# Content <!-- omit in toc -->

- [Basics](#basics)

- [File Hierarchy Standard (FHS)](#file-hierarchy-standard-fhs)

- [Commands](#commands)

- [Globs (Wildcards)](#globs-wildcards)

- [Regex](#regex)

- [Stream redirection](#stream-redirection)

- [Disk and File System Management](#disk-and-file-system-management)

- [General Disk Manipulation (non-LVM)](#general-disk-manipulation-non-lvm)

- [LVM](#lvm)

- [Automatic mounting (/etc/fstab)](#automatic-mounting-etcfstab)

- [Text Readers & Editors](#text-readers--editors)

- [Less](#less)

- [VI](#vi)

- [User and Group Management](#user-and-group-management)

- [File System Permissions](#file-system-permissions)

- [SSH](#ssh)

- [IPv6 (disable)](#ipv6-disable)

- [Cronjobs](#cronjobs)

- [Package Management](#package-management)

- [RPM](#rpm)

- [YUM](#yum)

- [Services (systemd)](#services-systemd)

- [NFS](#nfs)

- [Samba](#samba)

- [FTP](#ftp)

- [MySQL Server](#mysql-server)

- [Other](#other)

# Basics

## File Hierarchy Standard (FHS)

| Path | Content |

| ------ | ----------------------------------- |

| /bin | Binaries (User) |

| /boot | Static boot loader files |

| /etc | Host specific configs |

| /lib | Shared libraries and kernel modules |

| /sbin | Binaries (System/root) |

| /var | Varying files (e.g. Logs) |

| /usr | 3rd party software |

| /proc | Pseudo file system |

| /sys | Pseudo file system |

| /mnt | Mountpoint for internal drives |

| /media | Mountpoint for external drives |

| /home | User homes |

| /run | PID files of running processes |

## Commands

\*\*File System Commands\*\*

| Command | Param | Description |

| ------- | ---------------- | ------------------------------------------------- |

| `cd` | `-` | Navigate to last dir |

| | `~` | Navigate to home |

| | `~username` | Navigate to home of specified user |

| `pwd` | | Print working dir |

| `ls` | | Print dir content |

| | `-l` | Format as list |

| | `-a` | Show hidden items (`-A` without `.` and `..`) |

| | `-r` | Invert order |

| | `-R` | Recurse |

| | `-S` | Sort by size |

| | `-t` | Sort by date modified |

| `mkdir` | `-p` | Create dir with parents |

| `cp` | `-r` | Copy dir |

| `rmdir` | `-p` | Remove dir and empty parents |

| `rm` | `-rf` | Remove dir recursively, `-f` without confirmation |

| `mv` | | Move recursively |

| `find` | `-iname pattern` | Search dir/file case-insensitive |

| | `-mmin n` | Last modified n minutes ago |

| | `-mtime n` | Last modified n days ago |

| | `-regex pattern` | Path matches pattern |

| | `-size n[kMG]` | By file size (`-n` less than; `+n` greater than) |

| | `! searchparams` | Invert search |

\*\*File Manipulation\*\*

| Command | Param | Description |

| ------- | ------------------------------------------ | ------------------------------------------ |

| `cat` | `file` | Print content |

| `tac` | `file` | Print content inverted |

| `sort` | `file` | Print sorted |

| | `file -r -u` | Print sorted descending without dublicates |

| `wc` | `file` | Count Lines, Words, Chars (Bytes) |

| `head` | `-n10 file | tail -n5` | Print lines 5-10 |

| `tail` | `-f file` | Print new lines automatically |

| `cut` | `-f -4,7-10,12,15- file` | Print selected fields (tab delimited) |

| | `-c -4,7-10,12,15- file` | Print selected characters positions |

| | `-f 2,4 -d, --output-delimiter=$'\t' file` | Change delimiter (but use tab for output) |

| `uniq` | `file` | Hide consecutive identical lines |

| | `file -c` | Show consecutive identical line count |

| | `file -u` | Hide consecutive identical lines |

| `file` | `file` | Get file type |

\*\*Archiving\*\*

| Command | Param | Beschreibung |

| ---------------- | -------------------------------- | -------------------------------------------------------- |

| `tar` | `cfv archiv.tar file1 file2` | Archiv erstellen / Inhalt hinzufügen oder überschreiben |

| | `tfv archiv.tar` | Inhalt anzeigen |

| | `xf archiv.tar [-C ~/extracted]` | Archiv (nach ~/extracted) entpacken (und dekomprimieren) |

| | `cfvj archiv.tar.bz2 file` | bzip2 komprimiertes Archiv erstellen |

| | `cfvz archiv.tar.gz file` | gzip komprimiertes Archiv erstellen |

| | `cfa archiv.tar.[komp] file` | komprimiertes Archiv erstellen (auto Typ anhand Name) |

| `bzip2` / `gzip` | `file1 file2` | Dateien (einzeln) komprimieren |

| | `-d file1 file2` | Dateien dekomprimieren |

\*\*Other\*\*

| Command | Param | Beschreibung |

| ----------- | --------------- | ------------------------------------------ |

| `<command>` | `--help` | Help of current command (not standardized) |

| | `-h` | |

| | `-?` | |

| `man` | `<command>` | Manual page of command |

| | `-k keyword` | Search command by keyword (oder `apropos`) |

| `alias` | | Show aliases |

| | `name='befehl'` | Create alias |

## Globs (Wildcards)

The dot `.` in front of hidden items is ignored by glob patterns!

| Character | Description |

| --------- | ----------------------- |

| `?` | Any single character |

| `\*` | Any characters |

| `[ac-e]` | 1 character in enum |

| `[!ac-e]` | 1 character not in enum |

## Regex

Bash itself does not know regex. Use programs like `grep`, `sed`, `awk`.

\*\*Steuerzeichen\*\*

| Character | Description |

| -------------- | ----------------------- |

| `.` | Any single character |

| `[ac-e]` | 1 character in enum |

| `[^ac-e]` | 1 character not in enum |

| `^` | Start of string |

| `$` | End of string |

| `\d` | Digit |

| `\D` | Not a digit |

| `\s` | Whitespace |

| `\S` | Not a Whitespace |

| `\<` | Start of word |

| `\>` | End of word |

| `pattern?` | Quantifier 0 or 1 |

| `pattern\*` | Quantifier 0..n |

| `pattern+` | Quantifier 1..n |

| `pattern{x}` | Quantifier exactly x |

| `pattern{x,}` | Quantifier x..n |

| `pattern{x,y}` | Quantifier x..y |

| `pattern{,y}` | Quantifier 0..y |

\*\*Grep\*\*

| Command | Param | Description |

| ------- | ----------------- | -------------- |

| `grep` | `pattern file` | Extended Regex |

| | `-E pattern file` | Extended Regex |

| | `-v pattern file` | Invert match |

| | `-w pattern file` | Word match |

| | `-i pattern file` | Ignore case |

## Stream redirection

- `>` overwrite

- `>>` append

| Character | Description |

| --------------------- | ------------------------------- |

| `> file` or `1> file` | STDOUT to file |

| `< file` | Datei to STDIN |

| `2> file` | STDERR to file |

| `2>&1` | STDERR to same target as STDOUT |

| `> file 2>&1` | STDOUT and STDERR to file |

# Disk and File System Management

## General Disk Manipulation (non-LVM)

Creating physical partitions is \*\*not required\*\*! You can create PVs directly!

| Command | Description |

| ------------------------------------------- | ------------------------------------ |

| `fdisk -l` | List physical disks and partitions |

| `fdisk /dev/sdb`<br>`n` | Create new partition |

| `fdisk /dev/sdb`<br>`t`<br>`8e` | Change partition type to \*Linux LVM\* |

| `mkfs.xfs /dev/myVG/myVol` | Format LV with XFS |

| `mkfs.ext4 -f /dev/myVG/myVol` | Format LV with EXT4 (overwrite) |

| `blkid /dev/myVG/myVol` | Show UUID and formatting of volume |

| `mount` | Show what is mounted where |

| `mount -t ext4 /dev/myVG/myVol /mountpoint` | Mount LV to /mountpoint |

| `umount /dev/myVG/myVol` | Unmount LV from /mountpoint |

| `umount /mountpoint` | Unmount LV from /mountpoint |

| `mount -a` | Mount as configured in /etc/fstab |

| `df` | Show disk usage |

| `xfs\_growfs /dev/myVG/myVol` | Resize xfs filesystem |

| `resize2fs /dev/myVG/myVol` | Resize ext3/4 filesystem |

## LVM

| Item | Description |

| --------------- | --------------------------------- |

| Physical Volume | Representation of physical volume |

| Volume Group | Group of physical volumes |

| Logical Volume | Volumes placed on volume group |

\*\*Physical Volume Commands\*\*

| Command | Description |

| -------------------- | ------------------------------------------- |

| `pvs` | Show PVs |

| `pvdisplay` | Show detailed PV information |

| `pvcreate /dev/sdb1` | Initialize physical volume for use with LVM |

| `pvs -o+pv\_used` | Show usage of PVs |

| `pvmove /dev/sdb1` | Move contents of PV to other PVs of VG |

| `pvremove /dev/sdb1` | Remove PV |

\*\*Volume Group Commands\*\*

| Command | Description |

| ------------------------- | ----------------------------------- |

| `vgs` | Show VGs |

| `vgdisplay` | Show detailed VG information |

| `vgcreate myVG /dev/sda1` | Create new VG |

| `vgextend myVG /dev/sdb1` | Add PV to VG |

| `vgreduce myVG /dev/sdb1` | Remove PV from VG (`pvmove` first!) |

| `vgremove myVG` | Remove VG |

\*\*Logical Volume Commands\*\*

| Command | Description |

| -------------------------------------------- | --------------------------------------- |

| `lvs` | Show LVs |

| `lvdisplay` | Show detailed LV information |

| `lvcreate -n myVol -L 3G myVG` | Create LV |

| `lvextend -L +10G /dev/myVG/myVol -r` | Expand LV by 10GB and resize filesystem |

| `lvextend -L 10G /dev/myVG/myVol --resizefs` | Expand LV to 10GB and resize filesystem |

| `lvextend /dev/myVG/myVol /dev/sdb1` | Expand LV by PV's size |

| `lvextend -l +100%FREE /dev/myVG/myVol` | Expand LV to 100% available VG size |

| `lvreduce -L -10G /dev/myVG/myVol` | Reduce LV by 10GB |

| `lvremove /dev/myVG/myVol` | Remove logical volume |

# Automatic mounting (/etc/fstab)

Example:

| file system | mount point | type | option | dump | pass |

| ----------------------------------------- | ----------- | ---- | -------------------------------- | ---- | ---- |

| /dev/sdb1 | /myDir1 | ext4 | defaults | 0 | 0 |

| /dev/mapper/myVG-myVol | /myDir2 | xfs | defaults | 0 | 0 |

| UUID=a924e3de-0414-4228-86bf-b65a4c8ff8ab | /myDir3 | xfs | defaults | 0 | 0 |

| /dev/mapper/myVG-swap | swap | swap | defaults | 0 | 0 |

| 192.168.1.1:/shared | /targetdir | nfs | async | 0 | 0 |

| //server/share | /targetdir | cifs | credentials=/root/cifs/.username | 0 | 0 |

Column \*\*dump\*\* configures file system for backup with \*dump\* (program).

Column \*\*pass\*\* configures order of file system check during boot. `0` means no check.

# Text Readers & Editors

## Less

| Command | Description |

| ------------------- | ------------------------------- |

| `q` | Quit |

| `R` | Refresh content |

| `F` | Auto scroll |

| `g number` | Go to line |

| `m lowercaseLetter` | Mark line |

| `' lowercaseLetter` | Go to mark |

| `/pattern` | Search forward |

| `?pattern` | Search backward |

| `n` | Next search result |

| `N` | Last search result |

| `ESC u` | Remove highlighting from search |

## VI

\*\*Editing\*\*

To leave editing mode press `ESC`.

| Command | Description |

| --------- | --------------------- |

| `i` | insert before cursor |

| `a` | insert after cursor |

| `A` | insert at end of line |

| `o` | new line below |

| `O` | new line above |

| `u` | undo |

| `.` | repeat last command |

| `yy` | copy line |

| `5yy` | copy 5 lines |

| `p` | paste below |

| `P` | paste above |

| `x` | delete character |

| `5x` | delete 5 characters |

| `dd` | delete line |

| `5dd` | delete 5 lines |

| `:10,20d` | delete lines 10-20 |

| `d0` | delete to line begin |

| `d$` | delete to line end |

\*\*Navigation\*\*

Navigate as usual with `arrow keys`, `home`, `end`, `pg up`, `pg dn`.

| Command | Description |

| ------- | ---------------------- |

| `5G` | go to line 5 |

| `H` | go to top of screen |

| `M` | go to middle of screen |

| `L` | go to end of screen |

| `5w` | move over 7 words |

| `5b` | move back 5 words |

\*\*Other\*\*

| Command | Description |

| ----------- | ---------------------------- |

| `/foo` | search forward |

| `?foo` | search backwards |

| `n` | repeat search |

| `:w` | save |

| `:q` | close |

| `:wq` | save and close |

| `:q!` | close without saving |

| `:!command` | run bash command |

| `:r foo` | read file foo into this file |

# User and Group Management

\*\*UID\*\*

| UID | Type |

| ----- | -------------- |

| <1000 | system account |

| >1000 | user account |

\*\*User Database\*\*

User info without passwords is stored in `/etc/passwd`.

| username | PW | UID | GID | Kommentar | HOME | SHELL |

| -------- | --- | ---- | ---- | --------- | ----------- | --------- |

| hfict | x | 1000 | 1000 | | /home/hfict | /bin/bash |

\*\*Group Database\*\*

Group info with secondary group members are stored in `/etc/group`.

Primary group members are identified by GID in user database.

| groupname | PW | GID | Users |

| --------- | --- | --- | ----------- |

| wheel | x | 10 | hfict,user2 |

\*\*Password Database\*\*

Hashed user passwords are stored in `/etc/shadow`.

Password encryption is configured in `/etc/login.defs`.

| username | PW | Last PW change | Minimum | Maximum | Warn | Inactive | Expire |

| -------- | ------ | -------------- | ------- | ------- | ---- | -------- | ------ |

| hfict | [hash] | 17803 | 0 | 99999 | 7 | | |

PW:

- `[hash]` Verschlüsseltest Passwort

- `![hash]` Account Gesperrt

- `!!` oder `\*` Account gesperrt, kein Passwort gesetzt

\*\*Commands\*\*

| Command | Param | Beschreibung |

| ---------- | ---------------------------------------------- | -------------------------------------------------------- |

| `id` | `username` | ID und Gruppen eines Benutzers anzeigen |

| `who` | | Angemeldete Benutzer anzeigen |

| `last` | | Letzte Logins anzeigen |

| `lastb` | | Letzte fehlgeschlagene Logins anzeigen |

| `sudo` | `-u user command` | Befehl mit Rechten von user ausführen (Default ist root) |

| | `-i` oder `su -` | Shell mit root Rechten |

| `su` | | Shell als root (non-login shell) |

| | `-` | Shell als root (login shell) |

| | `- user` | Shell als user |

| `useradd` | ` -u 2101 -g primarygroup -c comment username` | Benutzer anlegen (ohne `-g` wird neue Gruppe erstellt) |

| `usermod` | `-G group1,group2` | Sekundärgruppen definieren (überschreiben) |

| | `-aG group,group2` | Sekundärgruppen hinzufügen |

| | `-l username` | Benutzernamen ändern |

| | `-L` | Lock Account |

| | `-U` | Unlock Account |

| | `-s shellpath` | Shell ändern |

| `userdel` | `-r username` | Benutzer inkl. Home und Mail-Spool löschen |

| `passwd` | `username` | Passwort ändern (interaktiv) |

| `groupadd` | `groupname` | Gruppe anlegen (Optional mit `-g` GID setzen) |

| `groupdel` | `groupname` | Gruppe löschen |

# File System Permissions

Permissions can be set on:

- User (owner)

- Group (owner)

- Others

Only root can change \*User\*. \*User\* can change \*Group\*.

Basic permissions (Add binary flags to combine):

| Char | Binary Flag | Permission |

| ---- | ----------- | ---------- |

| r | 4 | read |

| w | 2 | write |

| x | 1 | execute |

Advanced permissions (place in front of basic permissions: `chmod 1777 shared`).:

| Char | Binary Flag | Name | Description |

| ----- | ----------- | ---------- | -------------------------------------------------------------------------- |

| t / T | 1 | Sticky Bit | \*Others\* can't delete content (only applicable for directories) |

| s / S | 2 | SGID-Bit | File: run with permissions of \*Group\*<br>Dir: New elements inherit \*Group\* |

| s / S | 4 | SUID-Bit | File is run with permissions of \*User\* (only applicable for files) |

Advanced permissions replace the \*\*x\*\* when using `ls -l`. Lower case if \*\*x\*\* is set, upper case if \*\*x\*\* is not set.

\*Read\* permission on a directory only allows to see the directory itself but not it's contents. Use \*execute\* permission to show contents.

\*\*Commands\*\*

| Command | Param | Beschreibung |

| --------- | ------------------------ | ------------------------------------------------- |

| `chmod` | `-R [uog] dirname` | Set permissions recursively using binary flags |

| | `+[suog] filename` | Add permissions using binary flags |

| | `-[suog] filename` | Remove permissions using binary flags |

| | `u+x filename` | Add \*execute\* permission for \*User\* |

| | `g+wx filename` | Add \*write\* and \*execute\* permissions for \*Group\* |

| | `o-r filename` | Remove \*read\* permission for \*Others\* |

| `chown` | `-R user:group filename` | Change owner (\*User\* & \*Group\*) recursively |

| | `user filename` | Change owner (\*User\*) |

| | `:group filename` | Change owner (\*Group\*) |

| `chgroup` | `group filename` | Change owner (\*Group\*) |

# SSH

Configuration is done in `/etc/ssh/sshd\_config`.

Reload SSH service with `systemctl reload sshd` to apply changes!

DenyUsers, AllowUsers, DenyGroups, AllowGroups override each other and are applied in the order listed above.

| Config | Option | Description |

| ----------------- | ------------------ | --------------------------------------------- |

| `PermitRootLogin` | `no` | Deny root to login via SSH |

| | `yes` | Allow root to login via SSH |

| | `without-password` | Allow only with private/public key auth |

| `AllowUsers` | `user1 user2` | Allow only user1 and user2 |

| `DenyUsers` | `user1 user2` | Allow all users but user1 and user2 |

| `AllowGroups` | `group1 group2` | Allow only users from specified groups |

| `DenyGroups` | `group1 group2` | Allow all users but those in specified groups |

# IPv6 (disable)

Change the \*grub\* config `/etc/default/grub` to disalbe IPv6.

1. Add `ipv6.disable=1` to the `GRUB\_CMDLINE\_LINUX` property.

e.g. `GRUB\_CMDLINE\_LINUX="ipv6.disalbe=1 crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap biosdevname=0 net.ifnames=0 rhgb quiet"`

2. Recreate grub config with `grub2-mkconfig -o /boot/grub2/grub.cfg`

3. Reboot the system with `init 6` or `shutdown -r now`

# Cronjobs

\*\*Crontab\*\*

Cronjobs are configured in crontab files. Do not edit these files directly. Use `crontab -e` instead. This runs all required actions to activate a cronjob after saving the edited crontab. The locations are as follows:

- `/var/spool/cron/username` user specific

- `/etc/crontab` system wide crontab

The format of the files is (user specific crontabs \*\*do not\*\* have the column \*user-name\*):

```

Example of job definition:

.---------------- minute (0 - 59 | \*/5 [every 5 minutes])

| .------------- 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

```

| Command | Description |

| -------------------------------- | ----------------------------- |

| `rpm -q cronie` | Check if package is installed |

| `systemctl status crond.service` | Check if service is running |

| `crontab -l` | List current users crontab |

| `crontab -e` | Edit current users crontab |

| `crontab -e -u username` | Edit specific users crontab |

| `crontab -r` | Remove current users crontab |

\*\*Script folders\*\*

Scripts in one of the following directories will be executed at the intervall specified by the directory's name:

- `/etc/cron.hourly`

- `/etc/cron.daily`

- `/etc/cron.weekly`

- `/etc/cron.monthly`

\*\*Allow / Deny usage\*\*

Add user names one per line to the following files:

- `/etc/cron.allow` Whitelist

- `/etc/cron.deny` Blacklist

If none of the files exists, all users are allowed.

\*\*Logs and Results\*\*

Execution of cronjobs is logged in `/var/log/cron`.

Results are sent to the users mail `/var/spool/mail/username`.

# Package Management

## RPM

| Command | Description |

| ------------------------ | ----------------------------------------- |

| `rpm -i rpmfile|rpmurl` | Install package |

| `rpm -e packagename` | Uninstall package |

| `rpm -q packagename` | Check if package is installed |

| `rpm -ql packagename` | List files in a package |

| `rpm -qa` | List all installed packages |

| `rpm -qf /path/to/file` | Get package that installed the file |

| `rpm -qf $(which <exe>)` | Get package that installed the executable |

| `rpm -V packagename` | Validate installed package |

## YUM

YUM is configured in `/etc/yum.conf`

Repos are configured in `/etc/yum.repos.d/`

Log is in `/var/log/yum.log`

| Command | Description |

| ------------------------------------- | ----------------------------------------- |

| `yum install packagename [-y]` | Install package (`-y` no confirm message) |

| `yum remove packagename` | Uninstall package |

| `yum update` | Update all installed packages |

| `yum update packagename` | Update specific package |

| `yum update pattern\*` | Update packages using wildcard |

| `yum info packagename` | Get detailed info about package |

| `yum list packagename` | List installed and available packages |

| `yum search searchstring` | search for a package (name & summary) |

| `yum search all searchstring` | search for a package (all infos) |

| `yum deplist packagename` | List dependencies of a package |

| `yum reinstall packagename` | Reinstall (corrupted) package |

| `yumdownloader --resolve packagename` | Download rpm package with dependencies |

# Services (systemd)

| Command | Description |

| ------------------------------ | ------------------------- |

| `systemctl list-dependencies` | Show Services |

| `systemctl start servicename` | Start service |

| `systemctl enable servicename` | Start service with system |

| `systemctl status servicename` | Show service status |

# NFS

\*\*Installation (Client & Server)\*\*

```bash

yum install nfs-utils

```

\*\*Server\*\*

Configuration is done in `/etc/exports`.

```bash

systemctl start nfs-server.service # Start service

systemctl enable nfs-server.service # Start service at system start

vi /etc/exports # Change config

exportfs -fa # Reload config (server)

```

In `/etc/exports` create a new line for every directory you want to share in the following format:

`<dir> <destination1>(<options>) [<destination2>(<options>)]`

- `<dir>` Directory to export

- `<destinationx>` Client to allow access (by FQDN, hostname, ip, ip networks `192.168.0.0/28`)

- `<options>` Comma delimited configuration options

- `rw` Read/Write access

- `ro` \*\*default\*\* Read-only access

- `async` Aynchronous write (faster but can corrupt files)

- `root\_squash` \*\*default\*\* Anfragen von root werden anonymous zugewiesen

- `no\_root\_squash` Anfragen von root werden root zugewiesen

- `all\_squash` Alle Anfragen werden anonymous zugewiesen

Examples:

```

/ master(rw) trusty(rw,no\_root\_squash)

/projects proj\*.local.domain(rw)

/usr \*.local.domain(ro)

/home/joe pc001(rw,all\_squash,anonuid=150,anongid=100)

/pub \*(ro,insecure,all\_squash)

/foo 2001:db8:9:e54::/64(rw) 192.0.2.0/24(rw)

/build buildhost[0-9].local.domain(rw)

```

\*\*Client\*\*

```bash

showmount -e 192.168.1.1 # Show NFS shares

mkdir /targetdir # Create mountpoint

mount -t nfs 192.168.1.1:/shared /targetdir # Mount manually

umount /targetdir # Unmount manually

vi /etc/fstab # Mount on system start

mount -a # Test fstab

init 0 # Auto mount

```

Example of `/etc/fstab`:

```

192.168.1.1:/shared /targetdir nfs async 0 0

```

# Samba

\*\*Server\*\*

Configuration is done in `/etc/samba/smb.conf`.

Local accounts must be registered with Samba using `smbpasswd`.

Enter the same password as for the local account and add `unix password sync = yes` to `/etc/samba/smb.conf`. Then it should be synced automatically when you change it with `passwd`.

```bash

yum install samba samba-client samba-common # Samba installation

systemctl start smb # Start service

systemctl enable smb # Start service on system start

vi /etc/samba/smb.conf # Change config

testparm # Check if config is ok

systemctl restart smb # Reload config

smbpasswd -a username # Register user with samba

```

Example of `/etc/samba/smb.conf`:

```ini

[global]

workgroup = WORKGROUP # workgroup of smb server

netbios name = centos # name of smb server

security = user # security mode (user / ads / domain)

unix password sync = yes # sync unix password with smb password

invalid user = root bin daemon # deny access globally (config in share section overrides this)

[Transfer] # name of share

path = /transfer # path of directory to be shared

comment = File transfer # description

read only = no # access permissions

guest ok = no # allow guest access (no password)

browsable = yes # visible or hidden?

valid users = username @groupname # allow user/group to access the share

```

The special `[homes]` section automatically shares the users home directory under the following conditions:

- A users tries to access a share with the name of his user account

- There is no explicit section for this share in `smb.conf`

- The user account exists in `/etc/passwd`

If the path is not defined in the section, the home path of the user in `/etc/passwd` is used.

The `browsable` options has a different meaning in this section and specifies wheter the personal share should be listed. The usual `browsable` option is inherited from the `[global]` section.

You can use variables to change the path for example:

- `%U` current user name

- `%H` current users home directory

Example:

```ini

[homes]

browsable = no

writable = yes

path = %H/smb

```

\*\*Client\*\*

```bash

yum install samba-client cifs-utils # Install client and tools

smbclient -L //server # Show SMB shares

mkdir /targetdir # Create mountpoint

mount.cifs -o username=jenkins //server/james /mountpoint # Mount manually

umount.cifs /mountpoint # Unmount manually

vi /etc/fstab # Mount on system start

vi /root/cifs/.username # Create credential file

mount -a # Test fstab

init 0 # Auto mount

```

Example for `/etc/fstab`:

```

//server/share /targetdir cifs credentials=/root/cifs/.username 0 0

```

Example for `/root/cifs/.username`:

```ini

username=username

password=password

```

# FTP

Configuration is done in `/etc/vsftpd/vsftpd.conf`

Example:

```ini

anonymous\_enable=NO

local\_enable=YES

write\_enable=YES

local\_umask=022

dirmessage\_enable=YES

xferlog\_enable=YES

connect\_from\_port\_20=YES

xferlog\_std\_format=YES

listen=NO

listen\_ipv6=YES

pam\_service\_name=vsftpd

tcp\_wrappers=YES

use\_localtime=YES

dirlist\_enable=YES

pasv\_enable=YES

pasv\_min\_port=41361

pasv\_max\_port=65534

pasv\_address=192.168.1.10

```

```bash

yum install vsftpd ftp # Install server and client

systemctl start vsftpd # Start service

systemctl enable vsftpd # Start service on system start

vi /etc/vsftp/vsftpd.conf # Change config

systemctl restart vsftpd # Load modified config

```

# MySQL Server

```bash

yum install mariadb-server # Install server

systemctl start mariadb # Start service

systemctl enable mariadb # Start service on system start

mysql\_secure\_installation # Run security wizard

mysql -u root -p # Connect to MySQL

```

# Other

Stuff probably relevant for the exam.

Exam result file: `/sbin/result`

```bash

useradd username [-G wheel] # add user and allow sudo

usermod -aG wheel username # allow sudo for existing user

sudo -i # enter interactive root session

yum install man-pages # install man pages

timedatectl set-timezone Europe/Zurich # change time zone

timedatectl set-local-rtc 0 # set RTC to use UTC

yum check-update # check for updates

yum update # install all available updates

localectl # show locale configuration

localectl list-keymaps | grep ch # show available keymaps

localectl set-keymap ch # set keymap to ch (doesn't affect ssh session)

localectl set-x11-keymap ch # set keymap to ch (doesn't affect ssh session)

```