

Redhat Grundlagen

Agenda

1. Distributionen
 - [Überblick](#)
2. Verzeichnisse und Dateitypen
 - [Verzeichnisaufbau](#)
 - [Dateitypen](#)
3. Basisbefehle
 - [In den Root-Benutzer wechseln](#)
 - [Wo bin ich ?](#)
 - [Praktische Ausgabe von langen Seiten - less](#)
 - [Datei anlegen - touch](#)
 - [Autovervollständigen * und tab](#)
 - [Welches Programm wird verwendet](#)
4. Erweiterte Befehle (Nice to have)
 - [Alias Befehle anzeigen](#)
 - [Welche Bibliotheken verwendet ein ausführbares Programm](#)
 - [Ist ein Befehl extern, alias oder intern](#)
 - [History verwenden](#)
5. Dateien und Verzeichnisse
 - [Mit cd im System navigieren](#)
 - [Verzeichnisse in Listenansicht mit versteckten Dateien anzeigen -> ls -la](#)
 - [Inhalt in Datei schreiben und anhängen](#)
 - [Verzeichnisse anlegen](#)
 - [Verzeichnisse und Dateien löschen](#)
 - [Kopieren/Verschieben/Umbenennen von Dateien und Files](#)
 - [Arbeiten mit vi](#)
6. Dateimanipulation/Unix Tools
 - [Anfang oder Ende einer Datei/Ausgabe anzeigen](#)
 - [cat/head/tail-Beginn/Ende einer Datei anzeigen](#)
 - [zcat - Inhalte einer mit gzip komprimierten Datei anzeigen](#)
 - [wc - Zeilen zählen](#)
 - [Bestimmte Zeilen aus Datei anzeigen - grep](#)
 - [Erweiterte Suche mit Grep](#)
 - [Finden von files nach Kriterien - find](#)
 - [Doppelte Leerzeichen aus Zeile rauslösch](#)
 - [Spalten auslesen mit awk](#)
 - [Strings in datei ersetzen mit sed](#)
7. Backups
 - [Backup / Restore with tar](#)
 - [Backup with tar / zstd](#)
8. Komprimierung
 - [Datei komprimieren](#)
9. Prozesse
 - [Prozesse interaktiv mit top anzeigen](#)
 - [Prozesse anzeigen - ps/pstree -p](#)
 - [Alle Prozesse eines Dienstes anzeigen](#)
 - [Prozesse im Hintergrund laufen lassen](#)
10. Festplattenbelegung

- [Welche Verzeichnisse / Partitionen sind voll? du und df](#)
- 11. Benutzer, Gruppen und Rechte
 - [Rechte/Benutzer/Gruppe](#)
 - [Dateien für Benutzer und Gruppen](#)
 - [Benutzer anlegen](#)
 - [sudo Benutzer erstellen](#)
- 12. Logs/Loganalyse
 - [Logfile beobachten](#)
 - [Dienste debuggen](#)
 - [Rsyslog](#)
 - [Journal analysieren](#)
 - [Logrotate](#)
- 13. Dienste debuggen
 - [Dienste debuggen](#)
- 14. Variablen
 - [Setzen und verwenden von Variablen](#)
- 15. Dienste/Runlevel(Targets verwalten)
 - [Die wichtigsten systemctl/service](#)
 - [Systemctl - timers](#)
 - [Systemctl - timer example](#)
 - [Gegenüberstellung service etc/init.d/ systemctl](#)
 - [Default Editor systemctl setzen](#)
- 16. Systemd
 - [Die wichtigen Tools für die Kommandozeile \(ctl\)](#)
- 17. Systemadministration
 - [Hostname setzen/abfragen](#)
 - [ssh absichern](#)
- 18. Partitionierung und Filesystem
 - [parted and mkfs.ext4](#)
- 19. Boot-Prozess und Kernel
 - [Grub konfigurieren](#)
 - [Kernel-Version anzeigen](#)
 - [Kernel-Module laden/entladen/zeigen](#)
- 20. Hilfe
 - [Hilfe zu Befehlen](#)
- 21. Grafische Oberfläche und Installation
 - [X-Server - Ausgabe auf Windows umleiten](#)
 - [Installations-Images-Server](#)
- 22. Wartung und Aktualisierung
 - [Aktualisierung des Systems](#)
 - [Paketmanager dnf](#)
 - [Archive runterladen und entpacken](#)
 - [Apache installieren \(firewall und selinux\)](#)
 - [mbr sichern mit dd](#)
- 23. Firewall und ports
 - [firewalld](#)
 - [Scannen und Überprüfen mit telnet/nmap](#)
- 24. Netzwerk/Dienste
 - [IP-Adresse von DHCP-Server holen \(quick-and-dirty\)](#)
 - [Auf welchen Ports lauscht mein Server](#)

- [Interface mit nmtu-edit verwalten - schneller Weg](#)
- [Netzwerkinterface auf der Kommandozeile einrichten](#)
- [Scannen mit nmap](#)

25. Mails

- [lokale Mails installieren](#)

26. Bash/Bash-Scripting

- [Einfaches Script zur Datumsausgabe](#)
- [Ausführen/Verkettung von mehreren Befehlen](#)
- [Example with date and if](#)
- [Example with function and return value](#)
- [Example with test and if](#)
- [Example log function](#)
- [Example Parameter auslesen](#)

27. Timers/cronjobs

- [Cronjob - hourly einrichten](#)
- [cronjob \(zentral\) - crond](#)

28. Literatur

- [Literatur](#)

Backlog

1. Wartung und Aktualisierung

- [Paketmanager yum](#)

2. Firewall

- [Arbeiten mit firewalld](#)

3. Podman

- [Podman Walkthrough](#)

4. SELinux (Linux härten)

- [SELinux](#)

5. Tools/Verschiedens

- [Remote Desktop für Linux / durch Teilnehmer getestet](#)
- [Warum umask 022 und 0002 ? - Geschichte](#)

Distributionen

Überblick

Multi-Purpose - Distributionen (Ideal zum Starten)

Redhat-Familie

```
Centos
Redhat.  - rpm / (yum / dnf)
Oracle Linux
Fedora

## ab 2022 kann man Centos Linux nicht als vollwertigen Ersatz für Redhat verwenden

## ab 2022 sehr interessant bzw. unabdingbar
Rocky Linux
Alma Linux
```

Debian Familie

```
Debian
Ubuntu. - dpkg / apt
Mint
```

SuSE - Familie

```
SLES (SuSE Linux Enterprise)
OpenSuSE
```

Distris zur Sicherheitsüberprüfung / Hacken

```
Kali Linux
Parrot.  - Distributionen zum Hacken
```

Live-DVD (Linux ohne Installation)

- Knoppix - Live DVD - brauche nicht installieren

Spezial-Linuxe, z.B. für Router

```
OpenWRT
DDWRT
```

Seite mit Übersicht aller Linux-Distros

- <https://distrowatch.com/>

Verzeichnisse und Dateitypen

Verzeichnisaufbau

/etc

- Verzeichnis für Konfigurationsdateien

/dev

- Devices (Alle Gerätedateien - Ein- und Ausgabegeräte, wie bspw. Festplatten, Mouse)

/mnt

- früher viel verwendet:
- für händisches Einhängen gedacht (per Hand mounten)

/media

- das neue / moderne (wird heutzutage meistens verwendet)
- Verzeichnis für automatisch eingehängte Devices (z.B. usb-stick)

/opt

- Große Softwarepaket (z.B. LibreOffice, OpenOffice, Dritt-Anbieter)

/boot

- Files for booting (e.g. kernel, grub.cfg, initial ramdisk)

/proc

- Schnittstelle zwischen Kernel und User-Space (für Programme, Benutzer)
- Kommunikation erfolgt über Dateien

/root

- Heimatverzeichnis des root-Benutzers

/run

- Dateien mit Prozess-ID für laufenden Services
- um diese gut beenden zu können

/tmp

- Temporäre Dateien
- Löschen von Dateien kann unter /etc/tmpfiles.d verwaltet werden (erfolgt von systemd auf Tagesbasis)

/sys

- wie proc
- Schnittstelle zwischen Kernel und User-space

/var (=variable daten)

- Hier liegen Daten, die sich häufig ändern
- Log-Dateien, Datenbanken, Spool-Dateien, Cache-Dateien

/lib

- Bibliotheken (.so, .ko) wie unter Windows *dll's

/sbin

- Programme zur Systemadministration

/bin

- Normale Programme für alle (executables)

Dateitypen

Wo ?

- Erste Spalte bei ls -la

Welche ?

```
- file
d directory
l symbolischer Link
c Character-Device (Eingabegerät: Zeichenorientiert z.B. Tastatur)
b Block-Device (Ausgabegerät): Blockorientiert, z.B. Festplatte)
s socket (Für Kommunikation von client zu server / server zu client) auf der gleichen
Maschine
```

Basisbefehle

In den Root-Benutzer wechseln

```
## einloggen als normaler Benutzer z.B. benutzer: kurs (wenn ich unter kurs eingeloggt bin)
sudo su -
## eingeben des Passworts des Benutzers
```

Wo bin ich ?

```
## 1. Ich erkenne es am prompt (Beginn der Zeile )

## pwd - Print working directory
pwd
```

Praktische Ausgabe von langen Seiten - less

Open a file with less

```
##
less /etc/services

## Why ?
## Leichtere Navigation
```

Pipen mit less (ausgabe an less schicken)

```
ls -la | less
cat /etc/services | less
```

Suchen in less

```
##Innerhalb von less
/suchbegriff + RETURN
## nächstes Suchergebnis
n
## voriger Suchergebnis
N
```

Springen ans Ende/an den Anfang

```
## Innerhalb von less
## ans Ende
G
## an den Anfang
```

```
1g
## zu einer bestimmten Zeile (Zeile 5)
5g
```

In die Hilfe rein

```
h
## wieder raus
q
```

Datei anlegen - touch

```
touch dateiname
```

Autovervollständigen * und tab

Autovervollständigen *

```
## show all entries in directory starting with tod
## * = zero or more characters
echo tod*
## tod todo todotext
```

Autovervollständigen tab

```
echo tod <TAB><TAB> # bei mehreren Einträgen
echo todol<TAB> # bei einem weiteren Eintrag
```

Welches Programm wird verwendet

```
## Sucht in der Pfad-Variablen $PATH nach dem programm
## und zeigt ersten Fund --> d.h. dieses Programm würde ausgeführt
which false
```

Erweiterte Befehle (Nice to have)

Alias Befehle anzeigen

Alias anzeigen

```
## keine wirkliche Befehle, sondern nur andere Schreibweise/Abkürzungen
## kann u.U. so auf anderen Distributions nicht vorhanden sein
alias
```

Alias - Befehl in der Session setzen

```
## Achtung, existiert nicht nach Schließen der Session
alias l3='ls -la | head -n 3'
```

Alias-Befehl aufheben/löschen (unalias)

```
unalias l3
```

Welche Bibliotheken verwendet ein ausführbares Programm

```
ldd /usr/bin/ls
```

Ist ein Befehl extern, alias oder intern

```
type cd
type echo
type ls
type find
```

History verwenden

history

```
## Zeigt die letzten 100 Befehle an
history
```

```
## !<Zahl>
## führt Befehl 24 aus der history nochmal aus
!24
```

```
<ArrowUp - Pfeile nach oben>
Zeigt den letzten Befehl aus der history

## auch mehrmals möglich
```

Special force

```
## führt den letzten Befehl aus der history aus
!!

## kombinierbar: macht ein sudo mit den letzten Befehl aus der history
sudo !!
```

Reverse Search

```
## Rückwärtssuche über alle Einträge
## STRG + r
```

Dateien und Verzeichnisse

Mit cd im System navigieren

Ins Heimatverzeichnis und Wurzelverzeichnis (C: unter Windows) wechseln

```
## Ins Heimatverzeichnis wechseln
## cd ohne alles
cd

## Ins Wurzelverzeichnis des Filesystems wechseln // Windows -> C:\
cd /
```


Wie in ein Verzeichnis wechseln (relativ und absolut)

```
## relativ - nur in ein Unterverzeichnis meines bestehenden Verzeichnisses
cd etc

## absolut - wechselt dort rein, egal wo ich bin
cd /etc
```

Ins alte Verzeichnis wechseln

```
cd /var/log
cd /etc
## Wechselt in /var/log zurück
cd -
```

Verzeichnisse in Listenansicht mit versteckten Dateien anzeigen -> ls -la

```
ls -la
```

Inhalt in Datei schreiben und anhängen

Inhalte in Datei schreiben / anhängen

```
cd /home/kurs
## Alternative 1
## cd # wechselt auch ins Heimatverzeichnis
## Alternative 2
## cd ~

## eingefügt am anfang, überschreibt alte Inhalte
ls -la > todo
## angehängt
echo "hans hat durst" >> todo
```

Verzeichnisse anlegen

Einzelne Verzeichnisse anlegen

```
## Verzeichnis Dokumente anlegen im aktuellen Verzeichnis
cd
mkdir dokumente

## absolut verzeichnis anlegen
## Wird dann im Wurzelverzeichnis angelegt als root
## als kurs-benutzer hätte ich dort keine Berechtigung
sudo mkdir /docs
```

Verzeichnisstruktur anlegen

```
cd
## Elternverzeichnisse werden automatisch angelegt
mkdir -p dokumente/projekt/plan
```

Verzeichnisstruktur anzeigen

```
sudo apt install tree
tree dokumente

## or /etc
tree /etc | less
```

Verzeichnisse und Dateien löschen

Dateien und Verzeichnisse löschen

```
## bei symbolischen Links wird nur der symbolische Link und nicht die Datei gelöscht
rm symlink
## Datei löschen
rm dateiname
## Verzeichnis löschen
rm -r verzeichnis
```

Mehrere Dateien löschen

```
cd
touch datei1 datei2 datei3
echo datei*
rm datei*
```

Symbolische Links löschen (Verhalten)

```
cd
touch woche.txt
ln -s woche.txt woche1.txt
## file woche.txt is still present
rm woche1.txt
ls -la
## Symbolischen Link erneut setzen
ln -s woche.txt woche1.txt
## Symbolischer Link danach kaputt
rm woche.txt
ls -la
## woche1.txt nicht aufrufbar, da der symbolische Link ins Leere zeigt.
cat woche1.txt
```

Kopieren/Verschieben/Umbenennen von Dateien und Files

Dateien umbenennen, verschieben, kopieren

```
## wenn Zielverzeichnis nicht existiert -> Fehler !
cp -a todo.txt /dokumente/
## wenn Zielverzeichnis nicht existiert, wird dokumente2 erstellt als file - > Achtung !!
cp -a todo.txt /dokumente2

## umbenennen
mv datei1 neuernamedatei1
```

```
## verschieben in Verzeichnis
mv datei1 /dokumente/
## besser als:
## mv datei1 /dokumente
## weil hier die Datei dokumente angelegt wird, wenn der Ordner /dokumente nicht existiert !!
```

Rechte behalten bei kopieren

```
## -a macht das
cp -a todo.txt todoneu.txt

## ohne -a werden symbolische links aufgelöst und die Rechte des ausführenden Nutzers gesetzt
cp ab cd

## Verzeichnisse kopieren
cp -a /etc /etc3
```

Arbeiten mit vi

Zeilennummern aktivieren für alle

```
## Centos
##/etc/vimrc
## am ende
set number

## Ubuntu
## /etc/vim/vimrc.local
set number
```

vimtutor

```
## Interactives Tutorial zum Lernen von vi
## Wichtigste Befehle
vimtutor # sollte bereits mit vi installiert worden sein.
```

Wichtigste Aktionen

1. # Öffnen einer neuer Datei mit vi
vi dateiname
2. # Schreiben in der Datei
i # drücken
3. # Es erscheint unten in der Zeile
-- INSERT --
4. # Nun können Sie etwas hineinschreiben
- 5a. Beenden ohne Speichern (wenn geänderter Inhalt vorhanden ist
ESC + :q! # ESC Taste drücken, dann : und q! und enter
- 5b. Oder: Speichern und schliessen
ESC + :x # ESC Taste drücken, dann : und x und enter

```
# oder  
ESC + :wq # ESC Taste drücken, dann : und w und q
```

Virtual Mode

```
v Zeichenweise markieren einschalten  
V Zeilenweise markieren einschalten  
STRG + v Blockweise markieren  
  
## mit Cursortasten auswählen / markieren  
## Dann:  
x # Löschen des markierten Bereichs
```

Zeilen löschen im Normalmodus (Interactiver Modus)

```
ESC + dd # eine Zeile löschen  
## letzte Aktion rückgängig machen  
ESC + u # eigentlich reicht 1x Escape  
## mehrere Zeilen löschen z.B. 1000  
ESC + 1000dd # ESC - Taste drücken, dann 1000 eingeben, dann dd (sie sehen die 1000 nicht auf dem Bildschirm)
```

Neues Fenster und Fenster wechseln

```
## innerhalb von vi  
ESC + : -> vsplit # aktuelles Fenster wird kopiert  
## Fenster wechseln  
ESC + : wincmd w  
## oder  
STRG + w w
```

Cheatsheet

http://www.atmos.albany.edu/daes/atmclasses/atm350/vi_cheat_sheet.pdf

Dateimanipulation/Unix Tools

Anfang oder Ende einer Datei/Ausgabe anzeigen

Die ersten 10

```
## die ersten 10 Zeilen einer Datei anzeigen  
head /etc/services  
## Alternative 1  
cat /etc/services | head  
  
## die letzten 10 Zeilen  
tail /etc/services  
cat /etc/services | tail  
  
## einer ausgabe // erste 10 Zeilen eines Verzeichnislistings  
ls -la | head
```

Die ersten 20

```
head -n 20 /etc/services
head -n20 /etc/services
head -20 /etc/services
head --lines=20 /etc/services
```

Die letzten 20

```
tail -n 20 /etc/services
tail -n20 /etc/services
tail -20 /etc/services
tail --lines=20 /etc/services
```

cat/head/tail-Beginn/Ende einer Datei anzeigen

cat mit Zeilennummer

```
cat -n /etc/services
```

Die ersten -x Zeilen anzeigen

```
## ersten 10 Zeilen anzeigen
head /etc/services

## Ersten 20 Zeilen
head -n 20 /etc/services
```

Die ersten 10 Zeilen / Variante mit cat

```
cat services | head
## mit zeilennummen
cat -n services | head
```

Die letzten -x Zeilen anzeigen

```
## die letzten 10 Zeilen
tail /etc/services

## die letzten 40 Zeilen
tail -n 40 /etc/services
```

Ausgabe der letzten 10 Zeilen

```
cat /etc/services | tail
```

zcat - Inhalte einer mit gzip komprimierten Datei anzeigen

wc - Zeilen zählen

Datei

```
wc -l /etc/services
```

Zeilen aus Befehl

```
ls -la | wc -l
```

Bestimmte Zeilen aus Datei anzeigen - grep

Beispiele

```
## alle Zeilen in den tcp vorkommt
cat /etc/services | grep tcp
## alle Zeilen in denen tcp nicht vorkommt
cat /etc/services | grep -v tcp
## alle Zeilen in denen tcp nicht vorkommt
## egal ob gross oder klein geschrieben.
cat /etc/services | grep -iv TCP

cat /etc/services | grep '#'
cat /etc/services | grep "#"
cat /etc/services | grep "^#"
## alle Zeilen, die am Anfang der Zeile kein # haben
cat /etc/services | grep -v "^#"
cat /etc/services | grep -v "^#" > /root/services
cat /etc/services | grep -v "^#" | head -n 20

cat /etc/services | grep -v "s$"
## alle Zeilen die als letztes Zeichen ein s haben
cat /etc/services | grep "s$"
```

Recursive Suchen (grep -r) - Schweizer Taschenmesser

```
grep -r "PermitRootLogin" /etc
grep -irn "PermitRootLogin" /etc
```

Erweiterte Suche mit Grep

Nach einzelnen Wort suchen (Wort muss so vorkommen)

```
cat /etc/services | grep -i -w 'protocol'
```

Eines der Begriffe soll vorkommen

```
## Achtung, unbedingt -E für extended regex verwendet
cat /etc/services | grep -E 'protocol|mysql'
```

Eines der Wort soll am Anfang der Zeile vorkommen

```
## egrep ist das gleiche wie grep -E
egrep -i '^(mysql|Maira)' /etc/services
```

x-Zeilen vor bzw. nach "Finde-(Grep-)" - Ergebnis anzeigen

```
## -A x-Zeilen danach, z.B. -A 4 --> 4 Zeilen danach
## -B x-Zeilen davor
```

```
egrep -A 4 -B 4 -i '^(mysql|Maira)' /etc/services '^(mysql|Maira)' /etc/services
```

Einzelne Zeichen als Suchmuster nehmen

```
## 0, dann zwei beliebige Zeichen, dann tcp
grep '0..tcp' /etc/services
## 0, dann ein beliebiges Zeichen, dann tcp
grep '0.tcp' /etc/services
```

Tatsächlich eine Punkt suchen

```
## /root/dateinamen
hans.txt
hans1txt
peter.txt

grep 'hans\.txt' /root/dateinamen

root@ubuntu2004-101:/etc# grep 'hans\.txt' /root/dateinamen
hans.txt
root@ubuntu2004-101:/etc# grep 'hans.txt' /root/dateinamen
hans.txt
hans1txt
```

Einzelne Zeichen sollen vorkommen

```
root@ubuntu2004-101:~# echo "Klaus" >> /root/namen
root@ubuntu2004-101:~# echo "klaus" >> /root/namen
root@ubuntu2004-101:~# grep '[kK]l' /root/namen
Klaus
klaus
root@ubuntu2004-101:~# grep '[kK][la]' /root/namen
Klaus
klaus
root@ubuntu2004-101:~# echo "karin" >> /root/namen
root@ubuntu2004-101:~# grep '[kK][la]' /root/namen
Klaus
klaus
karin

echo "Klaus1" >> /root/namen
root@ubuntu2004-101:~# echo "Klaus2" >> /root/namen
root@ubuntu2004-101:~# grep '[kK][la]aus[0-9]' /root/namen
```

Mengeangabe

```
## Achtung unbedingt egrep oder grep -E verwenden
cat /root/namen
AxB nix
AxB nix
abc nix
a nix
```

```
egrep '^[a-zA-Z]{1,3} nix' /root/namen
```

```
echo "ab nix" >> /root/namen
## Mindestens 2 Zeichen
root@ubuntu2004-101:~# egrep '^[a-zA-Z]{2,} nix' /root/namen
AxB nix
AxB nix
abc nix
ab nix
```

Nach Zahlen Suchen

```
echo "12345 namen" >> /root/namen
grep "[[:digit:]]\{5\}" /root/namen
```

Cheatsheets

- <https://cheatography.com/tme520/cheat-sheets/grep-english/>

Ref:

- <https://www.cyberciti.biz/faq/grep-regular-expressions/>

Finden von files nach Kriterien - find

Simple find command

```
## find directories with specific name
find / -name tmpfiles.d -type d
```

Doppelte Leerzeichen aus Zeile rauslöschen

Beispiel: mehrere Leerzeichen rauslöschen (eines soll verbleiben)

```
cat /etc/services | tr -s ' '
```

Spalten auslesen mit awk

/etc/passwd

```
## use ':' as separator
cat /etc/passwd | awk -F: '{print $6 ":" $1}'
```

Strings in datei ersetzen mit sed

```
## good test it before on stdout without inplace param -> -i
cp -a /etc/ssh/sshd_config /etc/ssh/sshd_config_test
sed "s/PasswordAuthentication no/PasswordAuthentication yes/g" /etc/ssh/sshd_config_test |
less
## then in place
sed -i "s/PasswordAuthentication no/PasswordAuthentication yes/g" /etc/ssh/sshd_config_test
```

Backups

Backup / Restore with tar

Sichern / Backup

```
cd /usr/src
tar cfvz _etc.20220617.tar.gz /etc
## war das archivieren erfolgreich // dann 0
echo $?
tar tf _etc.20220617.tar.gz
```

Entpacken (Vorbereitung)

```
mkdir foo
mv _etc.20220617.tar.gz foo
cd foo
```

Entpacken (Variante 1)

```
tar xvf _etc.20220617.tar.gz

## Aufräumen
rm -fR etc/
```

Entpacken (Variante 2)

```
tar tf _etc.20220617.tar.gz

## Achtung Fehler - weil falscher Pfad
tar xvf _etc.20220617.tar.gz etc/sysctl.d/99-sysctl.conf /etc/services
echo $?

## So geht's
tar xvf _etc.20220617.tar.gz etc/sysctl.d/99-sysctl.conf etc/services
ls -la
```

Entpacken (Variante 3) - direkt in ein bestimmtes Verzeichniss

```
tar tf _etc.20220617.tar.gz
tar xvf _etc.20220617.tar.gz -C / etc/sysctl.d/99-sysctl.conf etc/services
```

Referenz:

- <https://linuxconfig.org/how-to-create-incremental-and-differential-backups-with-tar>

Backup with tar / zstd

Simple with default compression (3)

```
tar cvf _etc.foobar.tar.zst --zstd /etc
```

Use higher compression

```
## send result of tar to stdout
tar cvf - /etc | zstd -15 > _etc.mega.tar.zst
```

Komprimierung

Datei komprimieren

zcat

```
zcat /var/log/syslog.2.gz
```

Übung komprimieren

```
Phase 1: Komprimieren

cd /var/log
cat messages | grep 'Nov 13' | gzip
cat messages | grep '^Nov 13' | gzip > /usr/src/messages.gz

Phase 2: Entpacken
gzip -d messages.gz
## or
gunzip messages.gz
```

Prozesse

Prozesse interaktiv mit top anzeigen

```
top
```

Prozesse anzeigen - ps/pstree -p

Prozesse anzeigen

```
ps -ef
ps aux # x alle Prozesse anzeigen, die nicht an ein Terminal gebunden sind
```

systemctl (läuft Dienst)

```
systemctl status sshd
```

Prozeßbaum anzeigen (meist nicht für die Praxis notwendig)

```
pstree -p
```

Alle Prozesse eines Dienstes anzeigen

Show all mysql processes

```
## inkl header - 2 Befehle getrennt durch ';'
ps aux | head -n 1; ps aux | grep mysqld | grep -v 'grep'

### Ausgabe
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
```

```
mysql      16938  0.0  1.1 1778456 94776 ?        Ssl  09:51   0:00 /usr/libexec/mysqld --
basedir=/usr
```

Show all ssh-processes

```
ps -efaxo user,pid,ppid,cmd | grep ssh
```

```
ps -efaxo user,pid,ppid,cmd | head -n1; ps -efaxo user,pid,ppid,cmd | grep ssh
```

```
### Prozesse im Hintergrund laufen lassen
```

```
### Version with & (background task)
```

```
#### Step 1: Create and run script
```

```
cd /usr/local/bin nano endless.sh
```

```
let i=0 while true do let i=i+1 echo $(date)"..jetzt.."${i} >> /var/log/endlosscheife.log sleep 2
```

```
done
```

```
chmod u+x endless.sh
```

```
#### Step 2: Script starten und beenden
```

```
endless.sh
```

beend

STRG + c

```
#### Step 3: Im Hintergrund laufen lassen und abmelden
```

```
endless.sh &
```

aus ssh abmelden

```
exit
```

```
#### Step 4: in 2. Session - prüfen ob es noch läuft
```

ja, es läuft

```
ps aux | grep endless
```

```
### Version with & (background task) and nohup (no hangup)
```

```
#### Step 1: Create and run script
```

```
cd /usr/local/bin nano endless.sh
```

```
let i=0 while true do let i=i+1 echo $(date)"..jetzt.."${i} >> /var/log/endlosscheife.log echo $(date)"..hier ausgeben" sleep 2
done
```

```
chmod u+x endless.sh
```

```
#### Step 2: Im Hintergrund mit nohup laufen lassen und abmelden
```

```
pwd nohup endless.sh &
```

aus ssh abmelden

```
exit
```

```
#### Step 3: in 2. Session - prüfen ob es noch läuft
```

```
sudo su -
```

ja, es läuft

```
ps aux | grep endless tail nohup.out
```

```
## Festplattenbelegung

### Welche Verzeichnisse / Partitionen sind voll? du und df

### Ist meine Platte voll ?
```

Alle Partitionen

```
df -h
```

nur root partition

```
df -h /
```

eine bestimmte Partition

```
df -h /dev/sda1
```

ein file, welche partition ist das

```
df -h /etc/hosts
```

```
### Welche Verzeichnisse sind besonders voll ?
```

zeigt top-level verzeichnisse

```
du -h --max-depth=1 /
```

zeigt directories für /var/log an

```
du -h --max-depth=1 /var
```

```
## Benutzer, Gruppen und Rechte
```

```
### Rechte/Benutzer/Gruppe
```

```
### Arten
```

r = Lesen w = Schreiben x = Ausführen

```
### Für welchen Bereich ?
```

u = user g = gruppe o = others (die anderen / die Welt) a = für alle (d.h. gleichzeitig für u und g und o)

```
### Aufbau triple
```

```
kurs@ubuntu2004-101:~$ # rwx | rw- | r-- kurs@ubuntu2004-101:~$ # u g o kurs@ubuntu2004-101:~$ # 421 | 42- | 4--  
kurs@ubuntu2004-101:~$ # 7 | 6 | 4
```

rwx | rw- | r--

u g o

421 | 42- | 4--

7 | 6 | 4

```
### Berechtigungen mit Symbolen setzen
```

```
chmod g+w,o+r testfile
```

```
### Berechtigungen mit Octalzahlen setzen
```

```
chmod 777 testfile
```

```
### Berechtigungen recursiv setzen
```

```
chmod -R 777 testverzeichnis
```

```
### Besitzer ändern (nur als root)
```

```
ls -la datei5 -rw-r--r--. 1 kurs kurs 0 14. Nov 11:06 datei5
```

```
sudo chown root datei5
```

```
### Gruppe ändern (nur als root)
```

```
ls -la datei5 -rw-r--r--. 1 kurs kurs 0 14. Nov 11:06 datei5
```

```
sudo chown :root datei5
```

```
### Besitzer und Gruppe ändern (nur als root)
```

```
ls -la datei5 -rw-r--r--. 1 kurs kurs 0 14. Nov 11:06 datei5
```

```
sudo chown root:root datei5
```

```
### Dateien für Benutzer und Gruppen
```

```
cd /etc cat passwd cat shadow cat group
```

```
kurs@ubuntu2004-104:/etc$ ls -la passwd shadow group -rw-r--r-- 1 root root 1097 Mar 10 10:06 group -rw-r--r-- 1 root root 3164 Mar 10 10:06 passwd -rw-r----- 1 root shadow 1838 Mar 10 10:06 shadow
```

```
### Benutzer anlegen
```

```
### Benutzer anlegen (auf Ubuntu)
```

for shell script

```
useradd
```

for admins interactive

```
adduser
```

```
### sudo Benutzer erstellen
```

```
### Benutzer zum Sudo benutzer machen
```

```
adduser newuser usermod -aG sudo newuser
```

testing

su - newuser groups # see if we are in groups sudo id # shows the same but more info

need to enter password here

```
sudo su -
```

```
## Logs/Loganalyse
```

```
### Logfile beobachten
```

Terminal 1

```
tail -f /var/log/syslog
```

Terminal 2 - write to logfile e.g.

```
logger meine_nachricht
```

```
### Dienste debuggen
```

```
### Walkthrough
```

Dienst startet nicht / nach Ausführen von systemctl restart wird Fehlermeldung ausgegeben

```
systemctl restart mariadb.service
```

Schritt 1 : status -> was sagen die logs (letzte 10 Zeilen)

```
systemctl status mariadb.service
```

Nicht fündig-> Schritt 2:

```
journalctl -eu mariadb.service
```

Nicht fündig -> Schritt 3:

-e springt ans Ende des Pages

```
journalctl -e -u mariadb.service
```

Nicht fündig -> Schritt 4:

Spezifisches Log von Dienst suchen

und evtl. LogLevel von Dienst hochsetzen

z.B. bei mariadb (durch Internetrecherche herausfinden)

```
less /var/log/mariadb/mariadb.log
```

oder schneller

Zeige alle Zeilen mit dem Wort error an (case insensitive)

also auch z.B. ERROR

```
cat /var/log/mariadb/mariadb.log | grep -i error
```

Nicht fündig -> Schritt 5

Allgemeines Log

REdhat/Centos

```
/var/log/messages
```

```
### Wie verfahren bei SystemV
```

Wie bei walkthrough aber ab Schritt 4

```
### Find error in logs quickly
```

```
cd /var/log/mysql
```

-i = case insensitive // egal ob gross- oder kleingeschrieben

```
cat error.log | grep -i error
```

```
### Found wrong config-value, what now ?
```

You know the wrong config value, but not

where it is (in which file)

assuming gummitulpe is the wrong config value

```
grep -r gummitulpe /etc
```

Ausgabe

/etc/my.cnf.d/mariadb-server.cnf:gummitulpe=/nix

```
### Rsyslog

### Alle Logs an zentralen Log-Server schicken
```

/etc/rsyslog.conf

udp

.@192.168.10.254:514

tcp

.@@192.168.10.254:514

Ref: <https://www.tecmint.com/setup-rsyslog-client-to-send-logs-to-rsyslog-server-in-centos-7/>

```
### Journal analysieren

### Show all boots
```

journalctl --list-boots 0 3c3cf780186642ae9741b3d3811e95da Tue 2020-11-24 14:29:44 CET lines 1-1/1 (END)

```
### Show boot log
```

journalctl -b

```
### Journal persistent

* Normalerweise (auf den meisten Systemen), überlebt das Journal kein Reboot
```

persistent setzen

Achtung: in /etc/systemd/journald.conf muss Storage=auto gesetzt sein

Dies ist auch der Default - Fall

Achtung Achtung: Alle gezeigten Einträge mit # am Anfang sind die Default-Werte (in journald.conf)

mkdir /var/log/journal systemctl restart systemd-journal-flush.service

```
### Restrict how much is logged / data
```

in /etc/systemd/journald.conf

SystemMaxUse=1G

```
### journalctl
```

ubuntu

journalctl -u sshd

Nicht von Anfang, sondern die letzten Zeilen anzeigen

journalctl -eu sshd

```
### journalctl - Ausgabe json
```

sehr schön um alle felder zu sehen

journalctl -o json-pretty

```
### journalctl - konkreten Prozess anzeigen
```

journalctl _PID=5

```
### journalctl - was gibt es für Felder
```

journalctl -o json-pretty journalctl -u sshd.service -o json-pretty

```
### journalctl - mit Zeitangaben
```

alles seit gestern

journalctl --since yesterday journalctl --since now journalctl --since today

mit datum -> hier wichtig, dass richtige format

Mindestens Tag oder Tag und Uhrzeit (ohne sekunden)

nur Stunde geht nicht

journalctl --since "2022-08-17 00:05"

bis heute 09:45

journalctl --since yesterday --until "09:45"

```
### journalctl - immer die neuesten Infos ausgeben (wie bei tail -f)
```

```
journalctl -f -u apache2.service
```

```
### Help-pages
```

```
man journalctl man systemd.journal-fields
```

```
### Logrotate
```

```
cd /var/log cp -a messages output.log
```

```
cd /etc/logrotate.d vi output_log
```

```
/var/log/output.log { size 1k create 700 kurs kurs rotate 4 }
```

```
systemctl start logrotate.service ls -la /var/log/output*
```

```
## Dienste debuggen
```

```
### Dienste debuggen
```

```
### Walkthrough
```

Dienst startet nicht / nach Ausführen von systemctl restart wird Fehlermeldung ausgegeben

```
systemctl restart mariadb.service
```

Schritt 1 : status -> was sagen die logs (letzte 10 Zeilen)

```
systemctl status mariadb.service
```

Nicht fündig-> Schritt 2:

```
journalctl -eu mariadb.service
```

Nicht fündig -> Schritt 3:

-e springt ans Ende des Pages

```
journalctl -e -u mariadb.service
```

Nicht fündig -> Schritt 4:

Spezifisches Log von Dienst suchen

und evtl. LogLevel von Dienst hochsetzen

z.B. bei mariadb (durch Internetrecherche herausfinden)

```
less /var/log/mariadb/mariadb.log
```

oder schneller

Zeige alle Zeilen mit dem Wort error an (case insensitive)

also auch z.B. ERROR

```
cat /var/log/mariadb/mariadb.log | grep -i error
```

Nicht fündig -> Schritt 5

Allgemeines Log

REdhat/Centos

```
/var/log/messages
```

```
### Wie verfahren bei SystemV
```

Wie bei walkthrough aber ab Schritt 4

```
### Find error in logs quickly
```

```
cd /var/log/mysql
```

-i = case insensitive // egal ob gross- oder kleingeschrieben

```
cat error.log | grep -i error
```

```
### Found wrong config-value, what now ?
```

You know the wrong config value, but not

where it is (in which file)

assuming gummitulpe is the wrong config value

```
grep -r gummitulpe /etc
```

Ausgabe

```
/etc/my.cnf.d/mariadb-server.cnf:gummitulpe=/nix
```

```
## Variablen

### Setzen und verwenden von Variablen
```

```
DATEINAME=/etc/services echo $DATEINAME
```

Werte hochzählen

```
ZAHL=4 let ZAHL=ZAHL+1 echo $ZAHL
```

```
cat $DATEINAME
```

wird nicht der Inhalt verwendet sondern der Name \$DATEINAME

```
cat '$DATEINAME' cat "$DATEINAME"
```

Befehl ausführen und Rückgabewert anzeigen

```
date echo $?
```

Wert aus ausgeführtem Befehl in Variable schreiben

```
DATUM=$(date) echo $DATUM echo $DATUM >> /var/log/datumslog
```

```
## Dienste/Runlevel (Targets verwalten)

### Die wichtigsten systemctl/service

### Welche Dienste sind aktiviert ?
```

```
systemctl list-units --type=service
```

oder

```
systemctl list-units -t service
```

oder

```
systemctl -t service
```

```
### Wie finde ich einen service, der noch nicht aktiviert ist ?
```

```
systemctl list-unit-files -t service | grep mariadb
```

```
### Wie starte und stoppe ich einen Dienst ?
```

```
systemctl start httpd systemctl stop httpd
```

```
### Wie ist die Konfiguration eines Dienstes ?
```

```
systemctl cat sshd.service
```

```
### Wie sehe ich den status eines Dienstes ?
```

```
systemctl status sshd systemctl status sshd.service
```

ältere Variante

```
service sshd status
```

```
### Wie kann ich einen Dienst deaktivieren ?
```

d.h. dienst wird beim nächsten Boot nicht gestartet

```
systemctl disable sshd.service
```

oder

```
systemctl disable sshd
```

```
### Wie sehe ich, ob eine Dienst aktiviert / deaktiviert ist ?
```

```
systemctl is-enabled sshd.service echo $?
```

```
### Dienst aktivieren ?
```

```
systemctl enable sshd.service
```

```
### Wie sehe ich, wie ein Service konfiguriert ist / Dienstekonfiguration anzeigen ?
```

z.B. für Apache2

```
systemctl cat apache2.service
```

```
### Wie kann ich rausfinden, wie die runlevel als targets heissen ?
```

```
cd /lib/systemd/system root@ubuntu2004-104:/lib/systemd/system# ls -la run*target lrwxrwxrwx 1 root root 15 Jan 6 20:47  
runlevel0.target -> poweroff.target lrwxrwxrwx 1 root root 13 Jan 6 20:47 runlevel1.target -> rescue.target lrwxrwxrwx 1 root root
```

```
17 Jan 6 20:47 runlevel2.target -> multi-user.target lrwxrwxrwx 1 root root 17 Jan 6 20:47 runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jan 6 20:47 runlevel4.target -> multi-user.target lrwxrwxrwx 1 root root 16 Jan 6 20:47 runlevel5.target -
> graphical.target lrwxrwxrwx 1 root root 13 Jan 6 20:47 runlevel6.target -> reboot.target
```

```
### Welche Dienste sind enabled (preset) und auf dem System
```

```
systemctl list-unit-files -t service
```

```
### Dienste bearbeiten
```

```
systemctl edit sshd.service
```

Dann eintragen

```
[Unit] Description=Jochen's ssh-server
```

Dann speichern und schliessen (Editor)

```
systemctl daemon-reload systemctl status
```

```
### Targets (wechseln und default)
```

Default runlevel/target auslesen

```
systemctl get-default
```

in target wechseln

```
systemctl isolate multi-user
```

Default target setzen (nach start/reboot)

```
systemctl set-default multi-user
```

```
### Alle Target anzeigen in die ich reinwechseln kann (isolate)
```

Redhat / centos

```
grep -r "AllowIsolate" /usr/lib/systemd/system /usr/lib/systemd/system/reboot.target ... .. systemctl isolate reboot.target
```

```
### Dienste maskieren, so dass sie nicht gestartet werden können
```

```
systemctl mask apache2
```

kann jetzt gestartet werden

```
systemctl start apache2
```

de-maskieren

systemctl unmask apache2

kann wieder gestartet werden

systemctl start apache2

```
### systemctl - Diverse Beispiele
```

Status eines Dienstes überprüfen

service sshd status systemctl status sshd

Wie heisst der Dienst / welche Dienste gibt es ? (nur wenn der service aktiviert ist).

systemctl list-units -t service

für apache

systemctl list-units -t service | grep ^apache

die Abkürzung

systemctl -t service | grep ^apache

Wie finde ich einen service, der noch nicht aktiviert ist ?

systemctl list-unit-files -t service | grep ssh

Dienst aktivieren

systemctl enable apache2

Ist Dienst aktiviert

systemctl is-enabled apache2 enabled echo \$? 0 # Wenn der Dienst aktiviert ist

Dienst deaktivieren (nach Booten nicht starten)

systemctl disable apache2 systemctl is-enabled disabled echo \$? 1 # 1 wenn nicht aktiviert

Rebooten des Servers

verweist auf systemctl

reboot systemctl reboot shutdown -r now

Halt (ohne Strom ausschalten)

halt systemctl halt shutdown -h now

Poweroff

poweroff systemctl poweroff

```
### systemctl Cheatsheet

* https://access.redhat.com/sites/default/files/attachments/12052018_systemd_6.pdf

### Systemctl - timers

### Show all timers
```

alle Timer anzeigen

systemctl list-timers

```
### How ?

* .timer and .service file next to each other

### Example ?
```

timer - file

root@ubuntu2004-104:/etc# systemctl cat systemd-tmpfiles-clean.timer

/lib/systemd/system/systemd-tmpfiles-clean.timer

SPDX-License-Identifier: LGPL-2.1+

This file is part of systemd.

**systemd is free software; you can redistribute it and/or modify it
under the terms of the GNU Lesser General Public License as published by
the Free Software Foundation; either version 2.1 of the License, or
(at your option) any later version.**

[Unit] Description=Daily Cleanup of Temporary Directories Documentation=man:tmpfiles.d(5) man:systemd-tmpfiles(8)

[Timer] OnBootSec=15min OnUnitActiveSec=1d

Service - file

root@ubuntu2004-104:/etc# systemctl cat systemd-tmpfiles-clean.service

/lib/systemd/system/systemd-tmpfiles-clean.service

SPDX-License-Identifier: LGPL-2.1+

This file is part of systemd.

**systemd is free software; you can redistribute it and/or modify it
under the terms of the GNU Lesser General Public License as published by
the Free Software Foundation; either version 2.1 of the License, or
(at your option) any later version.**

[Unit] Description=Cleanup of Temporary Directories Documentation=man:tmpfiles.d(5) man:systemd-tmpfiles(8)
DefaultDependencies=no Conflicts=shutdown.target After=local-fs.target time-set.target Before=shutdown.target

[Service] Type=oneshot ExecStart=systemd-tmpfiles --clean SuccessExitStatus=DATAERR IOSchedulingClass=idle

```
### Personal Timer (timer for user)

* https://nielsk.micro.blog/2015/11/11/creating-systemd-timers.html

### Systemctl - timer example

### Step 1: Create script
```

cd /usr/local/bin nano script.sh

```
#!/bin/bash TAG='FREITAG' echo " ---- " date echo $TAG
```

```
### Step 2: Service-Unit für script
```

systemctl edit --full --force myscript.service

[Unit] Description=myscript

[Service] Type=oneshot ExecStart=/usr/local/bin/script.sh

systemctl start myscript.service

```
### Step 3: Timer-Unit für script
```

```
systemctl edit --full --force myscrip.timer
```

```
[Unit] Description=myscript timer
```

```
[Timer] OnBootSec=80 OnCalendar=*/0/2
```

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

```
### Step 4: Timer aktivieren
```

```
systemctl enable myscrip.timer systemctl start myscrip.timer systemctl list-timers
```

```
### Gegenüberstellung service etc/init.d/ systemctl
```

SySV

```
a) /etc/init.d/rsyslog status /etc/init.d/rsyslog start /etc/init.d/rsyslog status
```

```
b) service rsyslog
```

Systemd

geht auch (unter der Haube wird systemctl verwendet)

```
service rsyslog status
```

```
### Default Editor systemctl setzen
```

In der Session

```
export SYSTEMD_EDITOR=vim
```

in /root/.bashrc eintragen, wird dann bei jedem neuen Aufruf von bash z.B. sudo su - geladen

```
export SYSTEMD_EDITOR=vim
```

```
## Systemd
```

```
### Die wichtigen Tools für die Kommandozeile (ctl)
```

Oben die wichtigsten

```
systemctl journalctl # systemd logfiles abfragen hostnamectl # Hostname einstellen timedatectl localectl # locales konfigurieren
```

```
## Systemadministration

### Hostname setzen/abfragen
```

Abfragen

```
hostnamectl hostnamectl set-hostname centos4.training.local
hostnamectl
```

Trick für prompt - ist in der aktuellen, erst nach neueinloggen/bzw. neuer bash aktiv

su - root # bzw. su - benutzer

```
### ssh absichern

### Übung 1: Basisabsicherung mit AllowGroups
```

```
groupadd sshadmin usermod -aG sshadmin kurs
```

```
vi /etc/ssh/sshd_config
```

20241115 - jmetzger - only sshadmin

```
AllowGroups sshadmin
```

```
systemctl reload sshd
```

Testen... mit Nutzer kurs verbinden

per ssh

geht das

```
### sshd_config // Server

##/etc/ssh/sshd_config X11Forwarding no
```

if possible - no one can login with password

PasswordAuthentication no PermitRootLogin no

user must belong to a specific group to be allowed to login

AllowGroups wheel

if sftp is not need comment it - defaults to no

Subsystem sftp /usr/libexec/openssh/sftp-server

Restart sshd systemctl restart sshd

```
### Setup private/public key authentication
```

Authentication with password must be possible

When setting it up

Disable PasswordAuthentication afterwards

server2 client

as user kurs

ssh-keygen

set password set important

ssh-copy-id kurs@server1

Now you can login with public/private key

ssh kurs@server1

```
## Partitionierung und Filesystem

### parted and mkfs.ext4


### Walkthrough
```

Schritt 1: Platte in virtualbox oder gui-interface anlegen

Schritt 2: Platte identifizieren

lsblk

Schritt 3: Platte partitionieren

```
parted /dev/sdb
```

```
mklabel gpt mkpart data2 ext4 2048s 500M # data2 ist name der Partition bei gpt quit
```

Schritt 4: Partition formatiert

```
lsblk # Partition identifiziert mkfs.ext4 /dev/sdb1
```

Schritt 5: Mount-Punkt erstellen

```
mkdir /mnt/platte
```

Schritt 6: einhängen und aushängen

```
mount /dev/sdb1 /mnt/platte
```

Add-on: Eingehängte Partitionen anzeigen

```
mount
```

Aushängen

```
umount /mnt/platte
```

Schritt 7: Persistent konfigurieren

Eintragen in /etc/fstab

```
/dev/sdb1 /mnt/platte ext4 defaults 0 0
```

Schritt 8: Test, ob fstab gut ist (keine Fehler)

```
mount -av # v steht für geschwätzig.
```

Wenn das klappt: Schritt 9

```
reboot
```

Nach dem Rebooten

```
mount | grep platte # taucht platte hier auf ?
```

```
## Boot-Prozess und Kernel
```

```
### Grub konfigurieren
```

```
### Walkthrough
```

Step 1

z.B. timeout hochsetzen, wie lange er mit Booten im Bootmenu wartet

```
cd /etc/default vi grub
```

make wanted changes

```
##GRUB_TIMEOUT_STYLE=hidden GRUB_TIMEOUT=5
```

Step 2

```
update-grub
```

Step 3 - reboot

When grub menu appears enter arrow-down arrow-up ONCE

Dann zählt er nicht weiter runter und bootmenu bleibt stehen.

Mit e kann man einen boot-eintrag für den nächsten Boot ändern

Ändern und dann CTRL bzw. STRG + x für das Booten nach Änderung

Step 4 - be happy

```
### Kernel-Version anzeigen
```

```
uname -a
```

```
### Kernel-Module laden/entladen/zeigen
```

```
### Walkthrough
```

show kernel modules

```
lsmod
```

kernel - module entladen

```
modprobe -r psmouse lsmod | grep psmouse # now not present
```

damit wieder laden

```
modprobe psmouse lsmod | grep psmouse # now present
```

```
### Wo leben die Kernel - Module
```

kernel version is used, find out kernel version with

```
uname -a
```

```
cd /lib/modules/5.4.0-66-generic
```

e.g. psmouse

```
find /lib/modules -name psmouse* /lib/modules/5.4.0-66-generic/kernel/drivers/input/mouse/psmouse.ko
```

```
## Hilfe

### Hilfe zu Befehlen

### Möglichkeiten der Hilfe
```

anhand von ps

```
vi -h ps --help man ps info ps
```

```
### -h oder --help --> eines geht immer
```

Beispiel ls

```
ls -h # geht nicht für Hilfe ls --help # geht !
```

```
### Navigation in den man-pages
```

q - verlassen von man Pfeil oben/unten PageUp/PageDown G # für ans Ende der Datei springe 1g # in die erste Zeile

```
### Suche mit in man-pages
```

/Suchwort [Enter] n # nächster Treffer (kleines n) N # letzter Treffer

```
## Grafische Oberfläche und Installation

### X-Server - Ausgabe auf Windows umleiten

* https://www.thomas-krenn.com/de/wiki/Grafische_Linux_Programme_remote_von_einem_Windows_PC_mit_Xming_nutzen

### Installations-Images-Server

* https://ubuntu.com/download/server#download

## Wartung und Aktualisierung

### Aktualisierung des Systems
```



```
### Updaten des Systems
```

-y without asking

```
dnf -y update
```

or

```
dnf -y upgrade
```

is the same

```
### Paketmanager dnf
```

```
### Mögliche Paket anzeigen (die installiert sind und installiert werden können)
```

```
dnf list
```

```
### Installierte Pakete anzeigen
```

```
dnf list --installed
```

```
### Herausfinden, wie ein Paket heisst, dass ich installieren will
```

```
dnf list | grep mariadb
```

```
### Ist ein Paket installiert
```

```
dnf list --installed | grep mariadb
```

```
### Nach einem Paket suchen
```

```
dnf search mariadb
```

```
### Infos zu einem Paket abrufen
```

```
dnf info mariadb
```

```
### Archive runterladen und entpacken
```

Walkthrough

Schritt 1: Download-Link in Browser kopieren (rechte Maustaste)

Schritt 2:

```
cd /usr/src
```

falsche Dateiname -> umbenannt.

```
wget https://github.com/phayes/geoPHP/tarball/master mv master master.tar.gz
```

Schritt 3: Sicherheitsverzeichnis anlegen und entpacken

```
mkdir foo mv master.tar.gz foo cd foo tar xvf master.tar.gz
```

```
### Apache installieren (firewall und selinux)
```

```
### Walkthrough
```

Schritt 1:

suche // apache heisst auf centos httpd

```
yum search httpd
```

oder

```
dnf search httpd
```

Schritt 2:

```
yum install httpd
```

Wie heisst der Dienst und Starten und Enablen (für nächsten Reboot)

```
yum list-unit-files --type=service | grep httpd systemctl enable httpd systemctl start httpd
```

Schritt 3:

Konfiguration anpassen

/etc/httpd/conf/httpd.conf # Hauptkonfigurationsdatei

Änderungen mit Editor vornehmen z.B. nano

```
cd /etc/httpd/conf/httpd.conf; nano httpd.conf
```

Danach Neustart oder Reload

Restart funktioniert immer

```
systemctl restart httpd
```

Schritt 4:

Firewall freigeben

D.h. welche zone ist active -> public

```
firewall-cmd --get-active-zones
```

konfigurieren

```
firewall-cmd --add-service=http --permanent firewall-cmd --add-service=https --permanent firewall-cmd --reload
```

Schritt 5:

Mit Browser testen

```
### Apache started nicht wg Port-Änderung (Port: 82) - Quick and Dirty Lösung
```

/etc/httpd/httpd.conf

zeile hinzufügen

```
Listen 82
```

Es kommt ein Fehler bei Apache port 82 (Listen 82)

```
systemctl restart httpd
```

Schritt 1: Prüfen, ob selinux aktiv ist

```
sestatus # Sucht 2 Einträgen enforcing
```

Schritt 2: selinux testweise abschalten

```
setenforce 0 # das heisst, regeln werden protokolliert, aber nicht durchgesetzt
```

Schritt3:

```
systemctl restart httpd
```

Wenn das der Fall ist, selinux deaktivieren

```
/etc/selinux/config
```

mit editor

```
SELINUX=permissive
```

oder wenn man generell selinux nicht einsetzen möchte:

```
SELINUX=disabled
```

Danach rebooten

```
### Apache started nicht wg Port-Änderung (Port: 82) - Nice and Smooth (better!)
```

Falls sealert nicht installiert ist -> sealert -> command not found

```
yum whatprovides sealert
```

```
sealert -a /var/log/audit/audit.log > /root/report
```

In der Datei finden wir Handlungsanweisungen

Welche port-typen gibt es für http

```
semanage port -l | grep http
```

Wir entscheiden uns für http_port_t weil hier auch die 80 auftaucht

```
semanage port -a -t http_port_t -p tcp 82 setenforce 1 systemctl restart httpd
```

Don't forget to add firewall rules

firewall-cmd --list-all # is the port listed here ? firewall-cmd --add-port=82/tcp --zone=public --permanent # Sets in configuration but not in runtime firewall-cmd --reload

Now test with and your public ip

get it with

```
ip a
```

```
### mbr sichern mit dd

* Nur bei msdos mbr, nicht gpt

### Walkthrough
```

master boot record sichern

```
dd if=/dev/sda bs=512 count=1 of=mbr.img
```

Zurückspielen

```
dd if=mbr.img bs=512 count=1 of=/dev/sda
```

```
## Firewall und ports

### firewallld

### Install firewallld
```

on centos/redhat firewalld should installed

```
systemctl status firewalld
```

if not, just do it

```
yum install firewalld
```

```
### Is firewalld running ?
```

is it set to enabled ?

```
systemctl status firewalld firewall-cmd --state
```

```
### Command to control firewalld
```

```
* firewall-cmd
```

```
### Best way to add a new rule
```

Step1: do it persistent -> written to disk

```
firewall-cmd --add-port=82/tcp --permanent
```

Step 2: + reload firewall

```
firewall-cmd --reload
```

```
### Zones documentation
```

```
man firewalld.zones
```

```
### Zones available
```

```
firewall-cmd --get-zones block dmz drop external home internal public trusted work
```

```
### Active Zones
```

```
firewall-cmd --get-active-zones
```

```
### Show information about all zones that are used
```

```
firewall-cmd --list-all firewall-cmd --list-all-zones
```

```
### Add Interface to Zone ~ Active Zone
```

```
firewall-cmd --zone=public --add-interface=enp0s3 --permanent firewall-cmd --reload firewall-cmd --get-active-zones public  
interfaces: enp0s3
```

```
### Default Zone
```

if not specifically mentioned when using firewall-cmd

.. add things to this zone

```
firewall-cmd --get-default-zone public
```

```
### Show services
```

```
firewall-cmd --get-services
```

```
### What ports are opened in a service
```

Example ssh

```
cd /usr/lib/firewalld/services cat ssh.xml
```

```
### Adding/Removing a service
```

```
firewall-cmd --permanent --zone=public --add-service=ssh firewall-cmd --reload firewall-cmd --permanent --zone=public --remove-service=ssh firewall-cmd --reload
```

```
### Add/Remove ports
```

add port

```
firewall-cmd --add-port=82/tcp --zone=public --permanent firewall-cmd --reload
```

remove port

```
firewall-cmd --remove-port=82/tcp --zone=public --permanent firewall-cmd --reload
```

```
### Enable / Disabled icmp
```

```
firewall-cmd --get-icmptypes
```

none present yet

```
firewall-cmd --zone=public --add-icmp-block-inversion --permanent firewall-cmd --reload
```

```
### Working with rich rules
```

Documentation

man firewalld.richlanguage

throttle connections

```
firewall-cmd --permanent --zone=public --add-rich-rule='rule family=ipv4 source address=10.0.50.10/32 service name=http log level=notice prefix="firewalld rich rule INFO: " limit value="100/h" accept' firewall-cmd --reload # firewall-cmd --zone=public --list-all
```

port forwarding

```
firewall-cmd --get-active-zones firewall-cmd --zone=public --list-all firewall-cmd --permanent --zone=public --add-rich-rule='rule family=ipv4 source address=10.0.50.10 forward-port port=42343 protocol=tcp to-port=22' firewall-cmd --reload firewall-cmd --zone=public --list-all firewall-cmd --remove-service=ssh --zone=public
```

list only the rich rules

```
firewall-cmd --zone=public --list-rich-rules
```

persist all runtime rules

```
firewall-cmd --runtime-to-permanent
```

```
### References

* https://www.ispcolohost.com/2016/07/25/blocking-outgoing-ports-with-firewalld/
* https://www.linuxjournal.com/content/understanding-firewalld-multi-zone-configurations#:~:text=Going%20line%20by%20line%20through,or%20source%20associated%20with%20it.

* https://www.answertopia.com/ubuntu/basic-ubuntu-firewall-configuration-with-firewalld/

### Scannen und Überprüfen mit telnet/nmap

## Netzwerk/Dienste

### IP-Adresse von DHCP-Server holen (quick-and-dirty)

### Walkthrough
```

Ip nicht gesetzt - kurzfristig eine IP holen

```
ip a # zeigt die Netzwerkschnittstellen an. dhclient enp0s8 # ip - Adresse für Schnittstelle enp0s8 holen
ip a
```

```
### Auf welchen Ports lauscht mein Server
```

Zeigt alle ports an auf die gelauscht wird (ipv4)

```
lsof -i
```

alternative

```
netstat -tupel
```

```
### Interface mit nmtui-edit verwalten - schneller Weg
```

Achtung: richtigen profilnamen verwenden

einsehbar über nmtui oder

nmcli conn show

z.B. wenn enp0s9 als profil vorhanden ist

nmtui-edit enp0s8

```
### Netzwerkinterface auf der Kommandozeile einrichten
```

```
### Verbindungen anzeigen
```

nmcli connection show

or

nmcli conn show

```
### Netzwerk-Interface statisch auf Server neu einrichten (server 2)
```

muss in der Liste sichtbar sein

nmcli con add type ethernet con-name enp0s9 ifname enp0s9 ipv4.method manual ipv4.addresses 192.168.1.2/24 nmcli con mod enp0s9 autoconnect yes

verbindung neu hochziehen

nmcli con up enp0s9

verbindungseigenschaften anzeigen

nmcli con show

```
### Netzwerk-Interface statisch auf Server neu einrichten (server 3)
```

muss in der Liste sichtbar sein

nmcli con add type ethernet con-name enp0s9 ifname enp0s9 ipv4.method manual ipv4.addresses 192.168.1.3/24 nmcli con mod enp0s9 autoconnect yes

verbindung neu hochziehen

nmcli con up enp0s9

verbindungseigenschaften anzeigen

nmcli con show

```
### Netzwerk-Interface modifizieren (server 3)
```

muss in der Liste sichtbar sein

nmcli con add type ethernet con-name enp0s9 ifname enp0s9 ipv4.method manual ipv4.addresses 192.168.1.3/24 nmcli con mod enp0s9 autoconnect yes

verbindung neu hochziehen

nmcli con up enp0s9

verbindungseigenschaften anzeigen

nmcli con show

is ip gesetzt ?

ip a

```
### Ref:

* https://www.howtoforge.de/anleitung/wie-man-eine-statische-ip-adresse-unter-centos-8-
konfiguriert/

### Scannen mit nmap

### Scan Range
```

nmap -PE 192.168.1.2-5

```
## Mails

### lokale Mails installieren
```

apt install postfix mailutils

Internet Host

echo "testmail" | mail -s "subject" root

Gucken in der Datei

```
cat /var/mail/root
```

nach der gesendeten Email

```
## Bash/Bash-Scripting

### Einfaches Script zur Datumsausgabe
```

Mit nano öffnen / datei muss vorher nicht vorhanden sein

nano script.sh

Folgendes muss drin stehen, mit 1. Zeile beginnend mit

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

Speichern CTRL + O -> RETURN, CTRL X

Ausführbar machen

`chmod u+x script.sh ./script.sh # Ausführen und wohlfühlen`

```
### Ausführen/Verketteten von mehreren Befehlen
```

Beide Befehle ausführen, auch wenn der 1. fehlschlägt

```
befehl1; apt upgrade
```

2. Befehl nur ausführen, wenn 1. erfolgreich war.

```
apt update && apt upgrade
```

2. Befehl nur ausführen, wenn der 1. NICHT erfolgreich war

befehl1 oder befehl2 (im weitesten Sinne)

```
befehl1 || befehl2
```

```
### Example with date and if

### Example with function and return value

### Example with test and if

### Example log function

### Example Parameter auslesen
```

```
## Timers/cronjobs

### Cronjob - hourly einrichten

### Walkthrough
```

cd /etc/cron.hourly

nano datum

wichtig ohne Endung

Job wird dann um 17 nach ausgeführt ?

```
#!/bin/bash date >> /var/log/datum.log
```

chmod 755 datum # es müssen x-Rechte (Ausführungsrechte gesetzt sein)

Abwarten, Tee trinken

```
### cronjob (zentral) - crond

### Step 1: Festlegen, wann es laufen soll ?
```

cd /etc/cron.d

cronjob anlegen

Achtung: Dateiendung hier möglich, aber nicht in cron.daily, cron.hourly usw.

ls -la trainingscript

```
root@ubuntu2004-104:/etc/cron.d# ls -la trainingscript
```

```
-rw-r--r-- 1 root root 471 Mar 26 12:44 trainingscript
```

nano trainingscript

cat trainingscript

```
SHELL=/bin/sh PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

Example of job definition:

.----- 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**

`*/2 * * * * root /usr/local/bin/script.sh`

```
### Step 2: Script anlegen
```

`cd /usr/local/bin nano script.sh`

```
#!/bin/bash TAG='FREITAG' echo " ---- " >> /var/log/scripting.log date >> /var/log/scripting.log echo $TAG >> /var/log/scripting.log
```

Script - Berechtigungen setzten

`chmod u+x script.sh`

Scriptausführung testen

trägt es etwas im Log ein -> /var/log/scripting.log

`/usr/local/bin/script.sh`

```
### Step 3: Warten
```

nach 2 Minuten log betrachten

`ls -la /var/log/scripting.log`

cron daemon braucht nicht reloaded zu werden

```
## Literatur
```

```
### Literatur
```

```
### Literatur
```

```
* [Linux Grundlagen für Anwender und Administratoren]
(https://www.tuxcademy.org/product/grd1/)
* [Linux Systemadministration I für Anwender und Administratoren]
(https://www.tuxcademy.org/download/de/adm1/adm1-de-manual.pdf)
* [Alle Unterlagen] (https://www.tuxcademy.org/media/all/)

### Linux Sicherheit

* [Linux Sicherheit - inkl SELinux] (http://schulung.t3isp.de/documents/linux-security.pdf)

### Cheatsheet

* [..ansonsten Google :o)] (https://www.google.com/search?q=bash+cheatsheet)

## Wartung und Aktualisierung

### Paketmanager yum

### Mögliche Paket anzeigen (die installiert sind und installiert werden können)
```

yum list

```
### Installierte Pakete anzeigen
```

yum list --installed

```
### Herausfinden, wie ein Paket heisst, dass ich installieren will
```

yum list | grep mariadb

```
### Ist ein Paket installiert
```

yum list --installed | grep mariadb

```
### Nach einem Paket suchen
```

yum search mariadb

```
### Infos zu einem Paket abrufen
```

yum info mariadb-server

```
### Welche Programmpaket installiert ein bestimmtes Programm
```

Beispiel sealrt

yum whatprovides sealrt

```
### Cheatsheet
```

```
*
```

```
https://access.redhat.com/sites/default/files/attachments/rh\_yum\_cheatsheet\_1214\_jcs\_print-1.pdf
```

```
## Firewall
```

```
### Arbeiten mit firewalld
```

```
### Install firewalld
```

on centos/redhat firewalld should installed

```
systemctl status firewalld
```

if not, just do it

```
yum install firewalld
```

```
### Is firewalld running ?
```

is it set to enabled ?

```
systemctl status firewalld firewall-cmd --state
```

```
### Command to control firewalld
```

```
* firewall-cmd
```

```
### Best way to add a new rule
```

Step1: do it persistent -> written to disk

```
firewall-cmd --add-port=82/tcp --permanent
```

Step 2: + reload firewall

```
firewall-cmd --reload
```

```
### Zones documentation
```

```
man firewalld.zones
```

```
### Zones available
```

```
firewall-cmd --get-zones block dmz drop external home internal public trusted work
```

```
### Active Zones
```

```
firewall-cmd --get-active-zones
```

```
### Show information about all zones that are used
```

```
firewall-cmd --list-all firewall-cmd --list-all-zones
```

```
### Add Interface to Zone ~ Active Zone
```

```
firewall-cmd --zone=public --add-interface=enp0s3 --permanent firewall-cmd --reload firewall-cmd --get-active-zones public  
interfaces: enp0s3
```

```
### Default Zone
```

if not specifically mentioned when using firewall-cmd

.. add things to this zone

```
firewall-cmd --get-default-zone public
```

```
### Show services
```

```
firewall-cmd --get-services
```

```
### What ports a opened in a service
```

Example ssh

```
cd /usr/lib/firewalld/services cat ssh.xml
```

```
### Adding/Removing a service
```

```
firewall-cmd --permanent --zone=public --add-service=ssh firewall-cmd --reload firewall-cmd --permanent --zone=public --remove-  
service=ssh firewall-cmd --reload
```

```
### Add/Remove ports
```

add port

```
firewall-cmd --add-port=82/tcp --zone=public --permanent firewall-cmd --reload
```

remove port

```
firewall-cmd --remove-port=82/tcp --zone=public --permanent firewall-cmd --reload
```

```
### Enable / Disabled icmp
```

```
firewall-cmd --get-icmptypes
```

none present yet

```
firewall-cmd --zone=public --add-icmp-block-inversion --permanent firewall-cmd --reload
```

```
### Working with rich rules
```

Documentation

man firewalld.richlanguage

throttle connectons

```
firewall-cmd --permanent --zone=public --add-rich-rule='rule family=ipv4 source address=10.0.50.10/32 service name=http log level=notice prefix="firewalld rich rule INFO: " limit value="100/h" accept' firewall-cmd --reload # firewall-cmd --zone=public --list-all
```

port forwarding

```
firewall-cmd --get-active-zones firewall-cmd --zone=public --list-all firewall-cmd --permanent --zone=public --add-rich-rule='rule family=ipv4 source address=10.0.50.10 forward-port port=42343 protocol=tcp to-port=22' firewall-cmd --reload firewall-cmd --zone=public --list-all firewall-cmd --remove-service=ssh --zone=public
```

list only the rich rules

```
firewall-cmd --zone=public --list-rich-rules
```

persist all runtime rules

```
firewall-cmd --runtime-to-permanent
```

```
### References
```

```
* https://www.ispcolohost.com/2016/07/25/blocking-outgoing-ports-with-firewalld/
* https://www.linuxjournal.com/content/understanding-firewalld-multi-zone-configurations#:~:text=Going%20line%20by%20line%20through,or%20source%20associated%20with%20it.
```

```
* https://www.answertopia.com/ubuntu/basic-ubuntu-firewall-configuration-with-firewalld/
```

```
## Podman
```

```
### Podman Walkthrough
```

```
### Aufbau (Wirkweise)
```

```
![Aufbau Containerverwendung] (docker-podman.jpg)
```



```
### Walkthrough
```

runtergeladenen images

```
podman images
```

image von online ziehen (registry)

Sucht bei redhat danach bei docker.io

```
podman pull alpine:latest
```

Image ist jetzt lokal vorhanden

```
podman images
```

Container mit diesem image starten

```
podman run --name=myalpine alpine
```

Prozess läuft nicht mehr, da bereits beendet

```
podman ps
```

hiermit werden alle prozesse angezeigt auch die beendeten.

```
podman ps -a
```

Beendeten container löschen über container - id (muss eindeutig sein bei z.B. 2 Ziffern)

```
podman rm 08
```

liste der container ist jetzt leer

```
podman ps -a
```

```
### Container interactive mit terminal
```

das sind die Optionen -i -t

```
podman run -it --name=myalpine2 alpine
```

```
### Walkthrough II
```

interactive mit terminal und detached

Detached - es läuft weiter im hintergrund

```
podman run -dit --name=myalpine3 alpine
```

in maschine reinwechseln, Kommandas ls -la ausführen

danach wieder raus

```
podman exec -it myalpine3 ls -la
```

```
podman ps -a
```

geht nicht, weil es im container keine bash gibt

das ist bei alpine der fall, hier gibt es nur busybox

```
podman exec -it myalpine3 bash
```

einen sh - befehl gibt in jedem Linux

dieser verweist auf die aktuelle Shell

```
podman exec -it myalpine3 sh
```

Die Ausgabe des ersten Befehls wird geloggt

```
podman run -it --name=myalpine4 alpine ls -la
```

Logs anzeigen

```
podman logs myalpine4
```

```
### Configuration abfragen
```

Alle Konfigurationen

```
podman inspect myalpine3
```

oder container id

```
podman inspect a23e
```

```
podman inspect -f "{{.NetworkSettings.IPAddress}}" myalpine3 10.88.0.7
```

```
### Aufräumen (tabula rasa)
```

alle container und die, die noch laufen, vorher stoppen

```
podman rm -a --force
```

alle heruntergeladenen images löschen

```
podman rmi -a
```

```
### Image bauen
```

```
mkdir myimage cd myimage
```

vi Dockerfile beispiel ubuntu mit folgendem Inhalt

```
FROM ubuntu:20.04
```

```
RUN apt-get update RUN apt-get install -y nginx
```

```
ENV NEW_MODE laola ENV TRAINING_VERSION 1.0
```

```
FROM centos:latest
```

```
RUN yum install -y nginx ENV NEW_MODE laola ENV TRAINING_VERSION 1.0
```

choose any name for the image with -t

does not need to be the directory name

```
podman build -t myimage .
```

image als Basis für einen container verwenden

```
podman run -dit --name mycontainer myimage
```

Now work in the container if you want

```
podman exec -it mycontainer bash
```

do whatever you want in the container

e.g. env

```
## SELinux (Linux härten)

### SELinux

### sestatus

* Zeigt an, obwohl selinux aktiviert und wie

### Modi

* disabled
* enforcing (enabled)
* permissive (enabled)

### Persistente Konfiguration
```

/etc/selinux/config

```
### Dateien mit context anzeigen
```

ls -laZ

```
### Für nächsten Boot Kontext-Labels neu setzen
```

als root

cd / touch .autorelabel reboot

Achtung relabeln kann dauern !!! durchaus 5 Minuten

```
## Tools/Verschiedens

### Remote Desktop für Linux / durch Teilnehmer getestet

* https://wiki.ubuntuusers.de/Remmina/

### Warum umask 022 und 0002 ? - Geschichte
```

Just quoting redhat here.

The setting which determines what permissions are applied to a newly created file or directory is called a umask and is configured in the /etc/bashrc file.

Traditionally on UNIX systems, the umask is set to 022, which allows only the user who created the file or directory to make modifications. Under this scheme, all other users, including members of the creator's group, are not allowed to make any modifications. However, under the UPG scheme, this "group protection" is not necessary since every user has their own private group.

Ref:

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/4/html/reference_guide/s1-users-groups-private-groups