Redhat Linux 9 (Upgrade from Redhat 8)

Agenda

1. appstream /flatpak
    - Overview / Exercise appstream
    - Overview Flatpak
2. Redhat Alternativen
    - Überblick
3. Redhat 9 - New Features
    - Redhat 9 - LivePatching Kernel - now in WebConsole
    - Redhat container tools - new streams
    - Rootless Container, now supported
    - rpm - default compression zstd
4. Redhat 9 - Container Tools
    - Redhat 9 - container tools
5. Changes Filesystem
    - xfs - changes
    - ext4 - changes
    - scp now uses sftp - protocol
6. Changes Network / Security
    - iptables vs. nftables - What has changed
    - iptables vs. nftables - migrate
    - Network Profiles keyfile format
7. Network
    - eBPF & tools
    - MultiPath TCP
8. Future / Network
    - systemd-networkd / .network units)
9. Security
    - Deprecated signing with sha1
    - systemweite Krypto
    - fapolicyd
10. Security (SELinux)
    - Changes selinux in RHEL9
    - Troubleshooting Ports with selinux
    - Troubleshotting Files selinux
11. Upgrade from RHEL 8 to RHEL 9
    - in place upgrade RHEL8->9
12. Containers / Automation
    - Using docker-compose in RHEL 9
    - Capabilities with Ansible/Podman
    - Example alpine with dropped capabilities
13. Tipps & Tricks
    - Ins System reinkommen ohne Passwort und Änderungen vornehmen
14. Anbindung an AD-Server
    - Anbindung an AD-Server
15. Zertifikat - Store (CA)
    - Zertifikat-Store
16. auditd / systemd-coredump / crashkernel
    - Auditing mit auditd
    - Systemd coredump
    - crashkernel deaktivieren
17. Repo erstellen
    - repo erstellen
18. YoPad

- YoPad

Backlog

1. Distributionen
   - Überblick
2. Verzeichnisse und Dateitypen
   - Verzeichnisaufbau
   - Dateitypen
3. Basisbefehle
   - In den Root-Benutzer wechseln
   - Wo bin ich ?
   - Praktische Ausgabe von langen Seiten - less
   - Datei anlegen - touch
   - Autovervollständen * und tab
   - Welches Programm wird verwendet
4. Erweiterte Befehle (Nice to have)
   - Alias Befehle anzeigen
   - Welche Bibliotheken verwendet ein ausführbares Programm
   - Ist ein Befehl extern, alias oder intern
5. Dateien und Verzeichnisse
   - Mit cd im System navigieren
   - Verzeichnisse in Listenansicht mit versteckten Dateien anzeigen -> ls -la
   - Inhalt in Datei schreiben und anhängen
   - Verzeichnisse anlegen
   - Verzeichnisse und Dateien löschen
   - Kopieren/Verschieben/Umbenennen von Dateien und Files
   - Arbeiten mit vi
6. Dateimanipulation/Unix Tools
   - Anfang oder Ende einer Datei/Ausgabe anzeigen
   - cat/head/tail-Beginn/Ende einer Datei anzeigen
   - zcat - Inhalte einer mit gzip komprimierten Datei anzeigen
   - wc - Zeilen zählen
   - Bestimmte Zeilen aus Datei anzeigen - grep
   - Erweiterte Suche mit Grep
   - Finden von files nach Kriterien - find
7. Prozesse
   - Prozesse anzeigen - ps/pstree -p
   - Alle Prozesse eines Dienstes anzeigen
8. Benutzer, Gruppen und Rechte
   - Rechte
   - Dateien für Benutzer und Gruppen
   - Benutzer anlegen
   - sudo Benutzer erstellen
9. Logs/Loganalyse
   - Logfile beobachten
   - Dienste debuggen
   - Rsyslog
   - Journal analysieren
10. Variablen
    - Setzen und verwenden von Variablen
11. Dienste/Runlevel(Targets verwalten)
    - Die wichtigsten systemctl/service
    - Systemctl - timers
    - Gegenüberstellung service etc/init.d/ systemctl
    - Default Editor systemctl setzen
12. Systemd
    - Die wichtigen Tools für die Kommandozeile (ctl)
13. Firewall
    - Arbeiten mit firewalld
14. Systemadministration

appstream /flatpak

Overview / Exercise appstream

- Applikation streams where introduced in Redhat 8

Advantages

- You can switch to a different version
- More new version are introduced, and you can decide which version to use

Disadvantages

- Only one version of the software can be installed at a time

Overview over different software packages and versions

Modules, Stream and profiles

- module: Name of the software (e.g. postgresql)
- stream: The version (e.g. 15)
- profile: Different use cases, e.g. client / server

Walkthrough Postgresql

Step 1: What modules are available ?

```
dnf module list
```

Step 2: List all versions for postgresql

```
dnf module info postgresql
```

Step 3: Try to install a version

```
## This does not work
dnf install @postgresql
```

Step 4: We will decided for a version

- Format for a specific version: dnf install @module:version/profile

```
## for the profile we take the default -> server
dnf install @postgresql:15
```

Step 5: Switch to a newer version

```
dnf module reset postgresql
## this does not yet install the components
dnf list --installed | grep postgresql
```

```
## now install the newer version
dnf install @postgresql:16
dnf list --installed | grep postgresql
```

```
## just to be sure, all modules do have the proper version
dnf distro-sync
```

Step 6: switch back to version 15

```
dnf module reset postgresql
dnf install @postgresql:15
```

```
## now check for the installed version
dnf list --installed | grep postgresql
```

Reference

- https://www.redhat.com/en/blog/introduction-appstreams-and-modules-red-hat-enterprise-linux

Overview Flatpak

```
Flatpak is a universal package format for Linux desktop applications. It is available on most Linux distributions. It allows you
to install and run applications in a sandboxed environment, separate from the rest of the system. This sandboxing gives you more
control over the dependencies of your applications.
```

Reference:

- https://flathub.org/apps/search?q=vlc

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/administering_the_system_using_the_gnome_desktop_environment/assembly_installing-applications-using-flatpak_administering-the-system-using-the-gnome-desktop-environment#installing-flatpak-applications_assembly_installing-applications-using-flatpak

Redhat Alternativen

Überblick

Zur Erklärung binärkompatibel

```
Zwei Betriebssysteme sind binärkompatibel, wenn jedes Programm, das für das eine Betriebssystem kompiliert wurde, ohne erneutes
Kompilieren sofort auf dem anderen Betriebssystem lauffähig ist.
```

Centos 9 Stream

- CentOS Stream dient als Upstream-Entwicklungsplattform für kommende Produktreleases von Red Hat® Enterprise Linux. public development branch for RHEL. Specifically, CentOS Stream 8 is the upstream for the next minor release of RHEL 8, CentOS Stream 9 for the next minor release of RHEL 9, and so on.

Unterschied zu Centos x Linux

- Centos x Linux z.B. Centos 7 Linux, war eine Version, die aus RHEL gebaut wurde.
- Diese Veröffentlichungsweg ist eingestellt (komplett seit 2024)

OracleLinux

```
Oracle Linux Support. Oracle Linux, das 100 % anwendungs-binärkompatibel mit Red Hat Enterprise Linux ist,
```

- Quote: https://www.oracle.com/de/linux/

Rocky

Allgemein

- kostenfrei, von einer non-profit-organisation verwaltet
- Rocky ist am nächsten dran bzgl. der Binärkompabilität.

```
Rocky Linux releases closely follow RHEL releases, usually by days or weeks.
These brief delays are due to the rebuild process and community-driven development.
For example, RHEL 9.3 was released on November 7, 2023, and Rocky Linux 9.3 was released on November 20, 2023.
```

Rocky -> Kompabilität

```
Rocky Linux's goal is to be completely compatible with RHEL,
like CentOS was. The packages are all compiled from the same sources and patches.
```

Rocky and SourceCode

```
In terms of functionality and features, Rocky Linux and RHEL are virtually identical.
Rocky Linux formerly used the RHEL source code to build their own packages
(as did AlmaLinux, Oracle Linux, and many others)
but Red Hat's move to restrict RHEL source code access changed the method by which they maintain compatibility.
```

```
Rocky Linux still maintains 1:1 compatibility, but in a different manner
than AlmaLinux and Oracle Linux.
In a statement from July 29, 2023, Rocky Linux said it obtains
the “source code from multiple sources, including CentOS Stream,
pristine upstream packages, and RHEL SRPMS.”
```

AlmaLinux

Overview

```
Eine Open-Source-Linux-Distribution, die sich im Besitz der Community befindet
und von der Community verwaltet wird, sich auf langfristige Stabilität konzentriert
und eine robuste Plattform für die Produktion bietet. AlmaLinux OS ist binärkompatibel mit RHEL.®
...
Quote from Webpage
```

Reference:

- https://almalinux.org/de/

Redhat 9 - New Features

Redhat 9 - LivePatching Kernel - now in WebConsole

```
Kernel live patch management is also available via the web console to significantly
reduce the complexity of performing critical maintenance.
```

Redhat container tools - new streams

Metapackage for a couple of container - tools

- This holds a couple of packages: Podman, Buildah, Skopeo, CRIU, Udica

Install

```
dnf install -y container-tools
```

```
[root@redhat-node ~]# sudo dnf install container-tools
Subscription Management Repositorys werden aktualisiert.
Letzte Prüfung auf abgelaufene Metadaten: vor 11:20:08 am So 27 Okt 2024 21:28:3
1 CET.
Abhängigkeiten sind aufgelöst.
================================================================================
 Paket            Arch.  Version          Paketquelle                      Größe
================================================================================
Installieren:
 container-tools noarch 1-14.el9         rhel-9-for-x86_64-appstream-rpms 8.3 k
Abhängigkeiten werden installiert:
 podman-docker   noarch 4:4.9.4-13.el9_4 rhel-9-for-x86_64-appstream-rpms 106 k
 podman-remote   x86_64 4:4.9.4-13.el9_4 rhel-9-for-x86_64-appstream-rpms  10 M
 python3-podman  noarch 3:4.9.0-1.el9    rhel-9-for-x86_64-appstream-rpms 178 k
 python3-pyxdg   noarch 0.27-3.el9       rhel-9-for-x86_64-appstream-rpms 108 k
 python3-tomli   noarch 2.0.1-5.el9      rhel-9-for-x86_64-appstream-rpms  37 k
 skopeo          x86_64 2:1.14.5-2.el9_4 rhel-9-for-x86_64-appstream-rpms 8.5 M
 toolbox         x86_64 0.0.99.5-2.el9   rhel-9-for-x86_64-appstream-rpms 2.5 M
 udica           noarch 0.2.8-1.el9      rhel-9-for-x86_64-appstream-rpms  54 k
```

How it works in Redhat 9

2 ways to consume container tools

Way 1: move quickly -> application stream

- Applicaton Stream
- Released every 12 weeks
- For developers and users, who want to access the latest podman, buildah and skopeo

Way 2: stable stream

- Additional Subscription: Extended Update Support (EUS)
- Maintaining a consistent version of podman.
- Support security backports for 2 years.

Notes on podman-docker

```
The podman-docker package replaces the Docker command-line interface and docker-api with the matching Podman commands. Every time
you run a Docker command, the system will actually run a Podman command
```

container - tools

buildah

- Tool zum images bauen

Skopeo

- Hilfskommandos zum interagieren mit einer Registry
- https://github.com/containers/skopeo

UDICA

- SELinux policies
- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/8/html/using_selinux/creating-selinux-policies-for-containers_using-selinux#creating-and-using-an-selinux-policy-for-a-custom-container_creating-selinux-policies-for-containers

CRIU

- It can freeze a running container (or an individual application) and checkpoint its state to disk
- https://criu.org/Main_Page

.container - unit

- Units erstellen, die automatisch einen Container für ein bestimmtes Images startet

Reference:

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html-single/considerations_in_adopting_rhel_9/index#ref_changes-to-containers_assembly_containers

Rootless Container, now supported

Techn Preview in RHEL 8, now stable/supported in RHEL 9

Attention:

- The wording: rootless containers in this case means:
- You can run containers as unprivileged user (e.g. user kurs)
  - the containers themselves can also run privileged (though)

Walkthrough

```
## as unprivileged user e.g. kurs
## when adding a new user with useradd
## this uses is automatically setup to use podman unprivileged
## run image alpine from hub.docker.com and show the id
## after that please delete
## container still runs under root
podman run --rm -it alpine id
```

```
![image](https://github.com/user-attachments/assets/81318db9-9b9a-4a3c-aa3e-4ca38298b357)
```

Reference

- https://docs.redhat.com/de/documentation/red_hat_enterprise_linux/9/html-single/building_running_and_managing_containers/index#proc_setting-up-rootless-containers_assembly_starting-with-containers

rpm - default compression zstd

What is payload

- Payload is the cpio-archive that holds the binaries as an archive (like tar)
- This file is compressed.
  - It used to be xz, now it is zstd (Z Standard)

Why is zstd used ?

- Takes less cpu
- Is faster

Source from redhat

```
RPM now supports the Zstandard (zstd) compression algorithm

In RHEL 9, the default RPM compression algorithm has switched
to Zstandard (zstd). As a result, packages now install faster,
which can be especially noticeable during large transactions.
```

How to test ?

```
## as root
cd /usr/src
dnf download nano
## Which compression is used
rpm -q --queryformat '%{PAYLOADCOMPRESSOR}\n' -p nano*.rpm
```

We are using special variables, what are they ?

- https://rpm-software-management.github.io/rpm/manual/tags.html#packages-with-files

Redhat 9 - Container Tools

Redhat 9 - container tools

Metapackage for a couple of container - tools

- This holds a couple of packages: Podman, Buildah, Skopeo, CRIU, Udica

Install

```
dnf install -y container-tools
```

```
[root@redhat-node ~]# sudo dnf install container-tools
Subscription Management Repositorys werden aktualisiert.
Letzte Prüfung auf abgelaufene Metadaten: vor 11:20:08 am So 27 Okt 2024 21:28:3
1 CET.
Abhängigkeiten sind aufgelöst.
================================================================================
 Paket            Arch.  Version           Paketquelle                      Größe
================================================================================
Installieren:
 container-tools  noarch 1-14.el9          rhel-9-for-x86_64-appstream-rpms 8.3 k
Abhängigkeiten werden installiert:
 podman-docker    noarch 4:4.9.4-13.el9_4  rhel-9-for-x86_64-appstream-rpms 106 k
 podman-remote    x86_64 4:4.9.4-13.el9_4  rhel-9-for-x86_64-appstream-rpms  10 M
 python3-podman   noarch 3:4.9.0-1.el9     rhel-9-for-x86_64-appstream-rpms 178 k
 python3-pyxdg    noarch 0.27-3.el9        rhel-9-for-x86_64-appstream-rpms 108 k
 python3-tomli    noarch 2.0.1-5.el9       rhel-9-for-x86_64-appstream-rpms  37 k
 skopeo           x86_64 2:1.14.5-2.el9_4  rhel-9-for-x86_64-appstream-rpms 8.5 M
 toolbox          x86_64 0.0.99.5-2.el9    rhel-9-for-x86_64-appstream-rpms 2.5 M
 udica            noarch 0.2.8-1.el9       rhel-9-for-x86_64-appstream-rpms  54 k
```

How it works in Redhat 9

2 ways to consume container tools

Way 1: move quickly -> application stream

- Applicaton Stream
- Released every 12 weeks
- For developers and users, who want to access the latest podman, buildah and skopeo

Way 2: stable stream

- Additional Subscription: Extended Update Support (EUS)
- Maintaining a consistent version of podman.
- Support security backports for 2 years.

Notes on podman-docker

```
The podman-docker package replaces the Docker command-line interface and docker-api with the matching Podman commands. Every time
you run a Docker command, the system will actually run a Podman command
```

container - tools

buildah

- Tool zum images bauen

Skopeo

- Hilfskommandos zum interagieren mit einer Registry
- https://github.com/containers/skopeo

UDICA

- SELinux policies
- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/8/html/using_selinux/creating-selinux-policies-for-containers_using-selinux#creating-and-using-an-selinux-policy-for-a-custom-container_creating-selinux-policies-for-containers

CRIU

- It can freeze a running container (or an individual application) and checkpoint its state to disk
- https://criu.org/Main_Page

.container - unit

- Units erstellen, die automatisch einen Container für ein bestimmtes Images startet

Reference:

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html-single/considerations_in_adopting_rhel_9/index#ref_changes-to-containers_assembly_containers

Changes Filesystem

xfs - changes

bigtime && inobtcount

- bigtime. Date support beyond 2038
- inobtcount: Inode btree counters (inobtcount), to reduce mount time on large filesystems.

It looks like this in RHEL 9

```
xfs_info /dev/mapper/rhel*root | grep -e bigtime -e inobtcount
```

```
[root@redhat-master ~]# xfs_info /dev/mapper/rhel*root | grep -e bigtime -e inobtcount
         =                       reflink=1    bigtime=1 inobtcount=1 nrext64=0
[root@redhat-master ~]#
```

Not enabled in Redhat 8

```
xfs_info /dev/mapper/rhel*root | grep -e bigtime -e inobtcount
```

```
[root@rhel8 ~]# xfs_info /dev/mapper/rhel*root | grep -e bigtime -e inobtcount
         =                       reflink=1    bigtime=0 inobtcount=0
```

Downward compability

```
To create a new filesystem that will be compatible with the RHEL 8 kernel, disable these new features by adding -m
bigtime=0,inobtcount=0 to the mkfs.xfs command line. A filesystem created in this way will not support timestamps beyond the year
2038.
```

Migration

- The LEAPP - in place- migration - tool, does NOT do this for you (changing it with xfs_admin)
- You have do it manually, it does not work if filesystem is mounted

```
xfs_admin -O bigtime=1,inobtcount=1 /dev/mapper/rhel-root
```

Reference:

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/considerations_in_adopting_rhel_9/assembly_file-systems-and-storage_considerations-in-adopting-rhel-9#ref_file-systems_assembly_file-systems-and-storage

ext4 - changes

Too small inode size (128) cause 2038 - bug

```
## Diagnostics:
tune2fs -l /dev/<device> | grep "Inode size"
```

Remarks

```
ext4 filesystem now supports timestamps beyond the year 2038

The ext4 filesystem is now supporting timestamps beyond the year 2038.
This feature is fully automatic and does not require any user action to
leverage it. The only requirement is that the inode size is larger than 128 bytes,
which it is by default.
```

Reference:

- https://access.redhat.com/solutions/6983510

scp now uses sftp - protocol

- scp command can still be used to connect
- but in the background the sftp protocol is used

Reference:

https://www.redhat.com/en/blog/openssh-scp-deprecation-rhel-9-what-you-need-know

Changes Network / Security

iptables vs. nftables - What has changed

Overview

- iptables now uses the nftables api under the hood
- iptables - rules will still work, but will be "translated" to nftables backend under the hood
    - but they will not show up under nftables

```
iptables -v
```

```
[root@vbox ~]# iptables --version
iptables v1.8.10 (nf_tables)
[root@vbox ~]#
[root@vbox ~]#
[root@vbox ~]# cat /etc/os-release | grep -e "^NAME" -e "^VERSION="
NAME="Red Hat Enterprise Linux"
VERSION="9.4 (Plow)"
```

Recommended approach - migrate to nftables

- see next document

iptables vs. nftables - migrate

Example: Let us create a rule for iptables and http / ssh

Reference:

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/configuring_firewalls_and_packet_filters/getting-started-with-nftables_firewall-packet-filters?extIdCarryOver=true&sc_cid=701f2000000tyN6AAI#proc_converting-iptables-and-ip6tables-rule-sets-to-nftables_assembly_migrating-from-iptables-to-nftables

Network Profiles keyfile format

Old version

- Network Profiles used to be stored here

```
/etc/sysconfig/network-scripts
```

- This folder and format is deprecated
- New files will not be created there anymore

New version

- no stored here:

```
/etc/NetworkManager/system-connections
```

- ini-format (easily parseable)
- Example:

```
[connection]
id=enp0s8
uuid=d58039f9-d191-4b2a-beda-d3a84b11cd7e
type=ethernet
interface-name=enp0s8

[ethernet]

[ipv4]
method=auto

[ipv6]
addr-gen-mode=eui64
method=auto

[proxy]
```

Migration

- After in-place upgrade still the old files will be present
- migration (will be migrated and removed from /etc/sysconfig/network-scripts)

```
nmcli conn migrate
```

Network

eBPF & tools

Referenzen:

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/8/html/configuring_and_managing_networking/network-tracing-using-the-bpf-compiler-collection_configuring-and-managing-networking#displaying-tcp-connections-added-to-the-kernels-accept-queue_network-tracing-using-the-bpf-compiler-collection
- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/configuring_and_managing_networking/assembly_understanding-the-ebpf-features-in-rhel-9_configuring-and-managing-networking#ref_overview-of-networking-ebpf-features-in-rhel-9_assembly_understanding-the-ebpf-features-in-rhel-9

MultiPath TCP

```
echo "net.mptcp.enabled=1" > /etc/sysctl.d/90-enable-MPTCP.conf
sysctl -p /etc/sysctl.d/90-enable-MPTCP.conf
```

TestPage

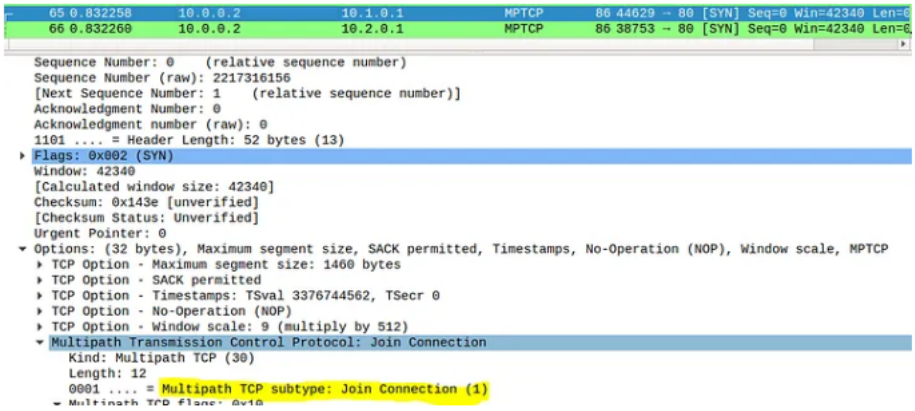
- curl http://www.multipath-tcp.org

Exercise

```
echo "net.mptcp.enabled=1" > /etc/sysctl.d/90-enable-MPTCP.conf
sysctl -p /etc/sysctl.d/90-enable-MPTCP.conf
sysctl net.mptcp.enabled
dnf install -y  mptcpd iperf3
## Listener
mptcpize run iperf3 -s &
while true ; do ss -nti '( dport :5201 )'; done

### Dann im Client ->
## client
mptcpize run iperf3 -c 127.0.0.1 -t 3
nstat -s MPTcp*
```

Joining



Join Connection to initiate a new subflow in MPTCP

Reference

- https://www.mptcp.dev/
- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/configuring_and_managing_networking/getting-started-with-multipath-tcp_configuring-and-managing-networking

Future / Network

systemd-networkd / .network units

Security

Deprecated signing with sha1

Signed Packages

- No problems with packages from redhat
- They are signed with sha256
- https://www.redhat.com/en/blog/rhel-security-sha-1-package-signatures-distrusted-rhel-9

DNSSEC

- https://access.redhat.com/solutions/6955455

ssh

- Problem connections from new to old
- If you're running a mixture of new and old RHEL versions, you may have problems SSHing from new to old

Workaround for connection to old system (RHEL5, RHEL 6)

- https://rwmj.wordpress.com/2022/08/08/ssh-from-rhel-9-to-rhel-5-or-rhel-6/

Enable it on the complete server

```
update-crypto-policies --set DEFAULT:SHA1
```

systemweite Krypto

```
## Welches Profil wird aktuell verwendet
update-crypto-policies --show
```

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/security_hardening/using-the-system-wide-cryptographic-policies_security-hardening#system-wide-crypto-policies_using-the-system-wide-cryptographic-policies

fapolicyd

Einschränkung:

- Standardmäßig darf der Root-Benutzer alle Kommandos ausführen
- d.h. als root funktioniert nicht Übung nicht

Exercise

```
## as root
dnf install -y fapolicyd
systemctl enable --now fapolicyd
cp -a /bin/ls /tmp/ls
```

```
## as unprivileged user
/tmp/ls
## <- now allowed
```

```
## as privileged user
fapolicyd-cli --file add /tmp/ls --trust-file myapp
fapolicyd-cli --update
```

```
## as unprivileged user
/tmp/ls
```

Reference:

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/security_hardening/assembly_blocking-and-allowing-applications-using-fapolicyd_security-hardening#marking-files-as-trusted-using-an-additional-source-of-trust_assembly_blocking-and-allowing-applications-using-fapolicyd

Security (SELinux)

Upgrade from RHEL 8 to RHEL 9

in place upgrade RHEL8->9

Step 1: What is not supported (but the leapp preupgrade will show you !)

- Find out if there is something that is not supported, e.g.
  - Having ansible tower installed (migration process is different)
  - wanting to shift from bios tu uefi boot

Step 2: Vorbereitungsschritte

1. Es sollte kein Ansible/Puppet Änderung am Systme machen
2. Von auch Rhel 7 auf RHEL 8 auch mit LEAPP durchgeführen ?
   - Dann löschen: sudo rm -rf /root/tmp_leapp_py3
3. Ist Abonnenment da ?
   - sudo subscription-manager list --installed -> Status: subskribiert

Step 3: Make backup of system

- In our case, we will make a "Sicherung" in virtualbox
- In addition we will create a clone beforehand

Step 4: Sicherstellen, dass beide Repos aktiviert sind, Stand sperren und update durchführen

```
sudo subscription-manager repos --enable rhel-8-for-x86_64-baseos-rpms --enable rhel-8-for-x86_64-appstream-rpms
sudo subscription-manager release --set 8.10
sudo dnf update
```

Step 5: leapp-upgrade installiere und alle Anwendungssperren entfernen

```
sudo dnf install -y leapp-upgrade
## if you get command not found, this dnf plugin is not installed
## that's o.k. , then you also will have no versionlocks
dnf versionlock clear
```

Step 6: Disable AllowDriftingZone in Firewall

```
cat /etc/firewalld/firewalld.conf | grep -i allowzonedrifting
## if you have yes here, disable -> set to no  (with vi or nano)
```

Step 7: Do the preupdate checks with leapp

```
## it might take its time, so verbose and debug might be a good idea.
## ON MY SYSTEM it TOOK 35 minutes and then 2nd on 17 minutes
sudo leapp preupgrade --debug --verbose --target 9.4
## Report is written to :
## /var/log/leapp/leapp-report.json
## And also writes results to the screen
```

(Optional) Step 8: View report in cockpit

```
dnf install cockpit-leapp
```

Step 9: Upcoming error Processor: Unsupported Family

```
## will adjust the configuration/checks file in leap, because we now this processor is working
## we installed it under RHEL 9 already
## Looks like a typical error in virtualbox
```

- https://access.redhat.com/solutions/7052222

Step 9.5: Upgrading system

```
sudo leapp preupgrade --debug --verbose --target 9.4
```

Step 9.6: Analyze errors: Possible error: cannot open database file

- Database file cannot be opened because of too many open files

```
## default seems to be 1024 - which could be too small
## shows max for open file descriptors
ulimit -n

## rerun command with strace, to see the problems
strace -fttTvyyo /tmp/leapp.strace -s 128 leapp upgrade --debug --verbose --target 9.4
## You will see the errors here
grep "1 EMFILE" /tmp/leapp.strace
```

- https://access.redhat.com/solutions/6878881

Step 9.7: Fix error

```
ulimit -n 16384
## rerun upgrade
leapp upgrade --debug --verbose --target 9.4
```

Step 10: Reboot into initramfs (for update)

- There was a special initramfs created to complete the upgrade

```
reboot
```

Step 11: Post-Upgrade check

```
cat /etc/redhat-release
uname -r
subscription-manager list --installed
subscription-manager release
```

- There are some notes, about it here: https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/upgrading_from_rhel_8_to_rhel_9/verifying-the-post-upgrade-state_upgrading-from-rhel-8-to-rhel-9#verifying-the-post-upgrade-state_upgrading-from-rhel-8-to-rhel-9

Step 12: Post-upgrade cleanup (Part 1)

- We need to do some cleanup

```
##  1. Delete all packages from the exclude list
dnf config-manager --save --setopt exclude=''
```

Step 13: Post-upgrade cleanup (Part 2)

```
## Locate the packages from RHEL 8
rpm -qa | grep -e '\.el[78]' | grep -vE '^(gpg-pubkey|libmodulemd|katello-ca-consumer)' | sort
```

```
## Delete all the packages from RHEL 8
dnf remove $(rpm -qa | grep -e '\.el[78]' | grep -vE '^(gpg-pubkey|libmodulemd|katello-ca-consumer)' | sort)
```

```
dnf remove leapp-deps-el9 leapp-repository-deps-el9
```

Optional: Step 14: Delete related upgrade data

- Eventually, you want to do this later, when everything is o.k.

```
rm -rf /var/log/leapp /root/tmp_leapp_py3 /var/lib/leapp
```

Step 15: Update Kernel Command (set new default)

```
BOOT_OPTIONS="$(tr -s "$IFS" '\n' </proc/cmdline | grep -ve '^BOOT_IMAGE=' -e '^initrd=' | tr '\n' ' ')"
echo $BOOT_OPTIONS > /etc/kernel/cmdline
```

Step 16: Delete existing initramfs for rescue mode and create new one

```
rm /boot/vmlinuz-*rescue* /boot/initramfs-*rescue*
```

```
/usr/lib/kernel/install.d/51-dracut-rescue.install add "$(uname -r)" /boot "/boot/vmlinuz-$(uname -r)"
```

Step 17: Verify new rescure system

```
ls /boot/vmlinuz-*rescue* /boot/initramfs-*rescue*
lsinitrd /boot/initramfs-*rescue*.img | grep -qm1 "$(uname -r)/kernel/" && echo "OK" || echo "FAIL"
```

```
## check if entry in bootmenu refers to the right rescue kernel
grubby --info $(ls /boot/vmlinuz-*rescue*)
```

Step 18: Check and activate security profile

```
## Are there any denials ?
ausearch -m AVC,USER_AVC -ts boot
## set tto enforcing
vi /etc/selinux/config
```

```
## from permissive
.... enforcing
```

```
reboot
```

Reference:

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/upgrading_from_rhel_8_to_rhel_9/index
- https://www.computerweekly.com/de/ratgeber/Wie-man-RHEL-8-auf-RHEL-9-aktualisiert

Containers / Automation

Using docker-compose in RHEL 9

Preparation / Installation

```
dnf -y update
dnf install -y podman git
## version needs to be >=3.0
podman --version
```

```
## for using docker-compose podman.service needs to be running
systemctl status podman.service
systemctl enable --now podman.service
systemctl status podman.service
## enabling and starting podman.service also creates a socket
systemctl status podman.socket
```

```
## docker commands get translated to podman commands under the hood
dnf install -y podman-docker
docker images
```

```
## get latest docker-compose
curl -L https://github.com/docker/compose/releases/download/v2.30.1/docker-compose-linux-x86_64 -o /usr/local/bin/docker-compose
chmod u+x /usr/local/bin/docker-compose
docker-compose
```

```
## now test it with a project
## we do this in the home folder of root
cd
git clone https://github.com/docker/awesome-compose
cd awesome-compose/
ls -la
```

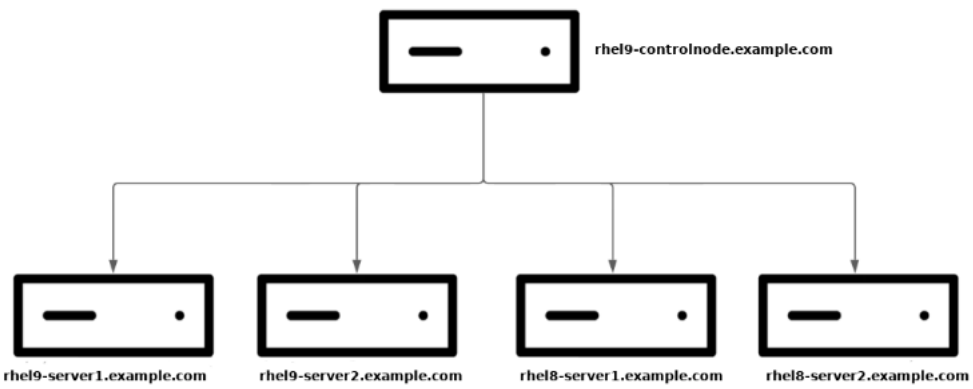
```
cd gitea-postgres/
## -d - daemon-mode -- let's it run in the background
docker-compose up -d
docker-compose images
```

```
docker-compose --help
## stop and delete everything
docker-compose down
docker-compose images
```

Reference:

- https://linuxhistory.com/2023/06/23/podman-how-to-configure-docker-compose-for-rhel9/

Capabilities with Ansible/Podman



Step 1: Configure control-node (on our redhat-master - node)

```
## this alos includes all the necessary ansible packages
sudo dnf install rhel-system-roles
```

```
## Create user
sudo useradd ansible
sudo su - ansible
```

```
## Create public/private key.
## For testing no pass please
ssh-keygen
```

```
## Create ~/.ansible.cfg file
nano ~/.ansible.cfg
```

```
[defaults]
inventory = /home/ansible/inventory
remote_user = ansible

[privilege_escalation]
become = True
become_method = sudo
become_user = root
become_ask_pass = True
```

```
## Create inventory file
nano ~/inventory
```

```
[DE]
redhat-node.training.local ansible_host=192.168.56.108
```

Step 2: on node: configure redhat-node.training.local (our node to deploy)

```
## on redhat-node as root
useradd ansible
usermod -aG wheel ansible
## Set a password and please remember it !
passwd ansible
```

Step 3: on master node:

```
## on maaster-node as ansible user (where we do have the public key
## change your ip accordingly
ssh-copy-id ansible@192.168.56.108
```

```
## Issue a test command
ansible -m ping all
```

Step 4: on master node: build the script

- Ref: https://www.redhat.com/en/blog/automating-podman-rhel-system-roles

```
nano ubi8-httpd-24.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: ubi8-httpd-24
spec:
  containers:
    - name: ubi8-httpd-24
      image: registry.access.redhat.com/ubi8/httpd-24
      ports:
        - containerPort: 8080
          hostPort: 8080
      volumeMounts:
        - mountPath: /var/www/html:Z
          name: ubi8-httpd-24-html
  volumes:
    - name: ubi8-httpd-24-html
      hostPath:
        path: /home/ansible/ubi8-httpd-24-html
```

```
## build the inventory.yml file
nano inventory.yml
```

```
all:
  hosts:
    redhat-node.training.local:
       ansible_host: 192.168.56.108
  vars:
    #podman system role variables:
    podman_firewall:
      - port: 8080/tcp
        state: enabled
    podman_create_host_directories: true
    podman_host_directories:
      "/home/ansible/ubi8-httpd-24-html":
        owner: ansible
        group: ansible
        mode: "0755"
    podman_kube_specs:
      - state: started
        run_as_user: ansible
        run_as_group: ansible
        kube_file_src: ubi8-httpd-24.yml

    #cockpit system role variables:
    cockpit_packages:
```

```
      - cockpit-podman
    cockpit_manage_firewall: true
```

```
nano system_roles.yml
```

```
- name: Run podman RHEL system role
  hosts: all
  roles:
    - redhat.rhel_system_roles.podman

- name: Run cockpit RHEL system role
  hosts: all
  roles:
    - redhat.rhel_system_roles.cockpit
```

```
ansible-playbook -i inventory.yml -b system_roles.yml
```

Reference:

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/automating_system_administration_by_using_rhel_system_roles/assembly_preparing-a-control-node-and-managed-nodes-to-use-rhel-system-roles_automating-system-administration-by-using-rhel-system-roles
- https://www.redhat.com/en/blog/automating-podman-rhel-system-roles

Example alpine with dropped capabilities

Requirements: Last exercise 01-ansible.....

Step 1: on master: Simple pod

```
cd
mkdir podtest
cd podtest
```

```
nano pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: alpine
spec:
  containers:
    - name: cont
      image: alpine
      securityContext:
        capabilities:
          drop: ["ALL"]
```

```
nano inventory.yml
```

```
all:
  hosts:
    redhat-node.training.local:
       ansible_host: 192.168.56.108
  vars:
    podman_kube_specs:
      - state: started
        run_as_user: ansible
        run_as_group: ansible
        kube_file_src: pod.yml
```

```
nano system_roles.yml
```

```
- name: Run podman RHEL system role
  hosts: all
  roles:
    - redhat.rhel_system_roles.podman
```

```
ansible-playbook -i inventory.yml -b system_roles.yml
```

Step 2: on node: check capabilities

```
### Step 5: On node
```

we will enter the container and look for capabilities

They are all dropped.

podman exec -it alpine-cont cat /proc/1/status | grep -i cap

```
## Tipps & Tricks

## Anbindung an AD-Server

### Anbindung an AD-Server

  *
https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/integrating_rhel_systems_directly_with_windows_active_direc
rhel-systems-directly-to-ad-using-sssd_integrating-rhel-systems-directly-with-active-directory#using-posix-attributes-defined-in-
active-directory_connecting-directly-to-ad

## Zertifikat - Store (CA)

### Zertifikat-Store

  * https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/8/html/securing_networks/using-shared-system-
certificates_securing-networks#adding-new-certificates_using-shared-system-certificates

## auditd / systemd-coredump / crashkernel

### Auditing mit auditd


### Example syscal rule
```

-a exit,always -F arch=b64 -F euid=0 -S execve -k audit-wazuh-c -a exit,always -F arch=b32 -F euid=0 -S execve -k audit-wazuh-c

```

```

after that

augenrules --load

```
  * Ref: https://wazuh.com/blog/monitoring-root-actions-on-linux-using-auditd-and-wazuh/
  * Ref: https://access.redhat.com/solutions/36278
  * Ref: https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/6/html/security_guide/sec-
defining_audit_rules_and_controls_in_the_audit.rules_file#sec-Defining_Audit_Rules_and_Controls_in_the_audit.rules_file

### Systemd coredump
```

systemctl list-units | grep core systemctl status systemd-coredump.socket

systemd-coredump maskieren

Dadurch wird er nicht verwendet

Deaktivieren

systemctl mask systemd-coredump.socket systemctl stop systemd-coredump.socket qq

Aktivieren

systemctl unmask systemd-coredump.socket systemctl start systemd-coredump.socket

######### Teil 2 - verwenden -> als root

Terminal 1: root

dnf install yum-utils debuginfo-install -y vim-debuginfo

config für systemd-coredump

/etc/systemd/system.conf

##DumpCore=yes DefaultLimitCore=infinity

Terminal 2: als kurs

ulimit -S -c unlimited > /dev/null 2>&1 vim testfile sleep 3 & kill -SEGV $!

was reinschreiben ohne speichern, d.h. vim offen lassen

Terminal 1: (root)

process id von vim rausfinden

ps ax | grep testfile kill -s SIGSEGV 41262 coredumpctl

```
### crashkernel deaktivieren

  * https://access.redhat.com/solutions/7070808

## Repo erstellen

### repo erstellen

  * https://www.percona.com/blog/how-to-create-your-own-repositories-for-packages/

## YoPad

### YoPad
```

Redhat 9 - Update

Documentation

https://github.com/jmetzger/training-redhat9update

Analyse - Server

1. Welche Interfaces ip a ip route top
   iostat

Übung 3.4

systemctl list-units | grep core systemctl status systemd-coredump.socket

systemd-coredump maskieren

Dadurch wird er nicht verwendet

Deaktivieren

systemctl mask systemd-coredump.socket systemctl stop systemd-coredump.socket qq

Aktivieren

systemctl unmask systemd-coredump.socket systemctl start systemd-coredump.socket

######### Teil 2 - verwenden -> als root

Terminal 1: root

dnf install yum-utils debuginfo-install -y vim-debuginfo

config für systemd-coredump

/etc/systemd/system.conf

##DumpCore=yes DefaultLimitCore=infinity

Terminal 2: als kurs

ulimit -S -c unlimited > /dev/null 2>&1 vim testfile sleep 3 & kill -SEGV $!

was reinschreiben ohne speichern, d.h. vim offen lassen

Terminal 1: (root)

process id von vim rausfinden

ps ax | grep testfile kill -s SIGSEGV 41262 coredumpctl

Übung 3.4

Übung 3.3.

https://github.com/jmetzger/training-redhat9update/blob/main/feature/filesystems-xfs/bigtime-inobtcount.md

Übung 3.2

https://github.com/jmetzger/training-redhat9update/blob/main/feature/cgroups-v2/overview.md

Documentation - Tag 3 / Übung 3.1

https://github.com/jmetzger/training-redhat9update/blob/main/network/mptcp/overview.md

echo "net.mptcp.enabled=1" > /etc/sysctl.d/90-enable-MPTCP.conf sysctl -p /etc/sysctl.d/90-enable-MPTCP.conf sysctl net.mptcp.enabled dnf install -y mptcpd iperf3

Listener

mptcpize run iperf3 -s & while true ; do ss -nti '(dport :5201)'; done

Dann im Client ->

client

mptcpize run iperf3 -c 127.0.0.1 -t 3 nstat -s MPTcp*

Documentation - Tag 1

https://github.com/jmetzger/training-redhat9update/blob/main/distros/overview-in-comparison-to-rhel-9.md https://github.com/jmetzger/training-redhat9update/blob/main/network/systemd-networkd/overview.md https://github.com/jmetzger/training-redhat9update/blob/main/network/ipv6/overview.md

Documentation - Tag 2

https://github.com/jmetzger/training-redhat9update/blob/main/feature/redhat-container-tools/overview.md https://github.com/jmetzger/training-linux-sicherheit-und-haertung/blob/main/secureboot/06-encrypt-data-with-luks-tpm.md

https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/considerations_in_adopting_rhel_9/index

https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/security_hardening/using-the-system-wide-cryptographic-policies_security-hardening#using-the-system-wide-cryptographic-policies_security-hardening

Übung 2.4.

training ALL=(ALL) /usr/bin/systemctl reload sshd,/usr/bin/chown useradd training : passwd training su - training cdcat /etioas

Übung 2.3

1. Reboot und in bootmanager gehen (ESC) -> e für edit der 1. Zeile
2. Zeile linux, und schreiben ans Ende: init=/bin/bash
3. CTRL x
4. mount -o remount,rw /
5. Passwort ändern passwd
6. touch /.autorelabel
7. -> vm stopen/schliessen
8. neu starten
9. einloggen mit root mit neuem Passwort

Übung 2.2

https://github.com/jmetzger/training-redhat9update/blob/main/feature/rpm-zstd/overview.md

Übung 2.1

Make redhat 8 great again https://github.com/jmetzger/training-redhat9update/blob/main/upgrade/in-place/step-by-step.md#step-96-analyze-errors-possible-error-cannot-open-database-file

Übung 1.2

in-place upgrade RHEL 8.10 ->> 9.4

https://github.com/jmetzger/training-redhat9update/blob/main/upgrade/in-place/step-by-step.md

Übung 1.1

https://github.com/jmetzger/training-redhat9update/blob/main/application_streams/overview.md

Default Target setzen

systemctl set-default multi-user.target reboot

in kernel param

systemd.unit=multi-user.target

Target während des Betriebs

systemctl isolate multi-user.target systemctl set-default multi-user.target reboot

Ip - Adresse anzeigen

ip a show enp0s8

```
## Distributionen

### Überblick


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

#### Redhat-Familie
```

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

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

ab 2022 sehr interessant bzw. unabdingbar

Rocky Linux Alma Linux

```
#### Debian Familie
```

Debian Ubuntu. - dpkg / apt Mint

```

#### SuSE - Familie
```

SLES (SuSE Linux Enterprise) OpenSuSE

```

### Distris zur Sicherheitsüberprüfung / Hacken
```

Kali Linux Parrot. - Distributionen zum Hacken

```

### Live-DVD (Linux ohne Installation)

  * Knoppix - Live DVD - brauche nicht installieren


### Spezial-Linuxe, z.B. für Router
```

OpenWRT DDWRT

```

### Seite mit Übersicht aller Linux-Distros

  * https://distrowatch.com/

## Verzeichnisse und Dateitypen

### Verzeichnisaufbau


### /etc

  * Verzeichnis für 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 (Für Kommunikation von client zu server / server zu client) auf der gleichen Maschine

```
## Basisbefehle

### In den Root-Benutzer wechseln
```

einloggen als normaler Benutzer z.B. benutzer: kurs (wenn ich unter kurs eingeloggt bin)

sudo su -

eingeben des Passworts des Benutzers

```
### Wo bin ich ?
```

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

pwd - Print working directory

pwd

```
### Praktische Ausgabe von langen Seiten - less

### Open a file with less
```

less /etc/services

Why ?

Leichtere Navigation

```
### Pipen mit less (ausgabe an less schicken)
```

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

```
### Suchen in less
```

##Innerhalb von less /suchbegriff + RETURN

nächstes Suchergebnis

n

voriger Suchergebnis

N

```
###  Springen ans Ende/an den Anfang
```

Innerhalb von less

ans Ende

G

an den Anfang

1g

zu einer bestimmten Zeile (Zeile 5)

5g

```
### In die Hilfe rein
```

h

wieder raus

q

```
### Datei anlegen - touch
```

touch dateiname

```
### 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 # bei mehreren Einträgen echo todol # bei einem weiteren Eintrag

```
### Welches Programm wird verwendet
```

Sucht in der Pfad-Variablen $PATH nach dem programm

und zeigt ersten Fund --> d.h. dieses Programm würde ausgeführt

which false

```
## Erweiterte Befehle (Nice to have)

### Alias Befehle anzeigen

### Alias anzeigen
```

keine wirkliche Befehle, sondern nur andere Schreibweise/Abkürzungen

kann u.U. so auf anderen Distris nicht vorhanden sein

alias

```
### Alias - Befehl in der Session setzen
```

Achtung, existiert nicht nacht schliessen der Session

alias l3='ls -la | head -n 3'

```
### Alias-Befehl aufheben/löschen (unalias)
```

unalias l3

```
### Welche Bibliotheken verwendet ein ausführbares Programm
```

ldd /usr/bin/ls

```
### Ist ein Befehl extern, alias oder intern

## 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 !!

```
### Rechte behalten bei kopieren
```

-a macht das

cp -a todo.txt todoneu.txt

ohne -a werden symbolische links aufgelöst und die Rechte des ausführenden Nutzers gesetzt

cp ab cd

Verzeichnisse kopieren

cp -a /etc /etc3

```
### Arbeiten mit vi


### Zeilennummern aktivieren für alle
```

Centos

##/etc/vimrc

am ende

set number

Ubuntu

/etc/vim/vimrc.local

set number

```
### vimtutor
```

Interactives Tutorial zum Lernen von vi

Wichtigste Befehle

vimtutor # sollte bereits mit vi installiert worden sein.

```
### Wichtigste Aktionen
```

1. **Öffnen 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 x und enter

oder

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

```
### Virtual Mode
```

v Zeichenweise markieren einschalten V Zeilenweise markieren einschalten STRG + v Blockweise markieren

mit Cursortasten auswählen / markieren

Dann:

x # Löschen des markierten Bereichs

```
### Zeilen löschen im Normalmodus (Interactiver Modus)
```

ESC + dd # eine Zeile löschen

letzte Aktion rückgängig machen

ESC + u # eigentlich reicht 1x Escape

mehrere Zeilen löschen z.B. 1000

ESC + 1000dd # ESC - Taste drücken, dann 1000 eingeben, dann dd (sie sehen die 1000 nicht auf dem Bildschirm)

```
### Neues Fenster und Fenster wechseln
```

innerhalb von vi

ESC + : -> vsplit # aktuelles Fenster wird kopiert

Fenster wechseln

ESC + : wincmd w

oder

STRG + w w

```
### Cheatsheet

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

## Dateimanipulation/Unix Tools

### Anfang oder Ende einer Datei/Ausgabe anzeigen


### Die ersten 10
```

die ersten 10 Zeilen einer Datei anzeigen

head /etc/services

Alternative 1

cat /etc/services | head

die letzten 10 Zeilen

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

einer ausgabe // erste 10 Zeilen eines Verzeichnislistings

ls -la | head

```
### Die ersten 20
```

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

```
### Die letzten 20
```

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

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


### cat mit 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 ersten 10 Zeilen / Variante mit cat
```

cat services | head

mit zeilennummen

cat -n services | head

```
### Die letzten -x Zeilen anzeigen
```

die letzten 10 Zeilen

tail /etc/services

die letzten 40 Zeilen

tail -n 40 /etc/services

```
### Ausgabe der letzten 10 Zeilen
```

cat /etc/services | tail

```
### zcat - Inhalte einer mit gzip komprimierten Datei anzeigen

### wc - Zeilen zählen


### Datei
```

wc -l /etc/services

```
### Zeilen aus Befehl
```

ls -la | wc -l

```
### Bestimmte Zeilen aus Datei anzeigen - grep


### Beispiele
```

alle Zeilen in den tcp vorkommt

cat /etc/services | grep tcp

alle Zeilen in denen tcp nicht vorkommt

cat /etc/services | grep -v tcp

alle Zeilen in denen tcp nicht vorkommt

egal ob gross oder klein geschrieben.

cat /etc/services | grep -iv TCP

cat /etc/services | grep '#' cat /etc/services | grep "#" cat /etc/services | grep "^#"

alle Zeilen, die am Anfang der Zeile kein # haben

cat /etc/services | grep -v "^#" cat /etc/services | grep -v "^#" > /root/services cat /etc/services | grep -v "^#" | head -n 20

cat /etc/services | grep -v "s$"

alle Zeilen die als letztes Zeichen ein s haben

cat /etc/services | grep "s$"

```
### Recursive Suchen (grep -r) - Schweizer Taschenmesser
```

grep -r "PermitRootLogin" /etc

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

```

```

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

```
### Mengeangabe
```

Achtung unbedingt egrep oder grep -E verwenden

cat /root/namen AxB nix AxB nix abc nix a nix

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

```

```

echo "ab nix" >> /root/namen

Mindestens 2 Zeichen

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

```
### Nach Zahlen Suchen
```

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

```
### Cheatsheets

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


### Ref:

  * https://www.cyberciti.biz/faq/grep-regular-expressions/
### Finden von files nach Kriterien - find

### Simple find command
```

find directories with specific name

find / -name tmpfiles.d -type d

```
## Prozesse

### Prozesse anzeigen - ps/pstree -p

### Prozesse anzeigen
```

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

```
### systemctl (läuft Dienst)
```

systemctl status sshd

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

pstree -p

```
### Alle Prozesse eines Dienstes anzeigen
```

inkl header - 2 Befehle getrennt durch ';'

ps aux | head -n 1; ps aux | grep mysqld | grep -v 'grep'

Ausgabe

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND mysql 16938 0.0 1.1 1778456 94776 ? Ssl 09:51 0:00 /usr/libexec/mysqld --basedir=/usr

```
## Benutzer, Gruppen und Rechte

### Rechte


### Arten
```

r = Lesen w = Schreiben x = Ausführen

```
### Für welchen Bereich ?
```

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

```
### Aufbau triple
```

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

rwx | rw- | r--

u g o

421 | 42- | 4--

7 | 6 | 4

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

chmod g+w,o+r testfile

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

chmod 777 testfile

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

chmod -R 777 testverzeichnis

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

cd /etc cat passwd cat shadow cat group

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

```
### Benutzer anlegen


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

for shell script

useradd

for admins interactive

adduser

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

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

adduser newuser usermod -aG sudo newuser

testing

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

need to enter password here

sudo su -

```
## Logs/Loganalyse

### Logfile beobachten
```

Terminal 1

tail -f /var/log/syslog

Terminal 2 - write to logfile e.g.

logger meine_nachricht

```
### Dienste debuggen


### Walkthrough
```

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

systemctl restart mariadb.service

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

systemctl status mariadb.service

Nicht fündig-> Schritt 2:

jourrnalctl -xe

Nicht fündig -> Schritt 3:

-e springt ans Ende des Pages

journalctl -e -u mariadb.service

Nicht fündig -> Schritt 4:

Spezifisches Log von Dienst suchen

und evtl. LogLevel von Dienst hochsetzen

z.B. bei mariadb (durch Internetrecherche herausfinden)

less /var/log/mariadb/mariadb.log

oder schneller

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

also auch z.B. ERROR

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

Nicht fündig -> Schritt 5

Allgemeines Log

REdhat/Centos

/var/log/messages

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

Wie bei walkthrough aber ab Schritt 4

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

cd /var/log/mysql

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

cat error.log | grep -i error

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

You know the wrong config value, but not

where it is (in which file)

assuming gummitulpe is the wrong config value

grep -r gummitulpe /etc

Ausgabe

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

```
### Rsyslog


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

/etc/rsyslog.conf

udp

. @192.168.10.254:514

tcp

. @@192.168.10.254:514

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

```
### Journal analysieren


### Show all boots
```

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

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

journalctl -b

```
### Journal persistent

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

persistent setzen

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

Dies ist auch der Default - Fall

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

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

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

in /etc/systemd/journald.conf

SystemMaxUse=1G

```
### journalctl
```

ubuntu

journalctl -u ssh

```
## Variablen

### Setzen und verwenden von Variablen
```

DATEINAME=/etc/services echo $DATEINAME

Werte hochzählen

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

cat $DATEINAME

wird nicht der Inhalt verwendet sondern der Name $DATEINAME

cat '$DATEINAME' cat "$DATEINAME"

Befehl ausführen und Rückgabewert anzeigen

date echo $?

Wert aus ausgeführtem Befehl in Variable schreiben

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

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

### Die wichtigsten systemctl/service


### Welche Dienste sind aktiviert ?
```

systemctl list-units --type=service

oder

systemctl list-units -t service

oder

systemctl -t service

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

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

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

systemctl start httpd systemctl stop httpd

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

systemctl cat sshd.service

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

systemctl status sshd systemctl status sshd.service

ältere Variante

service sshd status

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

d.h. dienst wird beim nächsten Boot nicht gestartet

systemctl disable sshd.service

oder

systemctl disable sshd

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

systemctl is-enabled sshd.service echo $?

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

systemctl enable sshd.service

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

z.B. für Apache2

systemctl cat apache2.service

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

cd /lib/systemd/system root@ubuntu2004-104:/lib/systemd/system# ls -la run*target lrwxrwxrwx 1 root root 15 Jan 6 20:47 runlevel0.target -> poweroff.target lrwxrwxrwx 1 root root 13 Jan 6 20:47 runlevel1.target -> rescue.target lrwxrwxrwx 1 root root 17 Jan 6 20:47 runlevel2.target -> multi-user.target lrwxrwxrwx 1 root root 17 Jan 6 20:47 runlevel3.target -> multi-user.target lrwxrwxrwx 1 root root 17 Jan 6 20:47 runlevel4.target -> multi-user.target lrwxrwxrwx 1 root root 16 Jan 6 20:47 runlevel5.target -> graphical.target lrwxrwxrwx 1 root root 13 Jan 6 20:47 runlevel6.target -> reboot.target

```
### Welche Dienste sind 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)

systemctl daemon-reload systemctl status

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

Default runlevel/target auslesen

systemctl get-default

in target wechseln

systemctl isolate multi-user

Default target setzen (nach start/reboot)

systemctl set-default multi-user

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

Redhat / centos

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

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

systemctl mask apache2

kann jetzt gestartet werden

systemctl start apache2

de-maskieren

systemctl unmask apache2

kann wieder gestaret werden

systemctl start apache2

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

Status eines Dienstes überprüfen

service sshd status systemctl status sshd

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

systemctl list-units -t service

für apache

systemctl list-units -t service | grep ^apache

die Abkürzung

systemctl -t service | grep ^apache

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

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

Dienst aktivieren

systemctl enable apache2

Ist Dienst aktiviert

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

Dienst deaktivieren (nach Booten nicht starten)

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

Rebooten des Servers

verweist auf systemctl

reboot systemctl reboot shutdown -r now

Halt (ohne Strom ausschalten)

halt systemctl halt shutdown -h now

Poweroff

poweroff systemctl poweroff

```
### systemctl Cheatsheet

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



### Systemctl - timers


### Show all timers
```

alle Timer anzeigen

systemctl list-timers

```
### How ?

 * .timer and .service file next to each other

### Example ?
```

timer - file

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

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

SPDX-License-Identifier: LGPL-2.1+

This file is part of systemd.

systemd is free software; you can redistribute it and/or modify it

under the terms of the GNU Lesser General Public License as published by

the Free Software Foundation; either version 2.1 of the License, or

(at your option) any later version.

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

[Timer] OnBootSec=15min OnUnitActiveSec=1d

Service - file

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

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

SPDX-License-Identifier: LGPL-2.1+

This file is part of systemd.

systemd is free software; you can redistribute it and/or modify it

under the terms of the GNU Lesser General Public License as published by

the Free Software Foundation; either version 2.1 of the License, or

(at your option) any later version.

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

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

```
### 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

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

In der Session

export SYSTEMD_EDITOR=vim

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

export SYSTEMD_EDITOR=vim

```
## Systemd

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

Oben die wichtigsten

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

```
## Firewall

### Arbeiten mit firewalld


### Install firewalld
```

on centos/redhat firewalld should installed

systemctl status firewalld

if not, just do it

yum install firewalld

```


### Is firewalld running ?
```

is it set to enabled ?

systemctl status firewalld firewall-cmd --state

```
### Command to control firewalld

  * firewall-cmd

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

Step1: do it persistent -> written to disk

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

Step 2: + reload firewall

firewall-cmd --reload

```
### Zones documentation

man firewalld.zones

### Zones available
```

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

```
### Active Zones
```

firewall-cmd --get-active-zones

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

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

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

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

```
### Default Zone
```

if not specifically mentioned when using firewall-cmd

.. add things to this zone

firewall-cmd --get-default-zone public

```
### Show services
```

firewall-cmd --get-services

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

Example ssh

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

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

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

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

add port

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

remove port

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

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

firewall-cmd --get-icmptypes

none present yet

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

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

Documentation

man firewalld.richlanguage

throttle connectons

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

port forwarding

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

list only the rich rules

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

persist all runtime rules

firewall-cmd --runtime-to-permanent

```
### References

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

## 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

server2 client

as user kurs

ssh-keygen

set password set important

ssh-copy-id kurs@server1

Now you can login with public/private key

ssh kurs@server1

```
## Partitionierung und Filesystem

### parted and mkfs.ext4


### Walkthrough
```

Schritt 1: Platte in virtualbox oder gui-interface anlegen

Schritt 2: Platte identifizieren

lsblk

Schritt 3: Platte partitionieren

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

Step 3 - reboot

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

Dann zählt er nicht weiter runter und bootmenu bleibt stehen.

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

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

Step 4 - be happy

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

uname -a

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


### Walkthrough
```

show kernel modules

lsmod

kernel - module entladen

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

damit wieder laden

modprobe psmouse lsmod | grep psmouse # now present

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

kernel version is used, find out kernel version with

uname -a

cd /lib/modules/5.4.0-66-generic

e.g. psmouse

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

```
## Hilfe

### Hilfe zu Befehlen


### Möglichkeiten der Hilfe
```

anhand von ps

vi -h ps --help man ps info ps

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

Beispiel ls

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

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

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

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

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

```
## Grafische Oberfläche und Installation

### X-Server - Ausgabe auf Windows umleiten


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

### Installations-Images-Server

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

## Wartung und Aktualisierung

### Aktualisierung des Systems
```

apt update apt upgrade apt dist-upgrade

oder geht auch auf älteren Systemen

apt-get update apt-get upgrade apt-get dist-upgrade

```
### Paketmanager yum


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

yum list

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

yum list --installed

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

yum list | grep mariadb

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

yum list --installed | grep mariadb

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

yum search mariadb

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

yum info mariadb-server

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

Beispiel sealert

yum whatprovides sealert

```
### Cheatsheet

  * https://access.redhat.com/sites/default/files/attachments/rh_yum_cheatsheet_1214_jcs_print-1.pdf

### Archive runterladen und entpacken
```

Walkthrough

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

Schritt 2:

cd /usr/src

falsche Dateiname -> umbenannt.

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

Schritt 3: Sicherheitsverzeichnis anlegen und entpacken

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

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

## Firewall und ports

### firewalld


### Install firewalld
```

on centos/redhat firewalld should installed

systemctl status firewalld

if not, just do it

yum install firewalld

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

is it set to enabled ?

systemctl status firewalld firewall-cmd --state

```
### Command to control firewalld

  * firewall-cmd

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

Step1: do it persistent -> written to disk

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

Step 2: + reload firewall

firewall-cmd --reload

```
### Zones documentation

man firewalld.zones

### Zones available
```

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

```
### Active Zones
```

firewall-cmd --get-active-zones

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

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

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

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

```
### Default Zone
```

if not specifically mentioned when using firewall-cmd

.. add things to this zone

firewall-cmd --get-default-zone public

```
### Show services
```

firewall-cmd --get-services

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

Example ssh

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

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

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

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

add port

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

remove port

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

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

firewall-cmd --get-icmptypes

none present yet

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

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

Documentation

man firewalld.richlanguage

throttle connectons

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

port forwarding

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

list only the rich rules

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

persist all runtime rules

firewall-cmd --runtime-to-permanent

```
### References

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

### Scannen und Überprüfen mit telnet/nmap

## Netzwerk/Dienste

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


### Walkthrough
```

Ip nicht gesetzt - kurzfristig eine IP holen

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

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

Zeigt alle ports an auf die gelauscht wird (ipv4)

lsof -i

alternative

netstat -tupel

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

Achtung: richtigen profilnamen verwenden

einsehbar über nmtui oder

nmcli conn show

z.B. wenn enp0s9 als profil vorhanden ist

nmtui-edit enp0s8

```
### Netzwerkinterface auf der Kommandozeile einrichten


### Verbindungen anzeigen
```

nmcli connection show

or

nmcli conn show

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

muss in der Liste sichtbar sein

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

verbindung neu hochziehen

nmcli con up enp0s9

verbindungseigenschaften anzeigen

nmcli con show

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

muss in der Liste sichtbar sein

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

verbindung neu hochziehen

nmcli con up enp0s9

verbindungseigenschaften anzeigen

nmcli con show

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

muss in der Liste sichtbar sein

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

verbindung neu hochziehen

nmcli con up enp0s9

verbindungseigenschaften anzeigen

nmcli con show

is ip gesetzt ?

ip a

```
### Ref:
```

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

### Scannen mit nmap


### Scan Range
```

nmap -PE 192.168.1.2-5

```
## Podman

### Podman Walkthrough


### Aufbau (Wirkweise)

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

### Walkthrough
```

runtergeladenen images

podman images

image von online ziehen (registry)

Sucht bei redhat danach bei docker.io

podman pull alpine:latest

Image ist jetzt lokal vorhanden

podman images

Container mit diesem image starten

podman run --name=myalpine alpine

Prozess läuft nicht mehr, da bereits beendet

podman ps

hiermit werden alle prozesse angezeigt auch die beendeten.

podman ps -a

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

podman rm 08

liste der container ist jetztleer

podman ps -a

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

das sind die Optionen -i -t

podman run -it --name=myalpine2 alpine

```
### Walkthrough II
```

interactive mit terminal und detached

Detached - es läuft weiter im hintergrund

podman run -dit --name=myalpine3 alpine

in maschine reinwechseln, Kommanda ls -la ausführen

danach wieder raus

podman exec -it myalpine3 ls -la

podman ps -a

geht nicht, weil es im container keine bash gibt

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

podman exec -it myalpine3 bash

einen sh - befehl gibt in jedem Linux

dieser verweist auf die aktuelle Shell

podman exec -it myalpine3 sh

Die Ausgabe des ersten Befehls wird geloggt

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

Logs anzeigen

podman logs myalpine4

```
### Configuration abfragen
```

Alle Konfigurationen

podman inspect myalpine3

oder container id

podman inspect a23e

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

```
### Aufräumen (tabula rasa)
```

alle container und die, die noch laufen, vorher stoppen

podman rm -a --force

alle runtergeladenen images löschen

podman rmi -a

```
### Image bauen
```

mkdir myimage cd myimage

vi Dockerfile beispiel ubuntu mit folgendem Inhalt

FROM ubuntu:20.04

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

ENV NEW_MODE laola ENV TRAINING_VERSION 1.0

FROM centos:latest

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

choose any name for the image with -t

does not need to be the directory name

podman build -t myimage .

image als Basis für einen container verwenden

podman run -dit --name mycontainer myimage

Now work in the container if you want

podman exec -it mycontainer bash

do whatever you want in the container

e.g. env

```
## SELinux (Linux härten)

### SELinux


### sestatus

 * Zeigt an, obwohl selinux aktiviert und wie

### Modi

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

### Persistente Konfiguration
```

/etc/selinux/config

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

ls -laZ

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

als root

cd / touch .autorelabel reboot

Achtung relabeln kann dauern !!! durchaus 5 Minuten

```
## Tools/Verschiedens

### Remote Desktop für Linux / durch Teilnehmer getestet

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

### Warum umask 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

```
### Example with date and if

### Example with function and return value

### Example with test and if

### Example log function

### Example Parameter auslesen

## Timers/cronjobs

### Cronjob - hourly einrichten


### Walkthrough
```

cd /etc/cron.hourly

nano datum

wichtig ohne Endung

Job wird dann um 17 nach ausgeführt ?

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

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

Abwarten, Tee trinken

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

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

```
## Literatur

### Literatur


### Literatur

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

### Linux Sicherheit
```

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

### Cheatsheet

  * [Cheatsheet bash](https://www2.icp.uni-stuttgart.de/~icp/mediawiki/images/b/bd/Sim_Meth_I_T0_cheat_sheet_10_11.pdf)
  * [..ansonsten Google :o)](https://www.google.com/search?q=bash+cheatsheet)

### Bash - Programmierung

  * [Bash Programmierung](https://tldp.org/LDP/Bash-Beginners-Guide/html/)
  * [Bash Advanced Programmierung](https://tldp.org/LDP/abs/html/loops1.html)
```