Linux Intensiv

Agenda

1. Grundlagen
   - Grundlagen
2. Systemd und Journalctl (Dienste verwalten)
   - Systemd
   - systemctl
   - journalctl
3. Arbeiten auf der Bash und unixtools
   - Arbeiten mit pipes
   - Arbeiten mit grep
   - Arbeiten mit less
4. Administrationsaufgaben / Major-Upgrade
   - Hostname setzen
   - Neue Festplatte konfigurieren und einbinden
   - Upgrade auf nächste Release
   - Auslastungsanalyse Netzwerk, Festplatte, CPU
5. Bash und Bash Programmierung
   - Bash - Variablen
   - Bash - Grundlegende Befehle der Systemadministration
   - Bash Programmierung
   - Bash script Beispiel
6. Kernel
   - Kernel Parameter
   - Kernel kompilieren
7. Find
   - Find
8. Verzeichnisse und Dateien
   - Grundlegende Ordnerstruktur
   - Grundlegende Dateioperationen
   - Ausgabe von Dateien
   - Ausgabe von gepackten Files / Entpacken von Files
   - Arbeiten mit Dateien und Verzeichnissen
9. Suche und Filtern
   - Suche
   - Übung mit Dateien filtern
10. Paketmanager (Ubuntu/Debian) und Software installieren
    - apt/dpkg
    - Software installieren
    - Aus Quelltext installieren
    - dnf
    - unattendend upgrades
11. Filesysteme
    - xfsdump und -restore
12. Sudo
    - sudo
13. ssh und scp
    - ssh
    - scp
    - ssh Kommandos auf Zielsystem ausführen
14. Bash und Bash-Programmierung
    - Strings escapen
    - Arbeiten auf der Bash
15. Editoren
    - Vi/vim
16. Logs
    - debugging log-files

17. Firewall
    - firewalld
    - ufw
18. Hilfe
    - Hilfe
19. Benutzer verwalten / Rechte
    - Benutzer
    - Rechte
20. Hilfreiche Programme
    - Hilfreiche Programme
21. Prozesse
    - Prozesse
22. Dienste verwalten /debuggen
    - Dienste
    - Dienste debuggen
23. Timer
    - Beispiel - Regelmäßiges Scannen mit nmap
24. Datensicherung
    - Datensicherung mit tar
25. Fragen,Tipps und Tricks / Howtos
    - Questions
    - Tipps und Tricks
    - Advanced Bash Scripting
    - Bash Scripting

Grundlagen

Grundlagen

Wann Linux, wann Windows ?

https://www.computerweekly.com/de/meinung/Das-beste-Server-Betriebssystem-Vergleich-zwischen-Linux-und-Windows#:~:text=Linux%20kommt%20im%20Data%20Center,viele%20verschiedene%20Einsatzzwecke%20zu%20verwenden.

Systemd und Journalctl (Dienste verwalten)

Systemd

layers of systemd

```
/lib/systemd/system  # never touch this is admin
/etc/systemd/system
/run/systemd/system
```

how it is started

```
units:

Innerhalb eines Targets , ausführen von

- service(s)
- mount(s)
- socket(s)
- timer (s)

Oder/und
     (sub)target(s)
        service
        mount
        timer
        socket
```

systemctl

```
systemctl status sshd
## Rausfinden welche aktivierten / laufenden Dienste
systemctl list-units -t service
## Rausfinde welche überhaupt
systemctl list-unit-files -t service

## Alle targets anzeigen
systemctl list-unit-files -t target

## is target aufrufbar ?
systemctl cat multi-user.target | grep AllowIsolate  # should be in last line

## default target anzeigen
## in welches target wird gebootet
systemctl get-default
systemctl set-default multi-user

## see configuration of units
systemctl cat multi-user.target
systemctl cat sshd.service # centos

## show running and next scheduled timers
systemctl list-timers
```

Dienste starten/aktivieren/deaktivieren/überprüfen

```
systemctl start httpd
systemctl start apache2

systemctl status apache2
systemctl status httpd

systemctl enable apache2
systemctl enable httpd

## enable and start
systemctl enable --now apache2
systemctl enable --now httpd

systemctl disable apache2
systemctl disable httpd

systemctl is-enabled apache2
systemctl is-enabled httpd
## wichtig wenn enabled rückgabe wert 0
echo $?
0
## wenn nicht aktiviert
echo $?
1
```

System runterfahren

```
## system runterfahren und ausschalten
systemctl poweroff
poweroff
```

tmpfiles.d

- Create manage temporary files

```
## just to have an idea how it works
cp -a /usr/lib/tmpfiles.d/tmp.conf /etc/tmpfiles.d/tmp.conf

## Edit file, that it has the following content
d /tmp/test 1777 root root -

## Get in / - directory, to see that is works everywhere
cd /

## Execute manually to check, if it works
systemd-tmpfiles --create

## check if dir exits
ls -la /tmp/test

## Hint: Deleting and setup is done daily and on boot
## by the following services and timers:
## systemctl list-unit-files | grep tmpfiles
systemd-tmpfiles-clean.service             static          enabled
systemd-tmpfiles-setup-dev.service         static          enabled
systemd-tmpfiles-setup.service             static          enabled
systemd-tmpfiles-clean.timer               static          enabled
```

timers

```
systemctl list-timers
```

journalctl

```
## show events of a specific unit
journalctl -u sshd.service
```

Example debugging of timer event

```
systemctl list-timers  | head -n 3
NEXT                         LEFT       LAST                         PASSED       UNIT
ACTIVATES
Mon 2020-11-23 13:11:17 CET  44min left Mon 2020-11-23 12:11:16 CET  15min ago    dnf-makecache.timer
dnf-makecache.service
Tue 2020-11-24 00:00:00 CET  11h left   Mon 2020-11-23 09:28:30 CET  2h 57min ago unbound-anchor.timer
unbound-anchor.service
[root@trn01 tmp]# journalctl -u dnf-makecache.service
```

systemctl

systemctl 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

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
```

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 aktiviert/deaktiviert

```
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)
```

```
## nur falls es nicht funktioniert !
## 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)

```
## Ubuntu
grep -r "AllowIsolate" /lib/systemd/system
/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 gestaret werden
systemctl start apache2
```

systemctl Cheatsheet

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

journalctl

journalctl

```
## ubuntu
journalctl -u ssh.service
## centos
journalctl -u sshd.service

## sehr schön um alle felder zu sehen
journalctl -o json-pretty

## alles was pid xy
journalctl _PID=5

## 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"

## nur neuen Sachen / Veränderungen ausgeben
journalctl -f -u apache2.service
```

Help-pages

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

Show all boots

```
 journalctl --list-boots
 0 3c3cf780186642ae9741b3d3811e95da Tue 2020-11-24 14:29:44 CET▒<80><94>T>
lines 1-1/1 (END)
```

Journal persistent

- Update: Ubuntu 22.04 ist per default 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
```

Arbeiten auf der Bash und unixtools

Arbeiten mit pipes

Beispiele

```
## Ausgabe an einen Pager schicken
ls -la | less
```

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

Ergebnis und 1 Zeile danach

```
apt search apache | grep -A 1 ^apache
## Alternativ für -B 10 davor (10 Zeilen davor)
```

Anzahl der Vorkommen anzeigen

```
ps aux | grep -c apache
```

Recursive Suchen (grep -r) - Schweizer Taschenmesser

```
grep -r "PermitRootLogin" /etc

## Mit Zeilennumber
grep -nr "PermitRootLogin" /etc
```

Arbeiten mit 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
```

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
```

Administrationsaufgaben / Major-Upgrade

Hostname setzen

```
## please do it as root user
hostnamectl
hostnamectl set-hostname server1.training.local
## only reflects after new login
su -
```

Neue Festplatte konfigurieren und einbinden

Walkthrough

```
## Schritt 1: Platte in virtualbox oder gui-interface anlegen

## Schritt 2: Platte identifizieren
lsblk

## Schritt 3: Platte partitionieren
## sdb platte auswählen
parted /dev/sdb

mkpart /dev/sdb1
mklabel gpt
mkpart data2 ext4 2048s 500M # data2 ist name der Partition bei gpt
quit

## Schritt 4: Partition formatiert
lsblk # Partition identfiziert
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 konfiguriren
## 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 ?
```

Upgrade auf nächste Release

Prerequsites

```
Sicherstellen, dass unser aktuelles System auf dem Stand der alten Release
apt update
apt upgrade
apt dist-upgrade

do-release-upgrade
```

Auslastungsanalyse Netzwerk, Festplatte, CPU

Top nach CPU - Auslastung

```
## High to low
top -o +%CPU
```

Top nach Speicher - Auslastung

```
top -o +%MEM
```

Netzwerkauslastung

```
bmon
```

Plattenauslastung

```
iotop
## Sortierung :
## Pfeiltaste links und rechts
## R - reverse
```

Bash und Bash Programmierung

Bash - Variablen

Bash - Grundlegende Befehle der Systemadministration

```
uname -a # welcher Kernel ist geladen

## erkannte devices auf usb anzeigen
lsusb

## welche block devices gibt es
lsblk
lslbk --fs # zeigt uuid und filesystem - typ
```

```
## partition mounten
 mount /dev/sdb1 platte
```

Bash Programmierung

Bash Programming - Howto's

https://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html https://tldp.org/LDP/abs/html/

Bash Script - Example

```
## /home/nobleprog/test.sh


##!/bin/bash
##
##ls -la
## Long Option with value
ls -la | head --lines=4
## Short option with value
ls -la | tail -n 4
```

Return Values

```
## Every command has a return value
## 0 = success
## 0 <> failure
example
date
echo $?
0

keinbefehl
echo $?
127
```

Testing conditions with test

```
nobleprog@jochen-g14d:~$ man test
nobleprog@jochen-g14d:~$ man test
## Test for directory
nobleprog@jochen-g14d:~$ test -d /etc
nobleprog@jochen-g14d:~$ echo $?
0
nobleprog@jochen-g14d:~$ test -d /etc2
nobleprog@jochen-g14d:~$ echo $?
1
## Does Directory not exist -> true = 0
nobleprog@jochen-g14d:~$ test ! -d /etc2
nobleprog@jochen-g14d:~$ echo $?
0
nobleprog@jochen-g14d:~$ man test
nobleprog@jochen-g14d:~$ [ ! -d /etc2 ]
nobleprog@jochen-g14d:~$ echo $?
0
nobleprog@jochen-g14d:~$ man test

## Be careful with spaces after [ and before ](must have)
nobleprog@jochen-g14d:~$ [! -d /etc2]
[!: command not found
nobleprog@jochen-g14d:~$ [ ! -d /etc2
```

If - Schleife

```
if /bin/true
  then
     echo 'then'
     exit 0
  else
     echo 'else'
     exit 1
fi
```

Beispiel-Script

```
##!/bin/bash

##echo 'scriptname:'$0
##echo 'param1:'$1
##echo 'alle params:'$@
##echo 'anzahl:'$#
source $HOME/config.sh
## . $HOME/config.sh

if test ! -d $DATEN
then
  echo "Verzeichnis $DATEN existiert nicht. Es wird angelegt" >> $LOGTO
  mkdir $DATEN
else
  echo "Verzeichnis $DATEN existiert. Alles gut" >> $LOGTO
fi

let i=0

while test $i -lt 1000000000000000
do
  let i=i+1
  DATUM=$(date)
  echo $DATUM" Zahl:"$i >>  $LOGTO
  echo $DATUM" Schlafe fuer 5 Sekunden" >> $LOGTO
  sleep 5
done
```

Bash script Beispiel

/etc/auswertung.conf

```
DATUM=$(date +"%Y-%m-%d")
LOG_BASE_DIR=/var/log/auswertung
LOGTO=$LOG_BASE_DIR"/"$DATUM".log"
```

/usr/local/bin/caller.sh

```
##!/bin/bash

##echo "Scriptname" $0
##echo "Parameter 1" $1
##echo "PArameter 2" $2
##echo "Parameter @- alle parameter" $@
##echo "Parameter Anzahlt" $#

## . /etc/auswertung.conf
source /etc/auswertung.conf

###
## Preparation
####
```

```
if test ! -d $LOG_BASE_DIR
then
  echo "lege verzeichnis auswertung an"
  mkdir $LOG_BASE_DIR
else
  echo "basedir existiert bereits"
fi

###
## Log function
mlog(){
  echo "[$(date +'%Y-%m-%d') $(hostname)] $1" >> $LOGTO

}

mlog "Alles Auf Anfang ...."


let COUNT=0

cd /etc
for i in a*
do
  if test -f $i
  then
    mlog "Datei gefunden: $i"
    let COUNT=COUNT+1
  fi
done

echo "Wieviel files"$COUNT

## /var/log/auswertung
let WIEOFT=0

while test $WIEOFT -lt 5
do
   FRAGE="Wie heisst Du ?"

   if [ $WIEOFT -gt 0 ]
   then
     FRAGE="Der Name gefällt mir nicht. Wie heisst Du sonst noch ?"
   fi


   echo $FRAGE
   read NAME
   echo "o.k. Du heisst $NAME"
   let WIEOFT=WIEOFT+1
done
```

```
cd /usr/local/bin
chmod u+x caller.sh
## Aufruf
caller.sh
```

Kernel

Kernel Parameter

Während der Laufzeit

```
root@ubuntu01:/proc/sys/net/ipv4# echo ip_forward
ip_forward
root@ubuntu01:/proc/sys/net/ipv4# cat ip_forward
```

```
0
## Ab sofort Pakete, die du bekommst, lieber kernel, weiterleiten
root@ubuntu01:/proc/sys/net/ipv4# echo 1 > ip_forward
root@ubuntu01:/proc/sys/net/ipv4#
```

At boot time

- Centos/Redhat:
- Ubuntu/Debian: man kernel-command-line

Kernel kompilieren

Ubuntu

- https://wiki.ubuntuusers.de/Kernel/Kompilierung/

Centos

- https://linuxhint.com/compile-linux-kernel-centos7/ # that one I would prefer
- https://www.tecmint.com/compile-linux-kernel-on-centos-7/
- -- bitte nicht die Original-Anleitung von centos im wiki verwenden.

Find

Find

Simple find command

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

Verzeichnisse und Dateien

Grundlegende Ordnerstruktur

- https://de.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

Grundlegende Dateioperationen

Verzeichnis wechseln und Liste anzeigen

```
## Verzeichnis wechseln
cd /

## In das Heimat oder User-Verzeichnis wechseln
cd

## Ein Verzeichnis höher wechseln
cd ..
```

In welchem Verzeichnis bin ich ?

```
## Verzeichnis anzeigen, in dem ich bin
pwd
```

Verzeichnis - Liste

```
## Verzeichnisliste anzeigen
ls -l

## Liste mit versteckten Dateien
ls -la

## Listing seitenweise anzeigen (Pager)
ls -la  | less
```

Hintergrund zu . und .. im Verzeichnis

```
### Hintergrund '.' und '..'

. = aktuelles Verzeichnis
.. = übergeordnetes Verzeichnis

ls -la training
total 8
drwxrwxr-x  2 nobleprog nobleprog 4096 Oct  5 09:22 .
drwxr-xr-x 24 nobleprog nobleprog 4096 Oct  5 09:22 ..
```

Verzeichnis/Datei anlegen, löschen und umbenennen

```
## mit relativem Pfad / im aktuellen Verzeichnis
mkdir training
mkdir /home/user/training2

## Verzeichnisstruktur anlegen (-p)
mkdir -p daten/2020/dezember
```

```
## Leeres Verzeichnis löschen
rmdir verzeichnisname
```

Datei anlegen/löschen

```
## Anlegen Datei
touch dateiname

## Löschen mit Nachfrage
rm -i dateiname

## Löschen ohne Nachfrage
rm dateiname

## Ausgabe - keine Berechtigung
nobleprog@jochen-g14d:/$ rm vmlinuz.old
rm: cannot remove 'vmlinuz.old': Permission denied

## Löschen von Verzeichnissen
rm -r verzeichnis
rm -R verzeichnis
```

```
## Umbenennen und Kopieren Datei/Verzeichnis
mv dateiname neuer_dateiname
mv verzeichnis neues_verzeichnis
cp dateiname neuer_dateiname

## Beim Kopieren alle Rechte übernehmen und Verzeichnis kopieren.
## in den meisten fällen besser als nur:
## cp verzeichnis -> weil rechte übernommen werden.
cp -a verzeichnis verzeichnis_neu
cp -a datei datei_neu
```

Verzeichnis kopieren

```
cp -r verzeichnis verteichnis_neu
```

Ausgabe von Dateien

Cat und Less (pager)

```
cat filename
## mit pager
cat filename | less
```

```
## direkt mit pager
less filename
```

Ausgabe von gepackten Files / Entpacken von Files

zcat und gzip -d

```
zcat /var/log/syslog.2.gz
cp /var/log/syslog.2.gz /root
cd /root
gzip -d syslog.2.gz
```

Arbeiten mit Dateien und Verzeichnissen

Beispiele

```
## Leere Datei anlegen
touch heute.txt
```

Suche und Filtern

Suche

Suche in Files mit grep

```
root@jochen-g14d:/etc# cat services | grep http
## Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-
numbers.xhtml .
http        80/tcp        www        # WorldWideWeb HTTP
https        443/tcp                # http protocol over TLS/SSL
http-alt    8080/tcp    webcache    # WWW caching service
root@jochen-g14d:/etc# cat services | grep voice box system
grep: box: No such file or directory
grep: system: No such file or directory
root@jochen-g14d:/etc# cat services | grep 'voice box system'
```

```
grep 'voice box system' /etc/services
```

Suche über alle Files mit grep -r (Schweizer Taschenmesser)

```
## durchsucht alle order recursive
grep -r muster /verzeichnis
```

Suche - egal ob gross oder klein

```
grep -ir VOICE /etc | grep services
```

Suche - alle Zeilen die nicht mit einem # anfangen

```
cat services | grep -v '^#'
```

Locate - Suche

```
## sucht in der datenbank, die täglich erstellt wird
locate training.sh
## auch wenn die datei gelöscht wurde am gleichen tag,
## wird diese angezeigt, weil sie nachts indiziert wurde
## nachts läuft
updatedb

## überprüfen, ob diese noch existiert vor der Ausgabe
locate -e training.sh
```

Übung mit Dateien filtern

```
1. Ins Heimatverzeichnis von nobleprog wechseln
2. Verzeichnisstruktur agenda/2020/januar anlegen
3. Das Januar - Verzeichnis in 01 umbenennen
4. Im Verzeichnis 01 eine Datei aufgaben anlegen (leer)
5. In diese Datei das Datum und in eine neue Zeile
einen beliebigen Text über die Kommandozeile anfügen.
6. Eine Sicherungskopie der Datei machen.
7. Aus dem Verzeichnis /etc die Datei services in
das Heimatverzeichnis des Benutzer nobleprog kopieren.
8. Alle Zeilen aus services ausgeben, die kein Kommentar (am Anfang der Zeile) enthalten und in die Datei
kommentarlos im Heimatverzeichnis des Benutzer nobleprog schreiben.
```

Paketmanager (Ubuntu/Debian) und Software installieren

apt/dpkg

Alle Pakete anzeigen, die installiert sind auf dem System

```
dpkg -l
## oder
apt list --installed
```

Alle Paket die zur Verfügung stehen

```
apt list
```

Wo sind die Repos konfiguriert

```
cat /etc/apt/sources.list
cd /etc/apt/sources.list.d
```

Paket deinstallieren und aufräumen

```
## mit Konfigurationsdateien deinstallieren
apt purge mariadb-server-10.3
## Konfgurationsdateien stehen lassen
apt remove mariadb-server-10.3

## Aufräumen / alle Pakete die nicht mehr benötigt werden
## nur binaries deinstalliren (alles ausser Konfiguration)
apt autoremove

## Abhängige Pakete mit Konfigurationsdateien deinstallieren
apt autopurge
```

Pakete händisch mit dpkg installieren

```
## Schritt 1: Im Browser
## Paket online finden und Link kopieren (Browser - Rechte Mauataste Link kopieren)

## Schritt 2: auf dem Linux Server
sudo apt install wget
cd /usr/src
wget http://archive.ubuntu.com/ubuntu/pool/main/a/acl/acl_2.2.53-10build1_amd64.deb
sudo dpkg -i acl_2.2.53-10build1_amd64.deb
```

Pakete mit apt search suchen

```
## Vorbereitung
apt update

## suche nache apache
```

```
apt search apache
## mit pager
apt search apache | less

## Alle Paket in denen apache am Anfang der Zeile fehlt
apt search ^apache | less
```

Installieren mit apt install

```
## mit genauem Namen
apt install apache2
```

Liste der Files aus dem Paket (wenn installiert)

```
dpkg -L openssh-server
```

Paket runterladen, wenn bereits installiert

```
apt install -d --reinstall openssh-server # -d steht für download-only
## Lädt das Paket unter
## /var/cache/apt/archives runter
```

Welche Dateien sind im Paket, die ausgerollt werden ? (ohne Installation)

```
cd /var/cache/apt/archives
dpkg --contents openssh-server-xyz.deb # im gleichen Verzeichnis oder vollen Pfad dorthin
## oder Paket haben händisch in ein anderes Verzeichnis runtergeladen (z.B. mit wget)
dpkg -c /usr/src/openssh-server-xyz.deb
```

Show all files/directories being installed by package

- Important: without files being created by install-script

```
dpkg -L openssh-server
```

dpkg -l - show all packages

```
dpkg -l
```

update repo & update system

```
apt update
apt upgrade
apt dist-upgrade
```

autoremove

```
apt autoremove
```

reinstall config - files

```
apt-cache pkgnames pulse |xargs -n 1 apt-get -o Dpkg::Options::="--force-confmiss" install --reinstall
```

deinstallation

```
apt remove package  # leave config-files
apt purge package # also delete config files
```

Software installieren

Debian / Ubuntu

```
## apt ist der Paketmanager
apt search apache2
## seitenweise ausgabe
apt search apache2 | less
## installieren
apt update
apt install apache2
```

```
## Repoositories, d.h. wo liegt die Software
## wird unter /etc/apt/source.list gepflegt
```

```
apt install
... installiert die Pakete (.deb - Dateien) und löst Abhängigkeiten automatisch auf
```

Debian/Ubuntu install with dpkg

```
## Download Package
wget https://repo.percona.com/apt/percona-release_latest.$(lsb_release -sc)_all.deb

ls -la percona-release_latest.focal_all.deb
-rw-rw-r-- 1 nobleprog nobleprog 11618 Sep  2 14:24 percona-release_latest.focal_all.deb

dpkg -i percona-release_latest.focal_all.deb
```

```
## rechner1 -> rechner2
## ins /tmp schreiben geht immer
training@rechner1$ scp paket.deb user@rechner2:/tmp

## auf rechner2
training@rechner2$ cd /tmp; sudo dpkg -i paket.deb
```

Debian/Update updaten / neuester Stand

```
apt upgrade
apt dist-upgrade
```

Aus Quelltext installieren

```
## 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
```

dnf

```
dnf search
dnf install
## remove package
dnf remove
dnf provides

## Zeigt installierte pakete an
dnf list --installed
```

```
## Zeigt alle verfügbaren Pakete an
dnf list
```

unattendend upgrades

- Automatic security updates in enabled by default on Ubuntu 20.04 LTS
- File 1

```
## That's default on Ubuntu 20.04
/etc/apt/apt.conf.d/20autoupgrades
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "1";
## apt autoremove // every 7 days
APT::Periodic::AutocleanInterval "7"
```

- File 2

  ```
  /etc/apt/apt.conf.d/50unattended-upgrades
  ## set needed configs here
  ```

```
## log - structure
```

tree /var/log/unattended-upgrades/ /var/log/unattended-upgrades/ ├── unattended-upgrades-dpkg.log ├── unattended-upgrades.log └── unattended-upgrades-shutdown.log

0 directories, 3 files

```
## Filesysteme

### xfsdump und -restore


### Backup and restore xfs
```

sdb ├─sdb1 ext4 ef2cc434-236f-4654-b773-72c7a61fe447 /mnt/platte └─sdb2 xfs 9fc967e0-eee0-4cf4-9fc8-fcb9cf1a037f /mnt/platte3

xfsdump -f /mnt/platte/_mnt_platte3_2.xfsdump /mnt/platte3 xfsrestore -f /mnt/platte/_mnt_platte3_2.xfsdump /mnt/platte3

```
### xfs inventar
```

Zeigt das Inventar an

Bedeutet, was wurde bereits gesichert

xfsdump -I

```
## Sudo

### sudo


#### Konfiguration
```

Erfolgt in /etc/sudoers /etc/sudoers.d/ (Verzeichnis)

Entscheidend eine Zeile die mit % für Gruppe beginnt, z.B. mit passwort-eingabe des ausführenden Benutzers.

Beispiel: Benutzer wäre training t training@foo$ sudo su - # Hier muss dann das Passwort von training eingegeben werden

Allow members of group sudo to execute any command

%sudo ALL=(ALL:ALL) ALL

Nutzer training muss der Gruppe sudo angehören

```
#### Konfigurations-Beispiel für Nobleprog
```

root@jochen-g14d:/etc/sudoers.d# cat nobleprog nobleprog ALL=(ALL:ALL) NOPASSWD:ALL

```
#### Sudo - User anlegen (root)
```

apropos user # find command adduser sudo adduser training

is group sudo present on system

cat /etc/group | grep sudo man usermod # Supplementary Groups

Add user training to supplementary group

usermod -aG sudo training

Testing

su - training # change to user sudo su - # find out if user training can execute sudo commands

```
#### Einen Nutzer zum sudo nutzer machen
```

auf Debian / Ubuntu

ist sudo also sudo - Gruppe definiert, die alles darf, was root darf

usermod -aG sudo dein_benutzer

auf Centos

usermod -aG wheel dein_benutzer

```
#### Be careful to not have enabled rootpw = true
```

the you must enter your root password instead of the user password

```
#### Eingeschränkte sudo - rechte für benutzer vergeben
```

adduser wartung cd /etc/sudoers.d echo "wartung ALL=(ALL) /bin/systemctl restart httpd" > wartung chmod 0440 wartung

zum testen

from root user

su - wartung sudo systemctl restart httpd

```
## ssh und scp

### ssh
```

```
#### Verbindung mit ssh von Linux zu Linux (auf der Kommandozeile)
```

ssh user@remoterechner

z.B

ssh 11trainingdo@56.34.12.11

```
### scp
```

scp remotels.sh trn01@10.10.11.126:~/ scp remotels.sh trn01@10.10.11.126:/home/trn01 scp trn01@10.10.11.126:/home/trn01/remotels.sh remotels.sh.bkup

TEMPDIR=$(mktemp -d) scp -r trn01@10.10.11.126:/home/trn01/ $TEMPDIR

```
### ssh Kommandos auf Zielsystem ausführen

### Einfacher Fall
```

Fall 1

COOL=/etc; echo $COOL; ssh trn01@10.10.11.126 'ls -la $COOL'

auf Zielsystem wird ausgeführt

ls -la $COOL # Allerdings ist diese Variable dort nicht gesetzt

Fall 2

COOL=/etc; echo $COOL; ssh trn01@10.10.11.126 "ls -la $COOL"

aus Zielsystem wird ausgeführt

ls -la /etc

```
### Komplexe Befehle in Variablen ausführen funktioniert nicht !

  * https://unix.stackexchange.com/questions/444946/how-can-we-run-a-command-stored-in-a-variable

### Kommandos lokal eingeben und auf zielrechner ausführen (interaktiv)
```

ssh trn01@10.10.11.126 'bash -s'

```
### Lokales script auf Zielrechner ausführen
```

ssh trn01@10.10.11.126 'bash -s' < lokalesscript.sh cat lokalesscript.sh | ssh trn01@10.10.11.126

```
## Bash und Bash-Programmierung

### Strings escapen
```

TEST='Mooshäusl'''s Fensterbau' echo $TEST

```
### Arbeiten auf der Bash
```

```
### 3 Arbeiten auf der Bash

#### Bash Builtins
```

Das sind Programme die in die Bash eingebaut und nicht extern sind

Mit type herausfinden, ob ein Befehl intern ist

nobleprog@jochen-g14d:~~/bin$ type jobs jobs is a shell builtin nobleprog@jochen-g14d:~~/bin$ help jobs

```
##### List alle bash-builtin Kommandos
```

Liste aller ... s.o. Überschrift

man bash-builtins

```
#### Bash Specials
```

Zeigt Heimatverzeichnis des aktuell eingeloggten Nutzers an

echo ~

Ausgeführte Befehle während meiner Session

history

Bestimmten Befehl aus der History ausführen:

!nr

Beispiel

!244 # Achtung wird direkt ausführt

```
#### Umgebungsvariablen anzeigen
```

Alle

env

eine spezielle, z.B

echo $PATH

```
#### Ein Programm verlassen
```

Variante 1: STRG + c (gleichzeitig drücken)

oder Variante 2: q

oder Variante 3: Termin schliessen

=> eines von beiden geht eigentlich immer

```
#### PATH / Wo werden Kommandos gesucht ?
```

env echo $PATH which ls which command

```
#### Variable setzen
```

LISTE=ls echo $LISTE

Kommando ausführen

$LISTE

mit Argumenten

$LISTE -la

```
#### Variablen auswerten (einfache und doppelte Hochkommas)
```

Einfache Hochkommas -> kein Auswertung

root@jochen-g14d:/var/log# echo 'Das ist mein Pfad $PATH'

root@jochen-g14d:/var/log# echo "Das ist mein Pfad $PATH" Das ist mein Pfad /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games root@jochen-g14d:/var/log# echo "Das ist mein Pfad $(date)" Das ist mein Pfad Mon 05 Oct 2020 01:20:57 PM UTC root@jochen-g14d:/var/log# echo "Das ist mein Pfad $PATH"

```
#### In den root-Benutzer (Administrator) wechseln
```

Hat alle rechte wie Administrator unter Windows

sudo su

oder mit ins Heimatverzeichnis wechseln

sudo su -

```
#### Variablen setzen und Variablen mit Funktionsausgaben setzen (Ausgabe des ausgeführten Programms)
```

date

keine Ausgabe des Datums

DATUM=date

Ausgabe des datums

DATUM=$(date)

direkt im echo

echo $(date)

Direkt in ein File schreiben

root@jochen-g14d:/var/log# echo $(date)' eine wunderbare Zeit' >> training.log root@jochen-g14d:/var/log# cat training.log heute ist ein schöner tag Mon 05 Oct 2020 01:16:15 PM UTC eine wunderbare Zeit root@jochen-g14d:/var/log#

```
#### Gehört kommando zur shell oder ist es eigenständig (oder alias)
```

nobleprog@jochen-g14d:~~$ type umask umask is a shell builtin nobleprog@jochen-g14d:~~$ type ls ls is aliased to `ls --color=auto' nobleprog@jochen-g14d:~$ type cd cd is a shell builtin nobleprog@jochen-g14d:~$ type grep grep is

`aliased to` grep --color=auto' nobleprog@jochen-g14d:~~$ type less less is /usr/bin/less nobleprog@jochen-g14d:~~$

```
#### Shell Expansion mit '*'
```

echo new* newdir newdir2 newfile newfile2 nobleprog@jochen-g14d:~$ ls -la newdir newdir2 newfile newfile2 -rw-rw-r-- 1 nobleprog nobleprog 0 Oct 6 11:11 newfile ---------- 1 nobleprog nobleprog 0 Oct 6 11:17 newfile2

newdir: total 8 drwxrwxr-x 2 nobleprog nobleprog 4096 Oct 6 11:13 . drwxr-xr-x 27 nobleprog nobleprog 4096 Oct 6 11:17 .. ls: cannot open directory 'newdir2': Permission denied

```
## Editoren

### Vi/vim


### 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 eine 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 w und enter

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

Logs

debugging log-files

```
Step1:
systemctl status service-name
## if step1 was not succesful

Step2:
journalctl -u service-name

## step 3: (if not step was not sucessful )
## Look in specific log files of service under
/var/log
e.g. mysql
/var/log/mysql
e.g. mariadb
/var/log/mysql
or
/var/log/mariadb

## if this does not work
/var/log/messages or /var/log/syslog
```

Firewall

firewalld

Install firewalld and restrict ufw

```
## Schritt 1: ufw deaktivieren
systemctl stop ufw
systemctl disable ufw
ufw disable # zur Sicherheit
ufw status
## -> inactive # this has to be the case

## Schritt 2: firewalld
apt update
apt install -y firewalld

## Schritt 3: firewalld
apt install firewalld
systemctl start firewalld
systemctl enable firewalld
systemctl status firewalld
systemctl status ufw
```

Is firewalld running ?

```
## is it set to enabled ?
systemctl status firewalld
firewall-cmd --state
```

Command to control firewalld

- firewall-cmd

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
## in our case empty
```

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
```

Show information about all zones that are used

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

Default Zone

```
## if not specifically mentioned when using firewall-cmd
## .. add things to this zone
```

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

Show services / Info

```
firewall-cmd --get-services
firewall-cmd --info-service=http
```

Adding/Removing a service

```
## Version 1 - more practical
## set in runtime
firewall-cmd --zone=public --add-service=http
firewall-cmd --runtime-to-permanent

## Version 2 - less practical
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --reload
```

```
### Service wieder entfernen
firewall-cmd --permanent --zone=public --remove-service=ssh
firewall-cmd --reload
```

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
```

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.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/

ufw

enable ufw / if not enabled yet

```
ufw status
ufw enable
ufw status
ufw disable
```

ipv6 or not ?

```
/etc/default/ufw
IPV6=no
```

check status and the current policies

```
ufw status verbose
```

set default policies

```
sudo ufw default deny incoming
sudo ufw default allow outgoing
```

Dienst erlauben

```
## ssh wird über /etc/services gefunden
sudo ufw allow ssh
## alternativ
sudo ufw allow 22
```

Portbereiche

```
sudo ufw allow 6000:6007/tcp
sudo ufw allow 6000:6007/udp
```

Port - Ranges

```
sudo ufw allow 6000:6007/tcp
sudo ufw allow 6000:6007/udp
```

Subnetze

```
sudo ufw allow from 203.0.113.0/24
## or specific port
sudo ufw allow from 203.0.113.0/24 to any port 22
```

zu spezifischer Netzwerkschnittstelle

```
sudo ufw allow in on eth0 to any port 80
```

Deny - Verbindung

```
sudo ufw deny http
sudo ufw deny from 203.0.113.4
```

Löschen von Regeln

```
## nach nummer
sudo ufw status numbered
sudo ufw delete 2

## nach tatsächlicher regel
sudo ufw delete allow http
```

Prüfen von Regeln

```
sudo ufw status verbose
```

Zurücksetzen der Regeln

```
ufw reset
```

ufw - port weiterleitung

- https://qastack.com.de/server/238563/can-i-use-ufw-to-setup-a-port-forward

References:

- https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-20-04-de

Hilfe

Hilfe

Man - Seiten

```
man ls
## (Verlassen mit q )

#### Suchen in der Hilfe
/suchbegriff + Enter
## nächster Vorkommen Taste ->
n (wie next)

## hilfeseite des man - kommandos
man man
```

--help / -h - Schalter in Befehlen

```
befehl --help
oder
befehl -h

Beispiel:
ls --help
```

Befehle suchen mit apropos

```
## nach befehlen suchen
apropos befehl
## copy soll nur in der Überschrift vorkommen
## Ausgabe durch pager
apropos copy files | grep copy | less
```

Benutzer verwalten / Rechte

Benutzer

Wer bin ich (der eingeloggte Nutzer) ?

```
## Wer bin ich / Unter welchem Benutzer bin ich eingeloggt
whoami
```

Mehr über mich:

```
## Mehr über mich
id
```

Alle Gruppen eines Benutzer anzeigen (in denen er Mitglied ist)

```
groups
```

Benutzer ausgeben / anlegen

```
## Location of users in system
cat /etc/passwd

man useradd
man adduser
## On Debian/Ubuntu
## use adduser to interactively add a new user
adduser training
## use useradd for scripts
## Important: Use options
useradd training
```

Unter einem anderen Benutzer anmelden (während einer Session)

```
su --login training
```

Shell eines Benutzer ändern

```
## Direktes Ausloggen nach Login
usermod -s /bin/false training
## Mit Ausgabe und Ausloggen nach Einloggen
usermod -s /usr/sbin/nologin training
```

Benutzer löschen

```
## Löschen als unprivilegierter Nutzer mit sudo-Rechten
sudo userdel training
```

Rechte

Arten

```
r = Lesen
w = Schreiben
x = Ausführen
```

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
```

Bedeutung Oktalzahlen

```
## rwx
## r = 4, w = 2, x = 1
```

Beispiele

```
- w -
0 2 0  = 020
```

Berechtigungen mit Symbolen setzen

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

Hilfreiche Programme

Hilfreiche Programme

wc (Word Count) - Zählen von Bytes, Worten, Zeilen

```
wc -l dateiname
cat /etc/services | wc -l
```

Prozesse

Prozesse

pstree / Baum der Prozess anzeigen

```
pstree -p | less

## beispiel mit finden der bash in welchen Prozessen
pstree -p | grep bash
```

top / Prozesse interaktiv betracht

```
top
q # mit q verlassen
```

```
Bedeutung:
load average: 0.08, 0.04, 0.00

Wieviel Prozesse warten auf Ausführung durch die CPU in der/den letzten 1 min / 5 min / 15 min
```

```
##
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   7107.8 avail Mem

Swap - Daten die auf Festplatte ausgelagert werden, weil zu wenig Arbeitsspeicher da
=> Achtung: Sollte auf Servern möglichst nicht > 0 sein
=> Chef, ich brauche mehr Arbeitsspeicher (wenn > 0)
```

ps - prozesse anzeigen als Liste

```
Fall 1: Alle Prozesse anzeigen

## macht in der Regel als root
sudo ps aux | less
sudo ps aux
```

```
## Information über bestimmtes kommando (Alle Prozesse)
ps aux | grep bash

## Wieviel Prozesse
ps aux | grep bash | wc -l
ps aux | grep -c bash
```

Beispiel: Prozess-Id (pid) finden

```
## starter.sh ist das script was wir suchen
ps aux | grep starter.sh # Info in Spalte 2 (Spalte 1 = user)

## mit header
ps aux | head --lines=1; ps aux | grep starter.sh
```

Beispiel: Prozess für Kommando in Files ausgeben

```
ps aux > /home/user/ausgabe.ps
```

Beenden Programme im Vordergrund

```
sleep 1000
STRG + c # beendet das programm
```

ps - optionen herleiten

```
2 Varianten:
ps aux
ps -ef
```

```
## Beispiele man-pages relativ weit
man ps
```

```
ps --help simple
```

kill - Signale schicken an Prozesse

kill sendet ein Signal an einen Prozess

```
## Alle möglichen Signale anzeigen
kill -L
```

```
## das gleich wie STRG + c
kill -15 1123 # 1123 ist die Prozess ID
das gleich wie:
kill 1123 # da 15 das Default Signal
```

```
## Pistolennummer
## Wir haben uns Kind 2x vorgewarnt
kill -9 1123
```

Besondere Signale

```
SIGSTOP - STRG + z (Prozess stoppen und im Arbeitsspeicher behalten)
kill -19 1234 # SIGSTOP

##
SIGHUP
kill -1 1234 # Reload Configuration File
```

Prozess im 2. Terminal beenden

```
## Terminal 1
## starter.sh starten
```

```
## Terminal 2
ps aux | grep starter.sh
## Prozess mit PID (Prozess Id) killen -> z.b 1234
kill 1234

## Terminal 2
## Reagiert der Prozess nicht, dann -9 (SIGKILL) nachschieben
kill -9 1234
```

Prozesse im Hintergrund starten / Jobs

```
## Prozess läuft komplett im Hintergrund wenn keine Ausgaben
## Ich kann weiter arbeiten im Terminal
starter.sh &
[1](nur in dieser session) // 123607 in allen Sessions verfügbar
```

```
## Zeige alle jobs an
nobleprog@jochen-g14d:~/bin$ jobs
[1]+  Running                 starter.sh &
## Schicke STOP signal = siehe kill -L
nobleprog@jochen-g14d:~/bin$ kill -19 %1
## War letzter Befehl erfolgreich
nobleprog@jochen-g14d:~/bin$ echo $?
0

[1]+  Stopped                 starter.sh
nobleprog@jochen-g14d:~/bin$ fg starter.sh
starter.sh # jetzt läuft er wieder im Vordergrund
```

Script im Hintergrund laufen lassen (auch nach Terminal schliessen)

```
## Terminal 1
nohup starter.sh &
ls -la nohup.out
## Terminal 1 schliessen

## Terminal 2
## Script läuft nach schliessen von Terminal 1 noch
ps aux | grep starter.sh # Spalte 2: pid: z.b. 1234
kill 1234
```

Dienste verwalten /debuggen

Dienste

Dienste verwalten

```
## Läuft apache?
systemctl status apache2
## Apache stoppen
systemctl stop apache2
##
systemctl status apache2
##
systemctl start apache2
## Beim Starten des Servers dienst nicht starten
systemctl disable apache2
## Status --> disabled
systemctl status apache2
## Enable (Beim Booten des Servers starten)
systemctl enable apache2
```

```
## Welches Services gibt es auf dem System
systemctl list-units -t service
```

Log-Datei von systemd => journalctl

```
## Alle Ereignisse zu apache2
journalctl -u apache2
```

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:
jourrnalctl -xe

## Nicht fündig -> Schritt 3:
journalctl -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/mysql/error.log

## Nicht fündig -> Schritt 5
## Allgemeines Log
## Debian/Ubuntu
/var/log/syslog
## 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
```

Schweizer Taschenmesser der Suche

```
## Fehler ist gummitulpe - option - falsch in Konfigurationsdatei, aber wo ?
grep -r gummitulpe /etc
```

Timer

Beispiel - Regelmäßiges Scannen mit nmap

Walkthrough

```
## Schritt 1:
## /usr/local/sbin/scan.sh
[root@centos8-01 sbin]# cat scan.sh
##!/bin/bash
IP_RANGE=192.168.56.100-103
```

```
nmap -A -T4 $IP_RANGE

## Schritt 2: service erstellen
## systemctl edit --force --full scan.service
## /etc/systemd/system/scan.service
[Unit]
Description=nmap scanning of environment

[Service]
Type=oneshot
ExecStart=/usr/local/sbin/scan.sh
##RemainAfterExit=true
StandardOutput=journal

### nach dem Speichern kann man das bereits testen mit
## systemctl start scan.service
### Ergebnis im Journal
## journalctl -u scan.service

## Schritt 3:
## Timer angelegt :
## systemctl edit --force --full scan.timer
## /etc/systemd/system/scan.timer
[Unit]
Description=Timer for Scan Service

[Timer]
OnCalendar=*:0/5

[Install]
WantedBy=basic.target

## Schritt 4:
## Timer starten und aktivieren
systemctl enable scan.timer
systemctl start scan.timer

## Schritt 5:
## Beobachten und glücklich sein
systemctl list-timers
## <- taucht der Timer dort auf

## Schritt 6:
## Reboot
## und Danch : taucht der timer auf ?
systemctl list-timers
```

Datensicherung

Datensicherung mit tar

Sichern / Backup

```
cd /usr/src
tar cfvz _etc.20220617.tar.gz /etc
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
```

Referenz:

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

Fragen,Tipps und Tricks / Howtos

Questions

How to boot/setup raid-1 // software raid with linux

- https://serverfault.com/questions/634482/setting-up-a-bootable-multi-device-raid-1-using-linux-software-raid

How to update only kernel on centos 8

- dnf upgrade kernel
- be sure to reboot, that new kernel is loaded

Why do bluetooth devices not work within lxc containers ?

- Kernel as of 2017 does not support bluetooth namespaces
- namespacing bluetooth devices
- https://github.com/lxc/lxd/issues/3265

How is Ubuntu Live Patch working

- Module with patches is compiled with patches
- modules is loaded and functions are redirected with ftrace
- https://ruffell.nz/programming/writeups/2020/04/20/everything-you-wanted-to-know-about-kernel-livepatch-in-ubuntu.html

Is docker based on lxc ?

- Yes !
- ... since it is based on LXC
- https://www.upguard.com/blog/docker-vs-lxc#:~:text=Docker%20is%20developed%20in%20the,provided%20by%20the%20underlying%20infrastructure.

```
Docker is developed in the Go language and utilizes LXC, cgroups, and the Linux kernel itself. Since it's based on LXC, a Docker container does not include a separate operating system; instead it relies on the operating system's own functionality as provided by the underlying infrastructure.22.10.2020
```

What is /var/log/hp on Ubuntu ?

```
## this is created and used by hplip (Hewlett Packard Image Printing)
## and is installed by default
dpkg -L hplip
...
...
/var/log/hp
...
```

What is wtmp and btmp ?

```
wtmp : historical data of users being logged in
last -f wtmp  # shows the information

btmp records only failed login attempts.
last -f btmp
```

numpad with vi and putty

```
The answer is in Numpad in PuTTY while using vi [Cialug]:

In the configuration, go to Terminal->Features and check "Disable application keypad mode". Save the
settings and enjoy a numeric pad that works!
```

Tipps und Tricks

RDP-Verbindung statt über VPN über SSH herstellen

```
## -p port
## -l user
## -N do not execute remote commands
ssh -L 5000:192.168.178.26:3389 LINUXSERVER -p 22500 -l LINUXUSER -N
## Dann aufruf von rdp -> localhost:5000
```

Reverse Forwarding

```
Specifies that connections to the given TCP port or Unix
socket on the remote (server) host are to be forwarded to
the local side.
## Es es nicht notwendig port-forwarding dafür auf dem Router zu machen
ssh -R 80:localhost:80 user@remoteserver
```

letzten Befehl als sudo ausführen (aus history)

- sudo !!

Verzeichnisstruktur als Baum anzeigen

```
## unter Ubuntu installieren
apt install tree

tree /home/centos01
```

Suche nach Konfigurationseinstellung / bzw. Inhalte eines unbekannten Files

```
## in which file is this directive
grep -r Listen /etc/httpd
```

Advanced Bash Scripting

- https://tldp.org/LDP/abs/html/

Bash Scripting

- https://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html