# Linux Individual-Training

## Agenda

1. Distributionen
    - Überblick
    - Aufbau/Komponenten
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
    - Autovervollständen * und tab
    - Welches Programm wird verwendet
4. Erweiterte Befehle (Nice to have)
    - Alias Befehle anzeigen
    - Welche Bibliotheken verwendet ein ausführbares Programm
    - Wo liegt ein ausführbares Programm
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
    - Grafisch Navigieren auf der Kommandozeile - mc
6. Systemadministration
    - Hostname setzen/abfragen
    - ssh absichern
7. Prozesse
    - Prozesse anzeigen - ps/pstree -p und top
8. Benutzer, Gruppen und Rechte
    - Rechte
    - Dateien für Benutzer und Gruppen
    - Benutzer anlegen
    - Wie funktioniert die Maske (umask)
    - sudo Benutzer erstellen
9. 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
- Find
- sed
- awk

10. Logs/Loganalyse
    - Logfile beobachten
    - Dienste debuggen
    - Rsyslog
11. Variablen
    - Setzen und verwenden von Variablen
12. Dienste/Runlevel(Targets verwalten)
    - Die wichtigsten systemctl/service
    - Systemctl - timers
    - Gegenüberstellung service etc/init.d/ systemctl
13. Partitionierung und Filesystem
    - parted and mkfs.ext4
14. Boot-Prozess und Kernel
    - Grub konfigurieren
    - Kernel-Version anzeigen
    - Kernel-Module laden/entladen/zeigen
15. Hilfe
    - Hilfe zu Befehlen
16. Grafische Oberfläche und Installation
    - Gnome unter Ubuntu installieren
    - X-Server - Ausgabe auf Windows umleiten
    - Installations-Images-Server
17. Installation/Wartung und Aktualisierung
    - Aktualisierung des Systems
    - Paketmanager apt/dpkg
    - Paketmanager rpm/yum
    - Archive runterladen und entpacken
    - Lokalen Mirrorserver aufsetzen - Centos
    - Installationsbeispiel Apache auf Centos
18. Firewall und ports
    - ufw (uncomplicated firewall)
    - firewalld - Ubuntu
    - firewalld - Centos 8
    - Scannen und Überprüfen mit telnet/nmap
19. Netzwerk/Dienste
    - IP-Adresse von DHCP-Server holen (quick-and-dirty)
    - IP-Adresse auslesen

- Netzwerk unter Centos konfigurieren - nmtui
- Netzwerk unter Centos konfigurieren - sysconfig
- Auf welchen Ports lauscht mein Server
- Netzwerkabel drin/nicht drin?
- Welcher DHCP-Server über NetworkManager

20. Tools/Verschiedens

- Remote Desktop für Linux / durch Teilnehmer getestet
- Warum umask 002 und 0002 ? - Geschichte
- lokale Mails installieren

21. Bash/Bash-Scripting

- Einfaches Script zur Datumsausgabe
- Ausführen/Verketten von mehreren Befehlen
- Howto: Bash Scripting
- Howto: Advanced Bash Scripting

22. Timers/cronjobs

- Cronjob - hourly einrichten
- cronjob (zentral) - crond
- Beispiel-Regelmäßiges Scannen mit nmap

23. Literatur/Documentation

- Literatur
- Linux Sicherheit/SELinux/nmap
- sssd gegen ADS
- realmd gegen ADS
- MariaDB Galera Cluster mit Ansible
- Ansible Module
- Linux Hardening - TelekomSecurity
- Example Question Linux Foundation LFCSA

24. Tipps&Tricks

- Output von terminal sesssion inkl. SSH-Verbindung loggen
- Vagrant (Windows/OS X)
- Centos auf Virtualbox installieren
- Digitalocean - droplet-virtual machine installieren
- bash-profile testen

25. Übung

- Übung Dateisystem
- Übung Komplett - Centos8

# Distributionen

## Überblick

### Multi-Purpose - Distributionen (Ideal zum Starten)

#### Redhat-Familie

```
Centos (bis 2021 wie Redhat, danach z.B. AlmaLinux)
Redhat.  — rpm / (yum / dnf)
Fedora

Redhat Developer Version: https://developers.redhat.com/blog/2016/03/31/no-cost-rhel-
developer-subscription-now-available#
```

#### Redhat Developer Program

- https://developers.redhat.com/blog/2016/03/31/no-cost-rhel-developer-subscription-now-available#

#### Debian Familie

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

#### SuSE - Familie

```
SLES (SuSE Linux Enterprise)
OpenSuSE - rpm, yum (zypper)
```

### 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/

### Aufbau/Komponenten

```
Installer
Bootmanager
Kernel
Paket-Verwaltung (geholt über die Repositories)
Software
Grafische Oberfläche
Administrations-Tool
```

### Grafische Oberfläche

```
Gnome (in der Regel bzw. das was vorgegeben wird) - Das ist die verbreiteste.
```

# Verzeichnisse und Dateitypen

## Verzeichnisaufbau

### /etc

- Verzeichnis für Konfigurationsdatein

### /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, initital 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
```

## Basisbefehle

### In den Root-Benutzer wechseln

```
## einloggen als normaler Benutzer z.B. benutzer: kurs
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
```

### 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
```

### Autovervollständen * 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

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

### Welche Bibliotheken verwendet ein ausführbares Programm

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

### Wo liegt ein ausführbares Programm

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

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

### 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 Verteichnis
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 Zeilverzeichnis 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
!!
```

### Ordner verschieben

```
## Wir sind root
cd /root
## Vorbereitung
mkdir test
cd test
mkdir pruefung
cd pruefung
touch ergebnis

## Verzeichnis training
mkdir /root/training

## Jetzt verschieben
## Schritt 1: Ins Verzeichnis reinwechseln, dass Verzeichnis beinhaltet, dass ich
verschieben möchte
cd /root/test
## Variante 1: mit absolutem Pfad
mv pruefung /root/training
## Alternativ: Variante 2: relativier Pfad
mv pruefung ../training
```

### 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
```

### Grafisch Navigieren auf der Kommandozeile - mc

```
## centos
yum install mc

## ubuntu - not sure, not tested
apt install mc
```

# 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

### 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

## server1 client
## as user kurs
ssh-keygen
## set password set important
ssh-copy-id kurs@server2

### Now you can login with public/private key
ssh kurs@server2
```

# Prozesse

### Prozesse anzeigen - ps/pstree -p und top

### Prozesse anzeigen

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

### Prozesse anzeigen für Benutzer

```
ps -u kurs -o pid,cmd # -o für felder der ausgabe
```

### Wieviele apache - Prozesse gibt es ?

```
## Achtung Ergebnis - 1 - weil grep noch mit auftaucht
ps aux | grep -c apache
```

### top für bestimmten user

```
top -U kurs
```

### systemctl (läuft Dienst)

```
systemctl status sshd
```

### Prozeßbaum anzeigen (meist nicht für die Praxis notwendig)

```
pstree -p
```

# Benutzer, Gruppen und Rechte

## 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
```

### Berechtigungen mit Symbolen setzen

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

### Ausführungsrechte für Benutzer für Script setzen

```
chmod u+x script.sh
```

### SGID - Bit setzen

```
## Alle neuen Dateien und Verzeichnisse bekommen als Gruppe die Gruppe des Verzeichnisses
ls -la | grep dokumente
drwxrwsr-x.   5 root dg4   131 15. Apr 10:18 dokumente

## d.h. hier sind neu angelegte Ordner und Dateien in der Gruppe 'dg4'
## Abweichend vom Standardverhalten: user/gruppe vom benutzer der Datei anlegt

## + Vererbung funktioniert hier auch.


chmod g+s /dokumente
```

### 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
```

### Benutzer anlegen (auf Centos)

```
## no difference between useradd/adduser
adduser
```

### Set password (you need to be root, if you do not know the former password)

```
## man passwd
passwd
```

### Wie funktioniert die Maske (umask)

### Welche Rechte haben neue Dateien

- Diese wird durch die Maske des eingeloggten Benutzers bestimmt (umask)

### Maske anzeigen / setzen

```
umask

## Nur während der Session aktiv
umask 0222
```

### Wie funktioniert es ?

```
## Maximale Rechte einer Datei
666 = rw-rw-rw-

## Maximale Rechte Verzeichnis
777 = rwxrwxrwx

## Beim Anlegen eines Verzeichnisse / Datei werden die Rechte berechnet:

## Datei
  0666
- 0002 <- diese Wert kommt von umask
======
  0664 <- Diese Rechte werden verwendet

## Verzeichnis
  0777
- 0002 <- diese Wert kommt von umask
======
  0775 <- Diese Rechte werden verwendet
```

### Maske während der Session setzten

```
umask 0000
```

### Mask persistent setzen für meinen Benutzer (auch nach Neuanmeldung gültig)

```
## Entweder mit Editor oder
## Achtung Zwei / 2 ! >>
echo "umask 0000" >> ~/.bashrc
## Testen
su - benutzer # benutzer ist der Name des Benutzer unter dem ich eingeloggt bin
```

### sudo Benutzer erstellen

### Benutzer zum Sudo benutzer machen (Ubuntu/Debian)

```
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 -
```

### Benutzer zum sudo benutzer machen (Redhat/Centos)

```
adduser newuser
usermod -aG wheel 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 -
```

# 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 Zeilennumer

```
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 letzten -x Zeilen anzeigen

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

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

cat /etc/services | tail -n 500
## mit umleitung datei
cat /etc/services | tail -n 500 >> neuedatei # anhängen - bestehende Zeilen werden nicht
überschrieben.
```

## Ausgabe der letzten Zeilen und ausgabe in Datei

```
cd /var/log
tail -n 100 syslog.1 >> fehlerlog
cat fehlerlog
```

### 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
```

### Erweiterte Suche mit Grep

### Nach einzelenen 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|Moira)' /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|Moira)' /etc/services'^(mysql|Moira)' /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/

## Find

### Simple find command

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

### Find diretories ending on ".d"

```
find / -type d  -name "*.d"
```

### Find files by inode and delete them

```
## helpful when file name is cryptic
## see what the inode with
## ls -lai # -i for inode number
find -inum 12604361 -type f -delete
```

### sed

### Examples

```
## Search for all occurences of tcp
cp /etc/services /root/services
cd /root
sed 's/tcp/linux/g' services

## line 1 to line 3
sed '1,3 s/unix/linux/' services
```

### Delete

### Delete Pattern matching line

```
## Example 1
cd /root
sed '/tcp/d' services

## Example 2
cd /root
sed '/^#/d' services > services_no_comments
```

### Edit in place

```
sed '/^#/d' -i.bkup services
```

### Ref

- https://www.geeksforgeeks.org/sed-command-in-linux-unix-with-examples/

## awk

### Examples

```
awk '{print $3 "\t" $4}' /etc/services
awk '/tcp/' /etc/services

awk '{print $2" "$1}' services
```

## Logs/Loganalyse

### Logfile beobachten

```
## Terminal 1
tail -f /var/log/syslog

## Terminal 2 - write to logfile e.g.
logger meine_nachricht
```

### Dienste debuggen

### Prerequisites

```
## Install mariadb-server - Centos
yum install mariadb-server

## Find out the service name
systemctl list-unit-files -t service | grep mariadb

##
```

### 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
##
## z.B.
## grep -ir mariadb /var/log
## und evtl. LogLevel von Dienst hochsetzen
##
## ODER:
## 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
```

### Falsche Option in welcher Konfigurationsdatei

```
## wir haben die falsche Option gummitulpe drin
grep -ir gummitulpe /etc
```

### 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
```

### 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/
```

## 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

### systemctl Beispiele

```
## Status eines Dienstes überprüfen
service sshd status
systemctl status sshd

## Wie heisst der Dienst / welche Dienste gibt es ?
systemctl list-units -t service
systemctl list-units --type=service
systemctl -t service # Shortcut
## für apache
systemctl list-units -t service | grep ^apache
## die Abkürzung
systemctl -t service | grep ^apache
```

### Dienste aktivieren / deaktivieren

```
## Dienst aktivieren
systemctl enable apache2
## Dienst aktivieren und gleich starten
systemctl enable --now apache2 # Ubuntu
systemctl enable --now httpd # Centos / Redhat

## 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
```

### Reboot / Poweroff

```
## 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 bei einem neu installierten Serverdienst (z.B. httpd / mariadb) wie der Dienst heisst ?

```
## ACHTUNG: Direkt nach der Installation klappt folgendes nicht  !::
## systemctl list-units -t service | grep -i mariadb # weil der Service noch nicht
aktiviert ist.

## Stattdessen nehme dann:
systemctl list-unit-files -t service | grep -i mariadb # Hier sucht er in allen
Konfigurationsdateien
```

### Dienste starten und stoppen

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

### 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 (enabled/disabled) - sollten laufen ?

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

### Welche Dienste laufen (active/inactive) ?

```
## Dort der Eintrag in der Zeile active/inactive
systemctl list-units --type=service | less
```

### Welche Dienste sollten zwar laufen, konnten aber nicht gestartet werden mit einbeziehen ?

```
systemctl list-units --all --type=service | less
```

### 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 - in welchen Modus/Status geht das System beim Booten
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

```
### Ubuntu
systemctl mask apache2
## kann jetzt gestartet werden
systemctl start apache2

## de-maskieren
systemctl unmask apache2
## kann wieder gestaret werden
systemctl start apache2
```

```
### Centos / Redhat
systemctl mask httpd
## kann jetzt gestartet werden
systemctl start httpd

## de-maskieren
systemctl unmask httpd
## kann wieder gestaret werden
systemctl start httpd
```

### 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
```

### Example Reference

- https://www.tutorialdocs.com/article/systemd-timer-tutorial.html

### Personal Timer (timer for user)

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

### 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
```

## 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
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 ?
```

# 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 # Version for ubuntu/debian
grub2-mkconfig -o /boot/grub2/grub.cfg  # Version for centos

## 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 in man-pages

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

## Grafische Oberfläche und Installation

### Gnome unter Ubuntu installieren

```
sudo apt install tasksel
sudo tasksel install ubuntu-desktop
```

### 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

## Installation/Wartung und Aktualisierung

### Aktualisierung des Systems

```
apt update
apt upgrade
apt dist-upgrade

## oder geht auch auf älteren Systemen
apt-get update
apt-get upgrade
apt-get dist-upgrade
```

### Paketmanager 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
## konfgiurationsdateien stehen lassen
apt remove mariadb-server

## Aufräumen / alle Pakete die nicht mehr benötigt werden
apt autoremove
```

### 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
apt search ^apache | less
```

## Installieren mit apt install

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

### Paketmanager rpm/yum

### Wo werden die Repos konfiguriert

```
/etc/yum.repos.d
```

### Grundbefehle

```
## Was ist installiert
yum list --installed

## Nach Paket suchen
yum search httpd

## Paket installieren
yum install httpd

## Welches Installations-Paket stellt nmap zur Verfügung
yum provides nmap
```

### System aktualisieren / updaten

```
## Installation updaten
yum upgrade # yum update is still available but not documented
### automatisch fragen bejahen
yum -y update
```

### Welche Paket stellt einen bestimmten Befehl zur Verfügung ?

```
yum whatprovides sealert
```

### Paket deinstallieren

```
yum remove mariadb-server
```

### 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
```

### Lokalen Mirrorserver aufsetzen - Centos

- https://wiki.centos.org/HowTos/CreateLocalMirror

### Installationsbeispiel Apache auf Centos

### 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

```
## Es kommt ein Fehler bei Apache port 82 (Listen 82)
systemctl start 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 einsetzten 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
```

## Firewall und ports

### ufw (uncomplicated firewall)

### Läuft der Dienst für die Firewall

```
systemctl status ufw
```

### Ist die Firewall scharfgeschaltet ?

```
ufw status
```

### Firewall aktivieren (Achtung ssh)

```
## Neue ssh - Verbindungen werden nicht angenommen
## Bestehende Bedingungen (ESTABLISHED) bleiben erhalten
ufw enable
ufw status
```

### Port hinzufügen

```
ufw allow 22 # for tcp and udp
## or
ufw allow ssh # uses /etc/services for detection of port - number
ufw status
```

### Port wieder rausnehmen

```
ufw delete allow http
ufw delete allow ssh
ufw delete allow 22

## ufw status numbered
## e.g.
ufw delete 1
```

### firewalld - Ubuntu

### Install firewalld and restrict ufw

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

## Schritt 2: 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

### 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
## in our case empty
```

### 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
```

### 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.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/

### firewalld - Centos 8

### 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 --persistant

## 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
## in our case empty
```

### 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
```

### 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
```

### Walkthrough apache / adding Port (Centos 8 / Redhat 8 with enabled SELinux (by default))

```

## /etc/httpd/conf/httpd.conf
## add port
## Listen 82
## Try to restart - not working port cannot be bound
sealert -a /var/log/audit/audit.log
## we will get this info to allow this port
semanage port -a -t http_port_t -p tcp 82
## start apache
systemctl start httpd
firewall-cmd --add-port=82/tcp --zone=public --permanent
```

### Enable / Disabled icm

```
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/

### Scannen und Überprüfen mit telnet/nmap

### nmap - examples

```
## nmap
nmap 10.10.9.124
nmap -A -F 10.0.9.124

## specific port
nmap -p 3306 -A 10.10.9.124
```

### telnet

```
##
telnet 10.10.9.124 3306
```

## 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
```

### IP-Adresse auslesen

```
ip a
ip address

ifconfig # geht nicht immer / muss manchmal nachinstalliert werden
```

### Netzwerk unter Centos konfigurieren - nmtui

```
1.  Menüpunkt bearbeiten
2.  Netzwerkinterface auswählen
3.  Dann bearbeiten.
4.  Einstellungen vornehmen
5.  Speichern
6.  Zurück
7.  Beenden

## Navigation erfolgt über die Tasten
TAB
Leertaste für Auswählen
und Pfeiltasten
```

### Netzwerk unter Centos konfigurieren - sysconfig

```
cd /etc/sysconfig/
nano ifcfg-enp0s8
## Änderungen vornehmen
systemctl restart NetworkManager
```

### Auf welchen Ports lauscht mein Server

```
## Zeigt alle ports an auf die gelauscht wird (ipv4)
lsof -i
## alternative
netstat -tupel
```

### Netzwerkabel drin/nicht drin?

```
## Variant 1 (Centos/Redhat  && Ubuntu/Debian)
ip link show enp0s8

## If cable it not plugin-in
NO-CARRIER

## Variant 2: Works on all systems
cat /sys/class/net/enp0s8/operstate

## Variant 3: nmcli (Centos/Redhat)
nmcli conn show  # Alle Connections
nmcli dev show  # Alle Netzwerk-Devices
## Show only active connection (activated with nmtui)
nmcli conn show --active
nmcli conn show enp0s8
## Kabel gezogen
nmcli dev show enp0s8
## -> must have: ...CARRIER: aus/an
nmcli dev show enp0s8 | grep -i carrier  # -i -> case insensitive -> egal ob gross oder
klein geschrieben
##
nmcli -f WIRED-PROPERTIES dev show enp0s8
##
```

### Netzwerkkarte aktivieren / deaktivieren nmcli

```
nmcli conn show
## Das wäre das gleich wie aktivieren/deaktivieren in nmtui
## Achtung . muss der Name der Connection sein und nicht Interface
## Auf centos8 testsystem war das identisch
nmcli conn up enp0s8
nmcli conn down enp0s8
```

### Welcher DHCP-Server über NetworkManager

```
/etc/NetworkManager/NetworkManager.conf
level=TRACE
domains=ALL

## Restart des NetworkManager
systemctl restart NetworkManager

journalctl -u NetworkManager | grep received | less

## Apr 16 12:46:10 centos8-01 NetworkManager[28829](enp0s8): received ACK of
192.168.56.101 from 0.0.0.0
```

## Tools/Verschiedens

### Remote Desktop für Linux / durch Teilnehmer getestet

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

### Warum umask 002 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
```

### 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 CRTL + O -> RETURN, CTRL X

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

### Ausführen/Verketten 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 befehlt2 (im weitesten Sinne)
befehl1 || befehl2
```

### Howto: Bash Scripting

- https://tldp.org/LDP/Bash-Beginners-Guide/html/sect_02_02.html

### Howto: Advanced Bash Scripting

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

## 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

```
cd /etc/cron.d
### cronjob anlegen
## Achtung: ohne Dateiendung
## ls -la trainingscript
## root@ubuntu2004-104:/etc/cron.d# ls -la trainingscript
## -rw-r--r-- 1 root root 471 Mar 26 12:44 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  /root/script.sh


### Script anlegen
cat 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 /root/script.sh

### Scriptausführung  testen
### trägt es etwas im Log ein -> /var/log/scripting.log
/root/script.sh


##### Warten


### nach 2 Minuten log betrachten
ls -la /var/log/scripting.log

### cron daemon braucht nicht reloaded zu werden
```

### 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
DATESTAMP=$(date +'%Y%m%d_%H%M%S')
LOGTO=/var/log/scan-$DATESTAMP.log

##nmap -A -T4 $IP_RANGE
echo "Starte nmap scan for $IP_RANGE"
nmap -Pn --open $IP_RANGE > $LOGTO
echo "Done....Written log to: $LOGTO"

## 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
```

# Literatur/Documentation

### Literatur

### Literatur

- Linux Grundlagen für Anwender und Administratoren
- Linux Systemadministration I für Anwender und Administratoren
- Alle Unterlagen

### Cheatsheet

- Cheatsheet bash
- ..ansonsten Google :o)

### Linux Sicherheit/SELinux/nmap

- https://schulung.t3isp.de/documents/linux-security.pdf

### sssd gegen ADS

- https://access.redhat.com/articles/3023951

### realmd gegen ADS

- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/windows_integration_guide/index#sssd-ad-proc

### MariaDB Galera Cluster mit Ansible

- https://github.com/jmetzger/ansible-galera-cluster-maxscale

### Ansible Module

- https://docs.ansible.com/ansible/2.9/modules

### Linux Hardening - TelekomSecurity

- https://github.com/telekomsecurity/TelekomSecurity.Compliance.Framework

### Example Question Linux Foundation LFCSA

- https://training.linuxfoundation.org/wp-content/uploads/2019/04/LFCS-Practice-Questions-v1.0.pdf

## Tipps&Tricks

### Output von terminal sesssion inkl. SSH-Verbindung loggen

- https://leszekjaskierny.wordpress.com/2017/01/27/mac-x-os-log-terminal-session-to-file/

### Vagrant (Windows/OS X)

### OSX

```
## Install virtualbox
## Install vagrant -> vagrantup.com

## Go on the commandline -> terminal
cd
mkdir centos8-directory
cd centos8-directory
vagrant init centos/8
vagrant up
vagrant ssh
```

### Windows

```
## git for windows ->
## https://git-scm.com/download/win
## Install virtualbox
## Install vagrant -> vagrantup.com

## Go on the commandline -> terminal
cd
mkdir centos8-directory
cd centos8-directory
```

```
vagrant init centos/8
vagrant up
vagrant ssh
```

### Centos auf Virtualbox installieren

```
1. Installation von virtualbox (virtualbox.org)
2. ISO von Centos runterladen. z.B. http://ftp.rrzn.uni-
hannover.de/centos/8.4.2105/isos/x86_64/CentOS-8.4.2105-x86_64-boot.iso
3. Neue VM aufsetzen in virtualbox.
```

### Digitalocean - droplet-virtual machine installieren

- https://www.digitalocean.com/

### bash-profile testen

```
source /root/.bash-profile
## oder kürzer
. /root/.bash-profile
```

# Übung

## Übung Dateisystem

```
1. In das Heimatverzeichnis von root wechseln
2. Ordner training anlegen
3. In Ordner training Inhalt der Datei /etc/services in datei ports schreiben
4. In Heimatverzeichnis root Struktur anlegen: staedte/kinos/saehle
5. In saehle leere datei anlegen mit name reservierung
6. Verzeichnis saehle löschen
7. In Heimatverzeichnis root Ordner papierkorb anlegen
8. Verzeichnis kinos in papierkorb verschieben
9. Namen der Staedte "Köln, Hamburg München" in verzeichnis staedte in Datei liste
schreiben
10. An liste staedte "Duisburg, Mannheim, Karlsruhe" anhängen.
```

### Übung Komplett - Centos8

```
Erstelle den Nutzer "teilnehmer"
Setze ein Passwort für den Nutzer
Sorge dafür, dass der Nutzer als "root" arbeiten kann
Test dies mit einer neuen ssh-Verbindung (Putty)
Erstelle eine neue Gruppe teamwork
Ornder teilnehmer dieser Gruppe zu.

Erstelle eine neue Verzeichnisstruktur im Heimatverzeichnis von "teilnehmer" ->
planung/themen/projekte
Kopiere die Datei /etc/services in den neue angelegten Unterordner projekt
Lass die ersten 1000 Zeilen der services datei in projekte ausgeben und schreibe diese in
ergebnis_ports
Lass die letzten 500 Zeilen der services datei in projekte ausgeben und hänge diese ans
Ende von ergebnis_ports an
Zählen die Zeilen in ergebnis_ports in denen am Anfang der Zeile ein Kommentar vorkommt
und schreibe das Ergebnis in ergebnis_ports_anzahl
Zähle alle Zeilen in ergebnis_ports und zwar nicht mit grep und hänge das Ergebnis in
ergebnis_ports_anzahl an.

Installiere mariadb-server
Finde heraus, wie der Dienst heisst (Hinweis: der mit dem @-Zeichen ist es nicht)
prüfe ob der Dienst nach läuft, starte den Dienst und aktiviere ihn (enable)
Find heraus, ob der Dienst nach aussen lauscht (port offen)

Finde heraus unter welchen Prozess-ID's (1 oder mehrere) der mariadb-server läuft
```