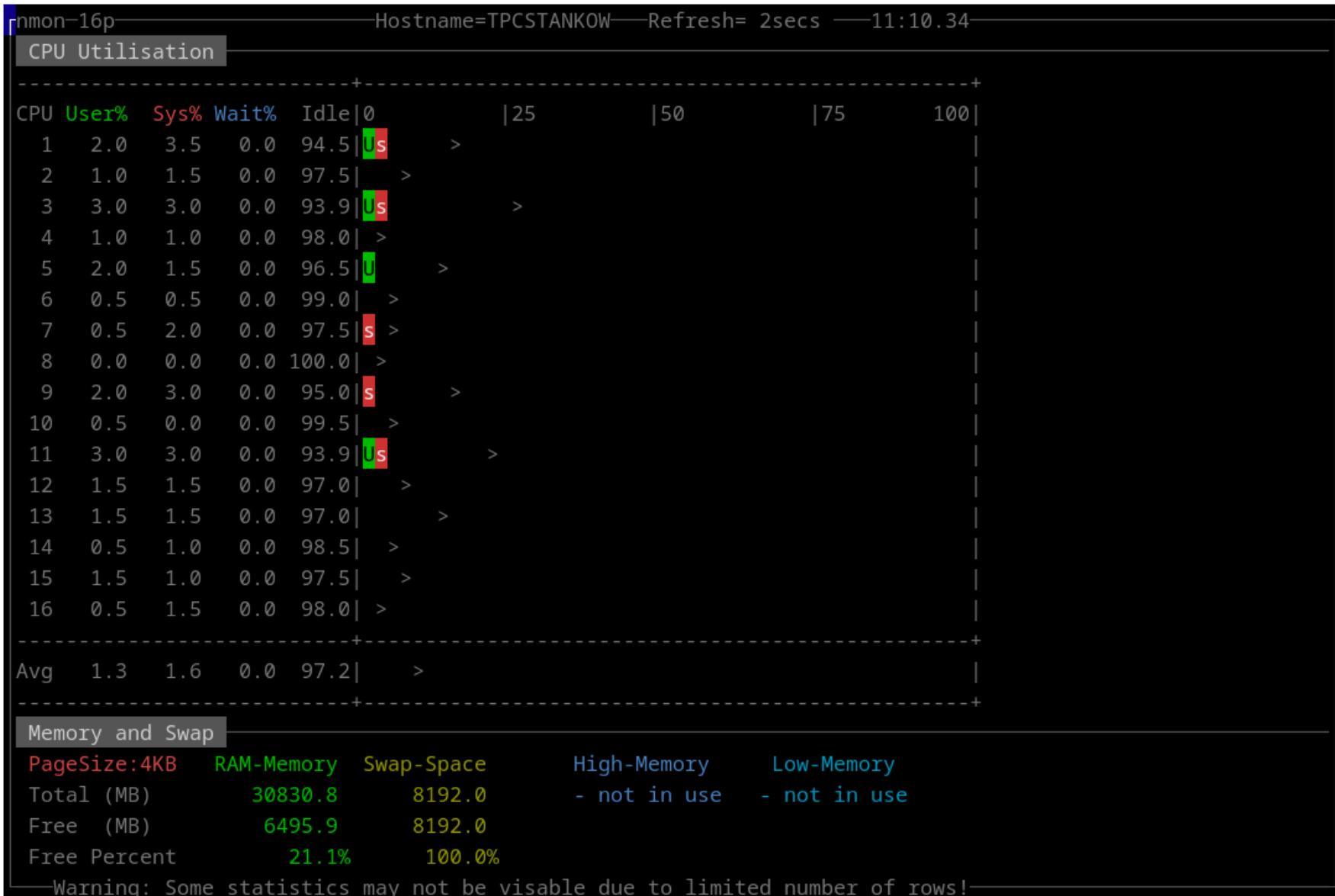
NMON

- > Nigel's Monitor
- > zeigt AIX- und Linux-Metriken

Parameter	Erklärung
-f	CSV-Export

Taste	Erklärung
Pfeiltasten	Navigieren
H	Hilfe

Taste	Erklärung
c	CPU
m	Arbeitsspeicher
d	Festplatten
j	Dateisysteme
k	Kernel
n	Netzwerk
t	Prozesse mit größter Last
- / +	Seltener/öfter aktualisieren
.	nur beschäftigte Entitäten zeigen



KILL

- > `kill` (*und auch top*) können **Prozesssignale** senden
- > damit können Prozesse (un)säuber beendet oder gesteuert werden:

Signal	Beschreibung
SIGTERM (15)	User wünscht sauberes Beenden (Standard)
SIGINT (2)	User wünscht sauberes Beenden (STRG+C)
SIGKILL (9)	Prozess wird rabiät gestoppt (unsäuber)

- > Programme können SIGINT und SIGTERM ignorieren
- > **Nicht mehr reagierende** Programme sollten zunächst mit SIGTERM oder SIGINT beendet werden

AUSLAGERUNGSSPEICHER

- > Es gibt verschiedene Speicherarten:
 - > **Physischer Speicher** = verbauter RAM mit **verbindlichen** Speicheradressen
 - > **Virtueller Speicher** = vom Betriebssystem **abstrahierte** Adressierung

AUSLAGERUNGSSPEICHER

- > Es gibt verschiedene Speicherarten:
 - > **Physischer Speicher** = verbauter RAM mit **verbindlichen** Speicheradressen
 - > **Virtueller Speicher** = vom Betriebssystem **abstrahierte** Adressierung
- > Linux verfügt über Auslagerungsspeicher, auch **Swap** genannt
- > kann u.a. **beinhalten**:
 - > Programmspeicher und -daten
 - > Dateisystem-Cache

AUSLAGERUNGSSPEICHER

- > Es gibt verschiedene Speicherarten:
 - > **Physischer Speicher** = verbauter RAM mit **verbindlichen** Speicheradressen
 - > **Virtueller Speicher** = vom Betriebssystem **abstrahierte** Adressierung
- > Linux verfügt über Auslagerungsspeicher, auch **Swap** genannt
- > kann u.a. **beinhalten**:
 - > Programmspeicher und -daten
 - > Dateisystem-Cache
- > wird i.d.R. nur benutzt, wenn RAM **erschöpft** ist
- > wenn ein System dauerhaft Swap benutzt, liegt eine **Fehlkonfiguration** vor

AUSLAGERUNGSSPEICHER

- > Swap kann auf **zwei Arten** angelegt werden:
 - > dedizierte Partition
 - > Datei auf bestehender Partition (**langsamer**)

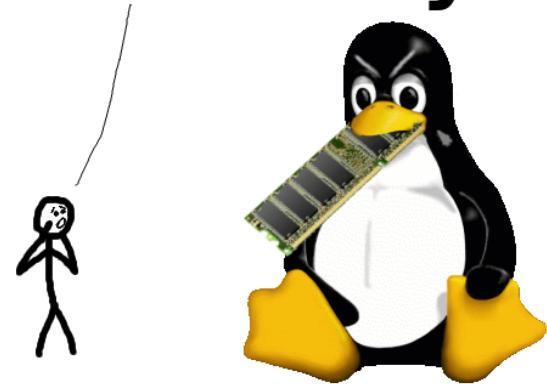
AUSLAGERUNGSSPEICHER

- > Swap kann auf **zwei Arten** angelegt werden:
 - > dedizierte Partition
 - > Datei auf bestehender Partition (**langsamer**)
- > Swap-Größe richtet sich nach RAM-Größe
 - > i.d.R. 20% der RAM-Größe, z.B. ~6 GB bei 32 GB
 - > **Ausnahmen** für besonders großen oder kleinen RAM
- > Programme wie swapon, free und top zeigen Swap-Auslastung an

EXKURS: LINUX ATE MY RAM

- > Linux nutzt ungenutzten Speicher als Dateisystem-Cache
 - > I/O wird dadurch beschleunigt
- > anscheinend wenig Speicher frei
 - > wird jedoch freigegeben, falls benötigt
- > Definition
 - > used - von Anwendungen benutzt
 - > available - als Cache belegt, kann aber freigegeben werden
 - > free - generell nicht benutzt

Linux ate my ram!



EXKURS: LINUX ATE MY RAM

Es besteht kein Grund zur Sorge, wenn:

- > freier Speicher (`free`) gering ist
- > verfügbarer Speicher (`available`) ausreichend gegeben ist
- > sich die Swap-Auslastung (`swap used`) nicht ändert

EXKURS: LINUX ATE MY RAM

Es besteht kein Grund zur Sorge, wenn:

- > freier Speicher (`free`) gering ist
- > verfügbarer Speicher (`available`) ausreichend gegeben ist
- > sich die Swap-Auslastung (`swap used`) nicht ändert

Probleme drohen, wenn:

- > verfügbarer Speicher (`available`) gering ist
- > die Swap-Auslastung (`swap used`) kontinuierlich steigt
- > das System aufgrund Speicherknappheit Prozesse beenden muss
 - > `dmesg | grep oom-killer`

AUSLAGERUNGSSPEICHER

Anzeigen von aktiertem Swap-Speicher:

```
$ swapon -s
Filename      Type            Size      Used  Priority
/dev/zram0    partition     8388604      0      100
```

Hinweis: Eine Partition mit ~8 GB Speicher ist aktiv, wird aber nicht benutzt.

Anzeigen von RAM- und Swap-Auslastung:

```
$ free
              total        used         free        shared      buff/cache   available
Mem:       31578084     8851112     12387652     281748     10339320     21979476
Swap:      8388604        0        8388604
```

LAB Mo1

PROZESSE KONTROLIEREN

SAR

- > **System Activity Report**
- > existiert schon seit UNIX System V, unter Linux Teil des sysstat-Pakets
- > Tool zur kontinuierlichen Erfassung von System-Metriken
 - > u.a. CPU, Arbeits- und Auslagerungsspeicher, Interrupts, Netzwerk
 - > aus /proc ausgelesen

SAR

- > **System Activity Report**
- > existiert schon seit UNIX System V, unter Linux Teil des sysstat-Pakets
- > Tool zur kontinuierlichen Erfassung von System-Metriken
 - > u.a. CPU, Arbeits- und Auslagerungsspeicher, Interrupts, Netzwerk
 - > aus /proc ausgelesen
- > Daten werden unterhalb /var/log gespeichert:
 - > /var/log/sa (Red Hat-artig)
 - > /var/log/sysstat (Debian-artig)
- > Metrik-Dateien pro Tag (sa02-sa30)

- > Konfigurationsdatei regelt maximale Aufbewahrung
 - > /etc/sysconfig/sysstat (Red Hat-artig)
 - > /etc/sysstat/sysstat (Debian-artig)

Direktive	Erklärung
HISTORY	Wie lange sollen Daten aufbewahrt werden?
COMPRESSAFTER	Ab wann sollen Daten komprimiert werden?
ZIP	Welches Komprimierungstool soll genutzt werden?

SAR

sar besteht aus mehreren **Programmen**:

Befehl	Erklärung
sar	zeigt aufgezeichnete Daten an
sadf	zeigt aufgezeichnete Daten, unterstützt auch CSV und XML
sadc	data collector , erfasst Daten
sa1	erfasst Daten alle 10 Minuten (via cron)
sa2	schreibt einen täglichen Report (via cron)

SAR: CPU

```
# sar
08:40:01          CPU    %user    %nice   %system   %iowait   %steal    %idle
08:50:01          all     1.69     0.01     1.21      0.07     0.01    97.02
09:00:00          all     0.16     0.00     0.24      0.00     0.00    99.60
...
Average:          all     0.29     0.04     0.35      0.01     0.01    99.30
```

Spalte	Erklärung
%user	Von Anwendungen verursachte Auslastung
%nice	Von priorisierten Anwendungen verursachte Last
%system	Systemlast (Kernel-Prozesse)
%iowait	Warten auf Erledigung von I/O
%steal	Warten auf Hypervisor (Auslastung durch andere vCPUs)
%idle	Leerlauf

SAR: ARBEITSSPEICHER

```
# sar -r
08:40:01      kbmemfree   kbavail  kbmemused  %memused  kbuffers  kbcached  [...]
08:50:01          1158812    1483380    294056      14.60      3124    419340  [...]
09:00:00          1151336    1476504    300672      14.92      3124    419936  [...]
```

Spalte	Erklärung
kbmemfree, kbmemused, kbavail	Freier/genutzter/freier Speicher in KB
%memused	Genutzter Speicher in Prozent
kbcached	Speicher in KB, der als RAM-Cache benutzt wird
kbuffers	Speicher in KB, der als Dateisystem-Cache dient
kbactive	Zuletzt aktiv genutzter Speicher in KB
kbinact	Länger nicht aktiv genutzter Speicher in KB (könnte freigegeben werden)
kbdirty	Dateimenge die noch nicht vom Dateisystem geschrieben wurde in KB

SAR: SWAP

```
# sar -S  
08:40:01  kbswpfree  kbswpused  %swpused  kbswpcad  %swpcad  
08:50:01          0          0      0.00          0      0.00  
09:00:00          0          0      0.00          0      0.00  
09:10:01          0          0      0.00          0      0.00
```

Spalte	Erklärung
kbswpfree, kbswpused	Freier/genutzter Auslagerungsspeicher in KB
%swpused	Genutzter Auslagerungsspeicher in Prozent
kbswpcad	Speicher in KB, der in Auslagerungsspeicher verschoben wurde, aber nun wieder im RAM liegt und entfernt werden kann
%swpcad	kbswpcad als Prozentwert

SAR: I/O

```
# sar -b
08:40:01      tps      rtps      wtps      dtps   bread/s   bwrtn/s   bdscd/s
08:50:01      10.15    8.19     1.96     0.00    633.96    56.09     0.00
09:00:00      0.58     0.05     0.53     0.00     1.79     9.25     0.00
09:10:01      1.46     0.77     0.70     0.00    10.17    246.81     0.00
```

Spalte	Erklärung
tps	transfers per second, alle I/O-Aufrufe an ein Speichergerät
rtps, wtps	alle Lese-/Schreibbefehle an ein Speichergerät pro Sekunde
dtps	verworfene Schreibbefehle pro Sekunde
bread/s, bwrtn/s	Insgesamt gelesene/geschriebene Daten in Blöcken (512 Bytes) pro Sekunde
bdscd/s	Insgesamt verworfene Daten in Blöcken pro Sekunde

SAR: NETZWERK

```
# sar -n DEV
10:40:00  IFACE    rxpck/s    txpck/s    rxkB/s    txkB/s    rxmcst/s    %ifutil
10:50:00    lo      0.00      0.00      0.00      0.00      0.00      0.00
10:50:00   eth0     0.82      0.22      0.05      0.02      0.00      0.00
10:50:00   eth1     0.50      0.00      0.03      0.00      0.00      0.00
```

Spalte	Erklärung
rxpck/s, txpck/s	Insgesamt empfangene/gesendete Pakete pro Sekunde
rxkB/s, txkB/s	Insgesamt empfangene/gesendete Kilobytes pro Sekunde
rxmcst/s	Empfangene Multicast -Pakete pro Sekunde
%ifutil	Auslastung einer Netzwerkkarte in Prozent

SAR

Anzeigen aller Werte und Metriken:

```
# sar -A
```

Anzeigen der Metriken zwischen 10:00 und 12:00:

```
# sar -s 10:00:00 -e 12:00:00
```

Anzeigen der RAM-Metriken des 14.Tags des Monats:

```
# sar -r -f /var/log/sa/sa14
```

LSOF

> **list open files**

> zeigt geöffnete Dateien, z.B. durch Dienste und Anwendungen

Parameter	Erklärung
<code>-u <user></code>	Beschränkung auf einen User
<code>-i, -i4, -i6</code>	Beschränkt auf IP(v4, v6)-Adressen
<code>-a</code>	Logische Verknüpfung
<code>-r <number></code>	Wiederholt alle n Sekunden
<code>-t</code>	Gibt nur eine PID-Liste aus

LSOF

Auflisten aller offenen Dateien:

```
# lsof
COMMAND      PID  USER   FD      TYPE      DEVICE SIZE/OFF NODE NAME
systemd        1  root    cwd      DIR  252,4     4096  128 /
systemd        1  root    rtd      DIR  252,4     4096  128 /
```

Auflisten aller Prozesse, die eine bestimmte Datei geöffnet haben:

```
# lsof /bin/bash
COMMAND      PID  USER   FD      TYPE      DEVICE SIZE/OFF NODE NAME
bash        11475  user  txt      REG  252,4  1389024 265438 /usr/bin/bash
bash        16640  root  txt      REG  252,4  1389024 265438 /usr/bin/bash
```

LSOF

Auflisten aller durch einen User geöffneten Dateien:

# lsof -u user						
COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF
					NODE	NAME
systemd	16847	user	cwd	DIR	252, 4	4096
systemd	16847	user	rtd	DIR	252, 4	4096

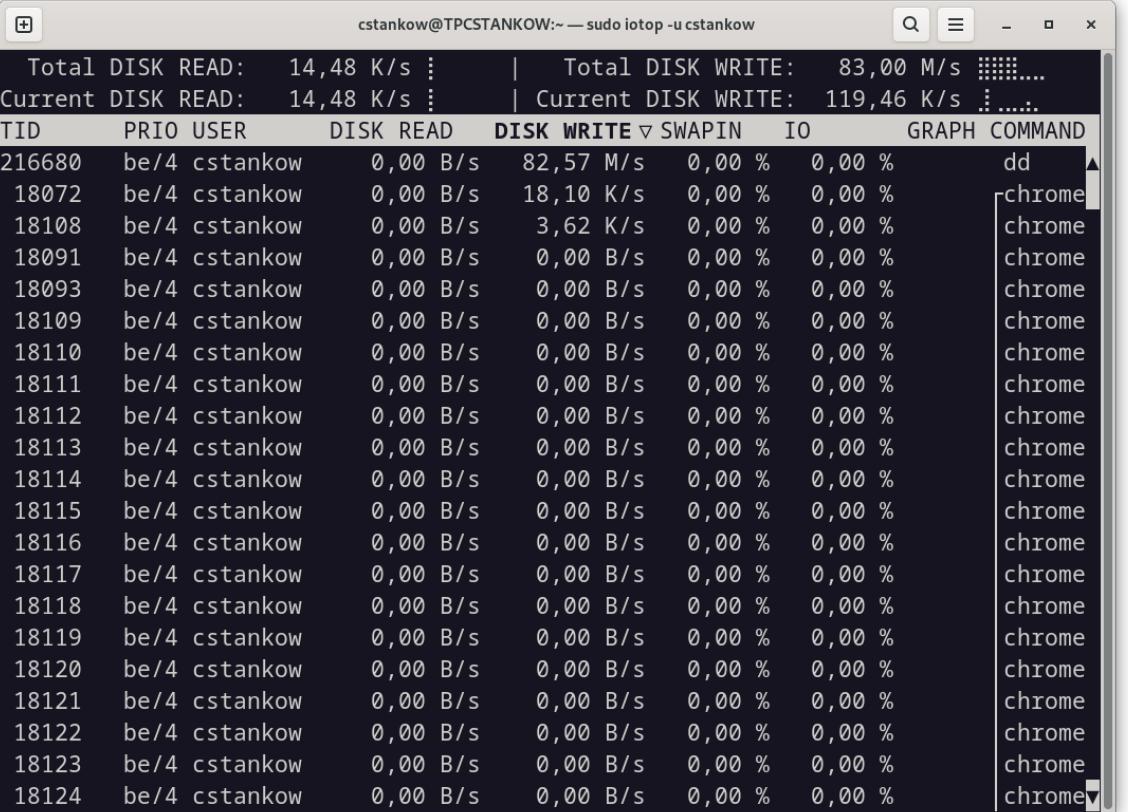
Zeigen aller offenen Ports eines Service-Users (-u und -i werden via -a kombiniert):

# lsof -a -i -u apache							
COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE NAME
httpd	792	apache	4u	IPv6	23653	0t0	TCP *:http (LISTEN)
httpd	793	apache	4u	IPv6	23653	0t0	TCP *:http (LISTEN)
httpd	794	apache	4u	IPv6	23653	0t0	TCP *:http (LISTEN)

IOTOP

- > zeigt die aktuellen I/O-Transferraten laufender Prozesse
- > wie top, nur für IO

Parameter	Erklärung
-p / --pid	auf Prozess beschränken
-u / --user	auf User beschränken
-b / --batch	nicht-interaktiver Batch-Modus



The screenshot shows the terminal window of a Linux system with the command `cstankow@TPCSTANKOW:~ — sudo iotop -u cstankow` running. The output displays disk I/O statistics for various processes. The top section shows summary statistics:

Total DISK READ:	14,48 K/s	:	Total DISK WRITE:	83,00 M/s
Current DISK READ:	14,48 K/s	:	Current DISK WRITE:	119,46 K/s

The main part of the output is a table with the following columns: TID, PRIO, USER, DISK READ, DISK WRITE, SWAPIN, IO, GRAPH, and COMMAND. The data shows that process 216680 is the primary consumer of disk bandwidth, while many other processes (18072, 18108, 18091, etc.) have near-zero activity.

TID	PRIO	USER	DISK READ	DISK WRITE	SWAPIN	IO	GRAPH	COMMAND
216680	be/4	cstankow	0,00 B/s	82,57 M/s	0,00 %	0,00 %	dd	
18072	be/4	cstankow	0,00 B/s	18,10 K/s	0,00 %	0,00 %	chrome	
18108	be/4	cstankow	0,00 B/s	3,62 K/s	0,00 %	0,00 %	chrome	
18091	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18093	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18109	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18110	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18111	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18112	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18113	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18114	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18115	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18116	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18117	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18118	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18119	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18120	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18121	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18122	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18123	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	
18124	be/4	cstankow	0,00 B/s	0,00 B/s	0,00 %	0,00 %	chrome	

IOSTAT

- > zeigt I/O-Statistiken für Geräte und Partitionen
- > Teil von sysstat

Parameter	Erklärung
-x	Erweiterte Statistiken
-d	Nur Geräte-Statistiken
-c	Nur CPU-Statistiken
-h	human-readable, schönere Formattierung und automatische Wahl der Einheit (KB, MB, etc.)

IOSTAT

Aktuelle Metriken anzeigen:

# iostat						
Linux 5.14.0-427.28.1.el9_4.x86_64 (0-node1.sva.de)						02/14/25
avg-cpu: %user %nice %system %iowait %steal %idle						
	0.23	0.02	0.35	0.01	0.01	99.39
Device	tps	kB_read/s	kB_wrtn/s	kB_dscd/s	kB_read	kB_wri
dm-0	0.00	0.07	0.00	0.00	1044	
vda	1.59	32.26	10.48	0.00	487197	1582
vdb	0.01	0.22	0.00	0.00	3376	
vdc	0.01	0.14	0.00	0.00	2092	

Metriken analog zu sar

IOSTAT

Nur Geräte-Statistiken anzeigen:

```
# iostat -d
Linux 5.14.0-427.28.1.el9_4.x86_64 (0-node1.sva.de)      02/14/25      _x86_64_
Device          tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wri
vda           1.58      32.13       10.45        0.00   487197     1584
vdb           0.01       0.22        0.00        0.00     3376        0
vdc           0.01       0.14        0.00        0.00     2092        0
```

Eine Minute lang sekündlich Statistiken ausgeben:

```
# iostat 1 60
```

VMSTAT

virtual memory statistics, zeigt Speicher-, CPU- und IO-Informationen

```
# vmstat
procs -----memory----- ---swap-- -----io---- -system-- -----cpu-----
 r b swpd   free   buff   cache   si   so    bi    bo   in   cs us sy id wa st
 0 0      0 1055852  3124 577384     0     0    16     5   56 107  0  0 99  0  0
```

Gruppe	Spalte	Erklärung
procs	r, b	runnable (könnte ausgeführt werden), blocked (warten auf IO)
memory	swpd, free, buff, cache	genutzter Swap, freier Speicher, Cache und Dateisystem-Cache
swap	si, so	von Swap gelesen, in Swap geschreiben
io	bi, bo	Von Gerät gelesene, auf Gerät geschriebene Blöcke
system	in, cs	Interrupts und Kontextwechsel pro Sekunde
cpu	us, sy, id, wa, st	user/system -Auslastung, idle , IO wait , stole durch andere VMs

MP STAT

- > zeigt CPU-Statistiken an
- > Teil von sysstat

Parameter	Erklärung
-N <number>/ALL	Beschränkung auf NUMA-Node
-o JSON	JSON-Ausgabe
-P <number>/ALL	Beschränkung auf bestimmte oder alle CPUs

Kann wiederholend ausgeführt werden, z.B. sekündlich für eine Minute:

```
# mpstat 1 60
```

MPSTAT

```
# mpstat
Linux 5.14.0-427.28.1.el9_4.x86_64 (0-node1.sva.de) 02/14/25 _x86_64_
13:20:38      CPU      %usr      %nice      %sys %iowait      %irq      %soft      %steal      %guest
13:20:38      all       0.22       0.02       0.33       0.01       0.00       0.00       0.01       0.00
```

Metriken analog zu sar, jedoch ergänzend:

Spalte	Erklärung
%irq	Zeitanteil, in welchem die CPU Hardware-Interrupts bedient
%soft	Zeitanteil, in welchem die CPU Software-Interrupts bedient
%guest	Zeitanteil, in welchem VCPUs bedient werden
%gnice	Zeitannteil für die Ausführung priorisierter Gäste

LAB Mo2

TROUBLESHOOTING EINER ANWENDUNG

- > **socket statistics**
- > Vergleichbar mit netstat, jedoch moderner und mächtiger

Parameter	Erklärung
-t, --tcp	TCP-Sockets anzeigen
-u, --udp	UDP-Sockets anzeigen
-l, --listening	Lauschende Sockets anzeigen
-p, --processes	Prozesse der Sockets anzeigen
-e, --extended	Erweiterte Infos wie UIDs anzeigen
-n, --numeric	Service-Namen nicht auflösen

Netid	State	Recv-Q	Send-Q	Local Address:Port
tcp	LISTEN	0	511	* :http

Spalte	Erklärung
Netid	Verwendetes Protokoll
State	Status des Sockets
Recv-Q, Send-Q	Empfangende/sendende Bytes, die derzeit im Puffer sind
Peer Address:Port	Adresse/Port der Gegenstelle
Process	Prozessinformationen

IPTRAF

- > TUI zur Analysieren von Netzwerkstatistiken in Echtzeit
- > zählt Pakete und Bytes
- > zeigt offene Verbindungen
- > unterstützt Protokolle, wie ARP, IP, ICMP, TCP und UDP
- > zeigt keine Paketinhalte an

```
iptraf-ng 1.2.1
TCP Connections (Source Host:Port) ━━━━━━ Packets ━━━━ Bytes ━━ Flag ━━ Iface ━━
192.168.56.1:49578 = 14 3298 --A-- eth1
192.168.56.20:80 = 8 1925 -PA- eth1

TCP: 1 entries ━━━━━━ Active ━━━━━━
ICMP echo reply (84 bytes) from 192.168.56.20 to 192.168.56.30 on eth1
ICMP echo req (84 bytes) from 192.168.56.30 to 192.168.56.20 on eth1
ICMP echo reply (84 bytes) from 192.168.56.20 to 192.168.56.30 on eth1
ICMP echo req (84 bytes) from 192.168.56.30 to 192.168.56.20 on eth1
ICMP echo reply (84 bytes) from 192.168.56.20 to 192.168.56.30 on eth1
ICMP echo req (84 bytes) from 192.168.56.30 to 192.168.56.20 on eth1
ICMP echo reply (84 bytes) from 192.168.56.20 to 192.168.56.30 on eth1
Bottom ━━━━ Time: 0:00 ━━━━ Drops: 0 ━━━━
Packets captured: 132 | TCP flow rate: 5.22 kbps
Up/Dn/PgUp/PgDn-scroll M-more TCP info W-chg actv win S-sort TCP X-exit
```

LAB Mo3

NETZWERKVERKEHR BEOBACHTEN

- > extended Berkeley Packet Filter
- > Technologie zur Ausführung von Prozessen zur **Metrikerfassung**
- > kann privilegiert auf Informationen des Kernels zugreifen
- > Anpassung des Kernels zur Laufzeit ohne Quellcode-Anpassung oder Nachladen von Modules
 - > integrierter **Verifier** verhindert Abstürze
- > Code kann durch **Hooks** auf Ereignisse reagieren
 - > wird interpretiert oder **JIT**-kompiliert
- > mächtig und komplex, nicht Scope der Schulung



ZUSAMMENFASSUNG

- > Linux bietet zahlreiche Tools zur Systemüberwachung, u.a.:
 - > Allgemein: top, htop, nmon, sar
 - > Prozesse: ps, top, htop
 - > Storage: iotop, iostat
 - > Speicher: free, top, htop, vmstat, smem, mpstat, swapon
 - > Netzwerk: netstat, ss, iptraf
- > Linux unterscheidet zwei verschiedene Speicherarten:
 - > **Physischer Speicher** = verbauter RAM mit verbindlichen Speicheradressen
 - > **Virtueller Speicher** = vom Betriebssystem abstrahierte Adressierung
- > Linux nutzt ungenutzten Speicher als Dateisystem-Cache

// GÄNGIGE SERVICES

GÄNGIGE SERVICES

Linux bietet eine breite Fülle an Diensten, u.a.:

Dienst	Beispiele
Webserver	Apache , NGINX , lighttpd , OpenLiteSpeed , Tomcat , NodeJS , Caddy
Mailserver	Postfix , Sendmail , Dovecot , Cyrus IMAPD , Fetchmail
Datenbanken	MySQL , MariaDB , PostgreSQL , Oracle Database , Redis , CouchDB
Datenaustausch	NFS , Samba , iSCSI, vsftpd , ProFTPD , TFTPD, WebDav
Proxyserver	Squid , Privoxy , Tinyproxy , Swiperproxy , Traefik , HAProxy , Varnish
VPN	OpenVPN , IPSec, WireGuard
Monitoring	Nagios , Icinga2 , Munin , check_mk , Zabbix , Elastic Stack
Management	Uyuni , Foreman , Katello , Ubuntu Landscape

APACHE

- > einer der bekanntesten und beliebtesten Webserver
- > erschien erstmalig **1995**
- > in der Programmiersprache **C** geschrieben, für viele Betriebssysteme erhältlich
- > einfach einzusetzen, vielseitig anpassbar
- > bei vielen Linux-Distributionen der Standard-Webserver
- > wird im Hosting in den letzten Jahren [von NGINX verdrängt](#)



APACHE

Über zahlreiche **Module** erweiterbar:

- > mod_ssl und mod_gnutls für TLS/SSL
- > Skriptsprachen u.a. mod_php, mod_perl, mod_python,...
- > Weiterleitungen via mod_rewrite und mod_proxy
- > Authentifizierung über zahlreiche mod_auth*-Module
- > mod_security für Web Application Firewall
- > mod_dav und mod_dav_fs für WebDAV



APACHE: INSTALLATION

Zur Inbetriebnahme genügend die folgenden Schritte:

- > Installation des entsprechenden Pakets
 - > bei den meisten Distributionen apache2, unter RHEL-artigen httpd
- > Konfigurieren der Firewall
- > Starten des Dienstes

Installation unter RHEL:

```
# dnf install httpd
# systemctl enable --now httpd
# curl http://localhost
```

LAB So1

APACHE INSTALLIEREN

APACHE: KONFIGURATION

- > Konfigurationsordner variiert je nach Distribution
 - > Red Hat: /etc/httpd
 - > die meisten anderen: /etc/apache2
- > Hauptkonfigurationsdatei* nach Möglichkeit nicht editieren
- > Unterordner und weitere Konfigurationen u.a. für
 - > Module
 - > Virtuelle Hosts und Sites

* httpd.conf (Red Hat) bzw. apache2.conf (Rest)

APACHE: KONFIGURATION

Beispiel (Ubuntu):

Ordnername	Beschreibung
conf-available	Verfügbare Konfigurationen (z.B. Logging und Security)
conf-enabled	Aktivierte Konfigurationen, Symlink zeigt auf conf-available-Datei
mods-available	Verfügbare Module (z.B. TLS/SSL)
mods-enabled	Aktivierte Module, Symlink zeigt auf mods-available-Datei
sites-available	Verfügbare Sites mit virtual Hosts
sites-enabled	Aktivierte Sites, Symlink zeigt auf sites-available-Datei

Dateiname	Beschreibung
apache2.conf	Hauptkonfiguration, lädt weitere Dateien
ports.conf	Definiert Adressen und Ports, auf denen gelauscht werden soll
envvars	Gesetzte Umgebungsvariablen

APACHE: KONFIGURATION

Beispiel (Red Hat):

Ordner-/Dateiname	Beschreibung
conf.d	Sites und weitere Konfigurationen
conf.modules.d	Modul-Konfigurationen
conf/httpd.conf	Hauptkonfiguration
modules	Moduldateien

APACHE: KONFIGURATION

Konfigurationen bestehen aus Direktiven, über 700 verfügbar:

Name	Beschreibung
ServerRoot	Haupt-Konfigurationsordner, i.d.R. /etc/apache2 oder /etc/httpd
Listen	Port(s) oder IP-Adressen, auf denen gelauscht werden soll - z.B. 80, 192.168.1.100:80
User, Group	User/Group unter welchem der Server ausgeführt wird, i.d.R. apache2
ServerAdmin	E-Mail des Administrators, z.B. root@localhost
DocumentRoot	Ordner, in welchem auszuliefernde Dateien liegen, i.d.R. /var/www/html

APACHE: KONFIGURATION

Der Syntax ist an **XML** angelehnt:

```
<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

- > Einstellungen für Aufrufe innerhalb des /var/www/html-Ordners
- > Es wurden zwei **Optionen** gesetzt:
 - > Indexes erlaubt das Auflisten und Durchsuchen von Verzeichnissen
 - > FollowSymLinks versucht symbolischen Links zu folgen
- > Berechtigungen können ordnerweise mit .htaccess-Dateien überschrieben werden
- > Zugriff auf dieses und untergeordnete Dateien und Ordner ist erlaubt

APACHE: KONFIGURATION

VirtualHosts erlauben das Betreiben mehrerer Webseiten auf einem Server:

```
Listen 80

<VirtualHost *:80>
    DocumentRoot "/var/www/html/app1"
    ServerName app1.example.com
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "/var/www/html/app2"
    ServerName app2.example.com
</VirtualHost>
```

Setzt voraus, dass die DNS-Einträge* entsprechend gesetzt sind.

* Es ist *natürlich* wieder DNS!

APACHE: INHALTE

- > Apache kann in anderen Programmiersprachen entwickelte Anwendungen bereitstellen, z.B.
 - > PHP
 - > Perl
 - > Python
- > Programmcode wird über Module i.d.R vorher **interpretiert**
 - > generierte **Ausgabe** wird dann an den Web-Browser übergeben

APACHE: PHP

Installation und Konfiguration unter Red Hat- und Debian-artigen Distributionen:

```
# dnf install php  
# systemctl restart httpd
```

```
# apt-get install libapache2-mod-php  
# systemctl restart apache2
```

Beispielhaftes PHP-Programm:

```
<?php phpinfo(); ?>
```

PHP Version 8.1.2-1ubuntu2.20



System	Linux 0-node2 5.15.0-91-generic #101-Ubuntu SMP Tue Nov 14 13:30:08 UTC 2023 x86_64
Build Date	Dec 3 2024 20:14:35
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exit.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-finfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,NTS
PHP Extension Build	API20210902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

LAB So2

PHP-ANWENDUNG INSTALLIEREN

APACHE: TLS

- > Apache unterstützt TLS zur **Verschlüsselung** übertragener Informationen
 - > selbstsignierte und offizielle Zertifikate unterstützt
- > Vorgehensweise
 - > mod_ssl aktivieren
 - > TCP-Port 443 in der Firewall öffnen
 - > VirtualHost konfigurieren
 - > **Zertifikat** und **Schlüssel** hinterlegen
 - > Konfiguration neu einlesen oder Webserver neu starten

EXKURS: TLS

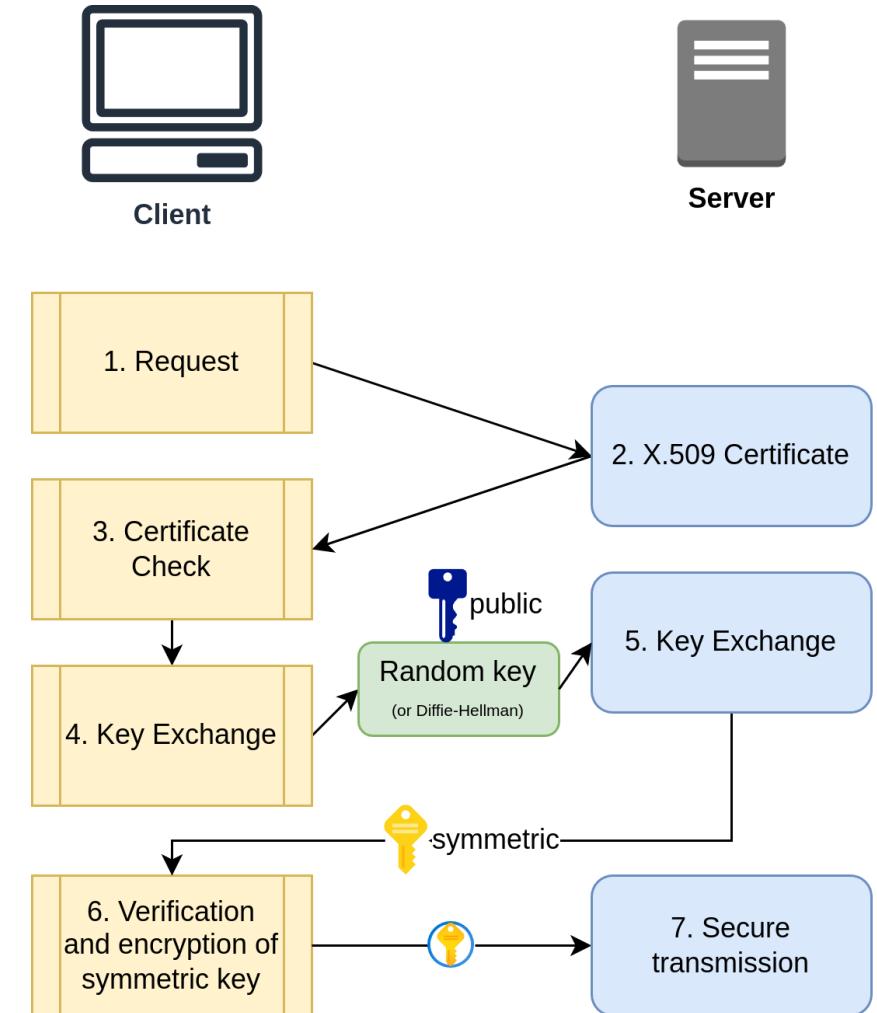
- > Transport Layer Security, früher Secure Sockets Layer
- > Verschlüsselungsprotokoll zur Datenübertragung im Internet
- > seit **1994** entwickelt, in verschiedenen **Versionen** erschienen
 - > SSL 1.0 bis 3.0 sowie TLS 1.0 und 1.1 werden nicht mehr unterstützt
 - > TLS 1.2 (2008) oder 1.3 (2018) verwenden

EXKURS: TLS

- > Transport Layer Security, früher Secure Sockets Layer
- > Verschlüsselungsprotokoll zur Datenübertragung im Internet
- > seit **1994** entwickelt, in verschiedenen **Versionen** erschienen
 - > SSL 1.0 bis 3.0 sowie TLS 1.0 und 1.1 werden nicht mehr unterstützt
 - > TLS 1.2 (2008) oder 1.3 (2018) verwenden
- > setzt auf **X.509**-Zertifikate
 - > Standard für Public-Key-Infrastruktur und digitale Zertifikate
- > Anwendung verfügt über ein **Zertifikat** sowie einen **Schlüssel**
 - > diese können über einen **Certificate Signing Request** (CSR) und einen privaten Schlüssel bei einer **Certificate Authority** (CA) angefragt werden

EXKURS: TLS

- > Server authentifiziert sich gegenüber Client mit **Zertifikat**
 - > wird i.d.R. von einer CA beglaubigt
- > Client überprüft Zertifikat auf **Plausibilität**
 - > Web-Browser vertrauen gängigen CAs und können überprüfen, ob das Zertifikat von ihnen signiert wurde
- > Client schickt Server eine mit seinem öffentlichen Schlüssel signierte **Zufallszahl**
- > Server antwortet, es wird ein kryptografischer **Schlüssel** abgeleitet
- > **Sitzung** wird mit ausgehandeltem Schlüssel verschlüsselt



MARIADB

- > freies relationales Datenbankmanagementsystem
- > 2009 von MySQL geforkt
 - > von früherem MySQL-Autor Michael Widenius
- > API-kompatibel zu MySQL
- > unterstützt zahlreiche Datentypen und Speicherformate
(Storage Engines)
- > Als Standalone-Server oder Galera-Cluster verfügbar
- > Wird häufig für Web-Anwendungen benutzt



MARIADB: INSTALLATION

```
# dnf install mariadb-server  
# systemctl enable --now mariadb-server  
  
# apt-get install mariadb-server
```

Nach der Installation empfiehlt sich eine Datenbank-Härtung:

```
# mariadb-secure-installation
```

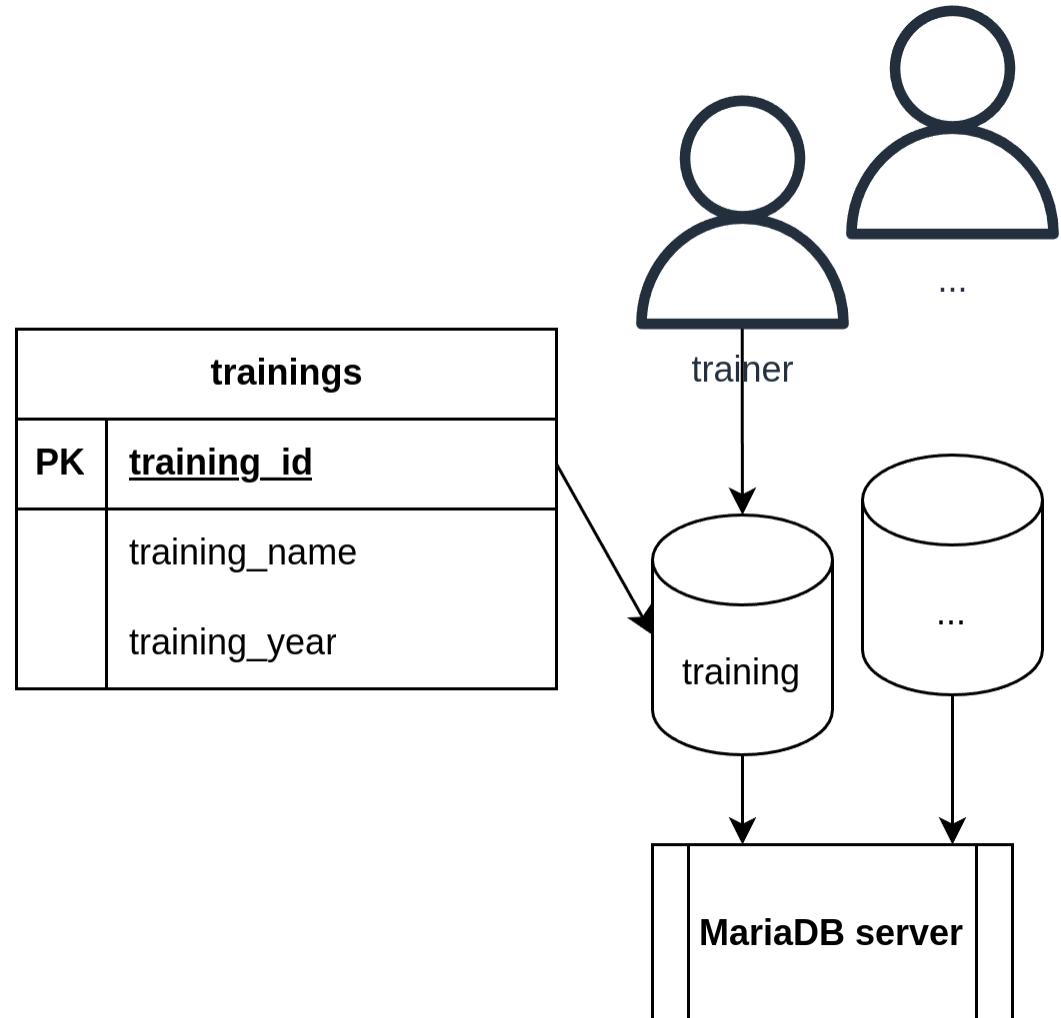
- > Setzen eines root-Kennworts
- > Entfernen von Beispiel-Inhalten
- > Remote-Zugriff erschweren

LAB So3

MARIADB INSTALLIEREN

MARIADB: KONZEPT

- > Server ist über TCP oder **Socket** erreichbar
- > User haben Zugriff auf eine oder mehrere **Datenbanken**
- > Datenbanken bestehen aus **Tabellen**
 - > Tabellen haben Spalten mit Namen und **Datentypen**
- > Definition und Auslesen mit **SQL**



MARIADB: DATENBANKEN UND USER ANLEGEN

```
# mysql -u root -p

MariaDB> CREATE USER 'sgiertz'@'localhost' IDENTIFIED BY 'fakuchad';
MariaDB> GRANT ALL PRIVILEGES ON robots.* TO 'sgiertz'@localhost;
MariaDB> CREATE DATABASE robots;
MariaDB> FLUSH PRIVILEGES;
```

```
# mysql -u sgierz -p robots
MariaDB> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| robots          |
+-----+
2 rows in set (0.001 sec)
```

MARIADB: TABELLE ANLEGEN

Eine einfache Tabelle anlegen und befüllen:

```
CREATE TABLE robots(  
    robot_id int auto_increment,  
    robot_name tinytext not null,  
    robot_purchase_date date,  
    robot_cost float,  
    PRIMARY KEY(robot_id)  
) ;
```

```
INSERT INTO robots (robot_name,  
robot_purchase_date, robot_cost)  
VALUES ('c274n', '1990-07-09',  
1337.00);
```

```
MariaDB> SELECT * FROM robots;
```

robot_id	robot_name	robot_purchase_date	robot_cost
1	c274n	1990-07-09	1337

MARIADB: BACKUP UND RESTORE

Mit `mariadb-dump` lassen sich Datenbanken sichern und wiederherstellen:

Befehl	Erklärung
<code>mariadb-dump <db></code>	Gibt den Dump einer bestimmten Datenbank aus
<code>mariadb-dump -u <user> -p <db> > backup.sql</code>	Erstellt ein Backup einer bestimmten Datenbank, User und Password benötigt
<code>mariadb-dump -u root -p --all-databases > backup.sql</code>	Sichert als Admin alle angelegten Datenbanken
<code>mariadb <db> < backup.sql</code>	Stellt eine Datenbank wieder her
<code>mariadb -u <user> -p <db> < backup.sql</code>	Stellt eine Sicherung unter Verwendung eines bestimmten Users wieder her

LAB So4

MARIADB-INHALTE VERWALTEN

SAMBA

- > seit **1992** entwickelte Suite für Windows-Interoperabilität
- > Name geht auf das Netzprotokoll **SMB*** zurück
 - > früher auch als **Common Internet File System** bekannt
 - > benutzt heutzutage den **TCP-Port 445**
- > bietet zahlreiche Funktionen an
 - > **Datenaustausch** über Netzfreigaben
 - > Druckerfreigabe
 - > Domain Controller



* **Server Message Block**

SAMBA

Überblick über bisherige SMB-Versionen:



Version	Unterstützung seit
1.0	Windows 2000
2.0	Windows Vista, Server 2008
2.1	Windows 7, Server 2008 R2
3.0	Windows 8, Server 2012
3.0.2	Windows 8.1, Server 2012 R2
3.1.1	Windows 10, Server 2016

SAMBA

- > Linux-Host kann Samba-Server oder -Client sein
 - > wird gerne für heterogene Netze genutzt
- > besteht aus verschiedenen **Komponenten**
 - > smbd (Datei-/Druckerfreigabe)
 - > nmbd (NetBIOS-Namensauflösung)
 - > winbindd (Benutzer-/Gruppenzuordnung)
 - > samba (Active Directory-Emulation)
- > je nach Konfiguration werden nicht alle verwendet



SAMBA: DATEI- UND DRUCKERFREIGABE

- > Konfigurationsdatei im **INI**-Syntax Beispielhafte **Freigabe**:
- > Freigaben werden als **Sektionen** angelegt
- > Allgemein Sektionen
 - > global - Globale Einstellungen
 - > homes - Heimatverzeichnisse
 - > printers - Druckerfreigabe

[memes]

```
comment = Memetic files
path = /memes
read only = Yes
browseable = Yes
```

SAMBA: DATEI- UND DRUCKERFREIGABE

smb.conf besteht aus zahlreichen Direktiven:

Einstellung	Erklärung
path	Pfad
read only	Nur lesen (yes, no)
writable	Schreibbare Freigabe (yes, no)
printable	Druckerfreigabe
guest ok	Anonyme Nutzung (yes, no)



Die Konfiguration sollte nach der Änderung mit testparm überprüft werden

SAMBA: DATEI- UND DRUCKERFREIGABE

- > Samba pflegt eine dedizierte Userdatenbank (**SAM**)
 - > Objekte werden auf ID-Basis voneinander unterschieden
 - > **SID** unter Windows, **UID** unter Linux/Unix
- > kann auch mit LDAP verbunden werden (`idmap`-Dienst)
- > Samba authentifiziert Sitzungen im User-Kontext
 - > Zugriff für Freigabe erfordert vorherigen Login
- > Tools
 - > `smbpasswd` - Passwörter setzen
 - > `pdbedit` - Accounts editieren

SAMBA: DATEI- UND DRUCKERFREIGABE

Befehl	Erklärung
smbpasswd -a <user>	Passwort anlegen
smbpasswd <user>	Passwort ändern
smbpasswd -d <user>	Passwort deaktivieren
smbpasswd -x <user>	Passwort löschen

Befehl	Erklärung
pdbedit -L	Alle Samba-User auflisten
pdbedit -a -u <user>	User anlegen
pdbedit -r -u <user>	User modifizieren
pdbedit -x -u <user>	User löschen

SAMBA: DATEI- UND DRUCKERFREIGABE

Mit smbclient können Samba-Server durchsucht werden:

Befehl	Erklärung
smbclient -L //<name>	Auflisten aller öffentlichen Freigaben
smbclient -U user //<name>	Auflisten aller Freigaben nach Authentifizierung
smbclient //<name>	Starten einer interaktiven Shell

Die Shell unterstützt einige Kommandos, u.a.: get, put, cd, ls, mkdir, rmdir, rm und exit.

SAMBA: DATEI- UND DRUCKERFREIGABE

Sofern das Paket `cifs-utils` installiert ist, kann mit `mount .cifs` eine Freigabe eingehängt werden:

```
# mount.cifs //node1/labs -o user=user /var/labs
```

Das Kommando akzeptiert verschiedene **Parameter**:

- > user / username
- > pass / password
- > credentials - Datei mit Zugangsdaten
- > domain, dom, workgroup - Domäne bzw.
Arbeitsgruppe
- > guest - nicht nach Zugangsdaten fragen
- > ro, rw - lesen/schreiben
- > sec - Security Mode (Passwort-Hashing)
- > vers - zu verwendende SMB-Version

Samba unterstützt immer noch SMB 1.0 - die Version sollte aus Sicherheitsgründen **nicht** benutzt werden.

SAMBA: DATEI- UND DRUCKERFREIGABE

Freigaben können auch beim Boot über einen Eintrag in der Datei `/etc/fstab` eingehängt werden:

```
# Freigabe mit Schreibzugriff  
//192.168.1.100/memes /media/Memes cifs _netdev, rw  
  
# Gastfreigabe  
//192.168.1.100/public /media/Public cifs _netdev, guest
```

Der Parameter `_netdev` gibt an, dass für die Freigabe der Netzwerkstack zur Verfügung stehen muss. Das Einhängen erfolgt also später als bei lokalen Speichern.

LAB So5

SAMBA-SERVER EINRICHTEN

LAB So6

SAMBA-CLIENT EINRICHTEN

NFS

- > Network File System
- > 1984 von SUN Microsystems entwickelt
- > Betriebssystem-unabhängiges Protokoll zur Dateienübertragung
- > sowohl unter UNIX/Linux als auch Windows benutzbar
 - > offener Standard, in zahlreichen RFCs definiert
- > Design basiert auf **RPC***
- > Gegenüber Samba authentifiziert NFS Clients statt Sitzungen

* Remote Procedure Call

NFS: VERSIONEN

Version	Jahr	Highlights
NFSv1	1984	nur intern bei SUN benutzt
NFSv2	1989	Kommunikation nur über UDP, stateless
NFSv3	1995	Support für größere Dateien und Dateisysteme, asynchroner Transfer, TCP
NFSv4	2000, 2003, 2015	Locking-, Performance- und Security-Optimierungen, Freigaben über mehrere Server (pNFS), UTF-8, nur noch TCP-/UDP-Port 2049 benötigt

 Server und Client sollten die gleiche Version einsetzen 

NFS: DATENFREIGABE

- > Freigaben werden in der Datei `/etc/exports` definiert
- > Einträge haben **zwei** Spalten
 - > Dateisystem-Pfad
 - > Zielsystem mit **Optionen**
- > Das System kann in verschiedenen **Formaten** angegeben werden:
 - > Hostname
 - > IP-Adresse
 - > Netzadresse
 - > Wildcard

NFS: DATENFREIGABE

Beispiele:

```
# Lesender Zugriff für ein System  
/mnt/memes          node1(ro)
```

```
# Schreibender Zugriff für ein System  
/mnt/music          192.168.1.100(rw)
```

```
# Lesender Zugriff für ein Netz  
/mnt/templates      192.168.1.0/24(ro)
```

```
# Lesender Zugriff für alle Systeme einer Domain  
/mnt/docs          *.evilcorp.lan(ro)
```

NFS: DATENFREIGABE

Zu den denkbaren Optionen gehören u.a.:

Option	Erklärung
rw, ro	Schreibender/lesender Zugriff
sync, async	Synchroner/asynchroner Zugriff
root_squash, no_root_squash	root-User zu nfsnobody transferieren/nicht transferieren
anonuid, anongid	Anonymen User ersetzen
nfsvers	Verwendete NFS-Version (3 oder 4)

NFS: DATENFREIGABE

Freigaben werden nach Änderung der Konfiguration wie folgt aktiviert:

```
# exportfs -ra
```

Auf einem Client-System erfolgt der Zugriff über `mount.nfs` bzw. `mount.nfs4`:

```
# mount.nfs4 node1:/memes /import/memes
```

NFS: DATENFREIGABE

Freigaben können auch beim Boot über einen Eintrag in der Datei `/etc/fstab` eingehängt werden:

```
# Freigabe mit Schreibzugriff
node1:192.168.1.100/memes          /media/Memes   cifs   _netdev, rw

# Gastfreigabe
node2:192.168.1.100/public         /media/Public  cifs   _netdev, ro
```

Der Parameter `_netdev` gibt an, dass für die Freigabe der Netzwerkstack zur Verfügung stehen muss. Das Einhängen erfolgt also später als bei lokalen Speichern.

MAILSERVER

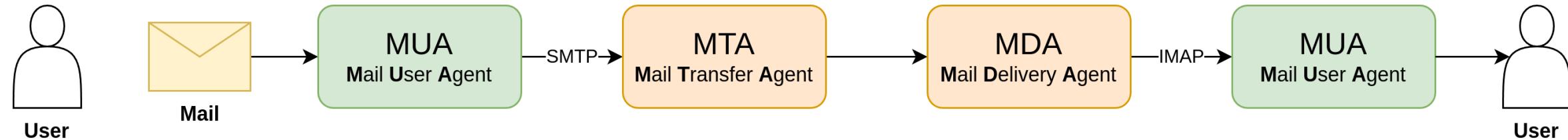
- > Linux eignet sich auch hervorragend als Mailserver
 - > würde jedoch den Rahmen dieser Schulung sprengen
- > Breite Auswahl an Programmen
 - > [Postfix](#), [Exim](#) (SMTP)
 - > [Dovecot](#) (IMAP, POP3)
 - > [Cyrus](#) (IMAP)
 - > [Apache SpamAssassin](#) (Anti-Spam)
 - > [ClamAV](#) (Antivir, kann in Mails integriert werden)
 - > [Citadel](#), [EGroupware](#) (Groupware)

MAILSERVER: BEGRIFFLICHKEITEN

- > **MTA - Mail Transfer Agent**
 - > SMTP-Server, versendet Mail zu anderem Mailserver
- > **MDA - Mail Delivery Agent**
 - > IMAP-Server, hält Mails vor und stellt diese Usern zu

MAILSERVER: BEGRIFFLICHKEITEN

- > **MTA - Mail Transfer Agent**
 - > SMTP-Server, versendet Mail zu anderem Mailserver
- > **MDA - Mail Delivery Agent**
 - > IMAP-Server, hält Mails vor und stellt diese Usern zu
- > **MUA - Mail User Agent**
 - > Programm zum Lesen, Schreiben, Senden und Empfangen von E-Mails
- > **MRA - Mail Retrieval Agent**
 - > Programm zum Empfangen von Mails, Mails werden i.d.R. lokal vorgehalten
 - > in Mail-Programmen enthalten



ZUSAMMENFASSUNG

- > Linux bietet eine breite Fülle an **Diensten**
 - > u.a Web-/Mail-/Proxy-Server, Datenbanken, Monitoring und VPN
- > **Apache** ist einer bekanntesten Webserver
 - > durch modulares Design erweiterbar, z.B um **Skriptsprachen**
 - > VirtualHosts erlauben das parallele Hosten mehrere Webseiten

ZUSAMMENFASSUNG

- > Linux bietet eine breite Fülle an **Diensten**
 - > u.a Web-/Mail-/Proxy-Server, Datenbanken, Monitoring und VPN
- > **Apache** ist einer bekanntesten Webserver
 - > durch modulares Design erweiterbar, z.B um **Skriptsprachen**
 - > VirtualHosts erlauben das parallele Hosten mehrere Webseiten
- > **MariaDB** ist ein relationales Datenbankmanagementsystem
 - > weit verbreitet, wird häufig für Web-Anwendungen genutzt
- > **Samba** ist eine Suite für Windows-Interoperabilität
 - > bietet u.a Datenaustausch- und Domain Controller-Dienste
- > **NFS** erlaubt Betriebssystem-unabhängige Dateiübertragungen
- > Zahlreiche Server-Komponenten unterstützen beim Versenden und Zustellen von **Mails**

// LINKS

- > Linux Performance-Tools (Brendan Gregg):
<https://www.brendangregg.com/linuxperf.html>
- > Linux ate my RAM: <https://www.linuxatemyram.com/>