

# Training SuSE Linux Enterprise 15 (SLES 15)

## Agenda

### 1. Grundlagen

- [Änderungen von SLES 12 zu 15](#)

### 2. Installation / Administration

- [Working with modules/extensions](#)
- [Working with SUSEConnect](#)
- [Systemctl](#)
- [firewalld](#)
- [supportconfig](#)

### 3. Upgrade

- [Upgrade SLES 15](#)

### 4. Installation Datenbanken

- [postgresql](#)
- [mariadb](#)

### 5. Management with zypper and rpm

- [Zypper commands](#)
- [Zypper cheatsheet](#)
- [Find out installed packages with version](#)

### 6. Verzeichnisse und Dateitypen

- [Filesysteme](#)
- [Verzeichnisaufbau](#)
- [Dateitypen](#)

### 7. sudo

- [sudo unter SLES sicher einrichten](#)

### 8. vim

- [vim](#)

### 9. timers

- [create a simple time with systemctl](#)

### 10. Unix-Tools

- [Beispiel - nur befehl von history auslesen mit unixtools](#)

### 11. nfs-server und lvm

- [LVM logical volume erweitern](#)
- [NFS-Server unter SLES 15 einrichten](#)

### 12. Autostart

- [autostart](#)

### 13. nfs / Zeitserver

- [Chrony](#)

### 14. Documentation

- [What is journal ordered in ext3/ext4](#)
- [Maximum blocksize of ext4 and tests](#)
- [Journal - Größe anpassen](#)
- [Server in AD einbinden](#)

## Backlog

Working with .ssh/config to predefine certain settings when connection

```
localhost:~/.ssh # cat config
Host peterson2
    HostName 192.168.1.101
    User training

localhost:~/.ssh # ls -la
total 8
drwx----- 1 root root  34 Aug  5 10:14 .
drwx----- 1 root root 182 Aug  5 10:14 ..
-rw----- 1 root root  58 Aug  5 10:14 config
-rw-r--r-- 1 root root 181 Aug  5 10:14 known_hosts
```

## Networking search order for hosts

```
localhost:/etc # grep -r "files dns" .
./nsswitch.conf:hosts:      files dns
./nsswitch.conf:networks:   files dns
localhost:/etc #
```

## Search for pattern recursively in folder

```
localhost:/etc # grep -r 192.168.1.100 .
```

## Ping fehlt ?

zypper install iputils

```
### bei installierten Paketen herausfinden, welches paket befehl bereitstellt
zypper se --provides ping
```

## Samba - Server

Gute Einführung [https://www.thomas-krenn.com/de/wiki/Samba-Server\\_Grundlagen](https://www.thomas-krenn.com/de/wiki/Samba-Server_Grundlagen)

Spezialwissen (WINS-Server, Local Master Browser): <http://www.linux-praxis.de/linux3/samba6.html>

SLES 15: <https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-samba.html>

```
Manage samba-password db:
# list all files
pdbedit -L

## Important
smbpasswd -a training # add linux user to samba password db

# test configuration
testparm
```

## Testing connection with smbclient

```
smbclient -U training -L //10.10.2.106
```

## The things with the acls

Supports windows acls [https://wiki.samba.org/index.php/Setting\\_up\\_a\\_Share\\_Using\\_Windows\\_ACLs](https://wiki.samba.org/index.php/Setting_up_a_Share_Using_Windows_ACLs)

But a bit tricky

## Mount other share on other server (samba)

```
mount.cifs //10.10.2.106/dokumente /mnt/dokumente -o username=training,password=training
# or mount -t cifs
```

## basic networking

<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-network.html#sec-network-addresses>

## sed

Good introduction

<https://www.digitalocean.com/community/tutorials/the-basics-of-using-the-sed-stream-editor-to-manipulate-text-in-linux>

```
# Recognize Email
sed 's/.*/ &/;s/.* \([^ @]*@[^ @]*.com\).*/CENSORED/' emailstobecensored
# Ref: https://www.unix.com/shell-programming-and-scripting/181361-sed-regex-extract-email-address.html
```

## Example for awk script (splitting fields with "-" as separator between fields)

```
### file awktest.sh
### chmod u+x awktest.sh

#!/usr/bin/awk -f
/tcp/ {
ORS=" ";
for (i=1; i<=5; i++) {
    if ( i == 1 ) print $i;
    else {
        print " - ", $i
    }
}
print "\n"
}

### Executing of script
./awktest.sh /etc/services
```

## Great resources for setting specific VAR for NEWLINE and FIELDSOUTPUT-Separator

[https://www.funtoo.org/Awk\\_by\\_Example\\_Part\\_2](https://www.funtoo.org/Awk_by_Example_Part_2)

## awk

[https://www.tutorialspoint.com/awk/awk\\_basic\\_examples.htm](https://www.tutorialspoint.com/awk/awk_basic_examples.htm)

## Change console layout

```
# switch to german layout
sudo loadkeys de
```

## grep - show header

```
ps xo pid,comm | head -n 1; ps xo pid,comm | grep 11
```

## systemctl cheatsheet

<https://www.linuxtrainingacademy.com/systemd-cheat-sheet/>

## zypper

<https://en.opensuse.org/images/1/17/Zypper-cheat-sheet-1.pdf>

```
zypper patch-check
# show only the first column -> result -> amount of patches that can be done
zypper patch-check | tail -n 1 | cut -d ' ' -f1 # show field 1 -> -f1 && use space as delimiter -d ' '
```

## Kernel Parameters at boot time

<https://documentation.suse.com/sles/15-SP1/html/SLES-all/cha-boot-parameters.html>

## Kernel - crashkernel

<https://www.suse.com/de-de/support/kb/doc/?id=000016171>

## Package Management

Cheatsheet: <https://danilodellaquila.com/en/blog/linux-package-management-cheatsheet>

## Commands

```
# Kernelversion die geladen ist, anzeigen
less filename # pager
cat /boot/meinfile | less # verlassen mit q + return
uname -a
# which - zeigt alle Vorkommen eines Befehls im Pfad -> echo $PATH
which depmod
# show all line with c at the beginning of line
ls -la /dev | grep ^c
# show all lines with k at the end of the line
ls -la /dev | grep k$
```

## Grundlagen

### Änderungen von SLES 12 zu 15

#### Modul - Struktur

- Unified Installer
- SLES 15 ist dann in Module aufgespalten
- Jedes Modul hat einen klar definierten Bereich mit verschiedenen Lebenszyklen und Update Zeitplänen

#### zypper search-packages

- Durchsucht auch repos die noch nicht eingebunden sind
- z.B. zypper search-packages iperf
  - findet paket in package hub

#### Kernel/Bash Versionen

- Kernel 4.4 -> 4.12
- Bash 4.3.42 -> 4.4.23

#### ntp -> chrony

- NTP -> chrony

#### Firewall

- SuSEFirewall2 -> firewalld

#### Python

- python2 -> python3

#### SMT (Subscription Management Tool) -> RMT (Registry Mirror Tool und Registration Proxy für SCC (SuSE Connect Center))

- Spiegeln von SUSE Repositories und Kunden Repositories. Proxy zu anderen RMT Servern

#### LDAP

- OpenLDAP -> 389ds (LDAPv3 compliant server)

#### Automation

- puppet->salt
- salt wird für das Konfigurationsmanagement benutzt. SUSE Manager

#### X-Server

- X11 -> Wayland
- Gnome is now using Wayland by default (X11 before)

#### unrar -> unar

- Unrar ersetzt aus lizenzrechtlichen Gründen unrar.

#### Changes in /etc/systemd/system.conf

```
##CtrlAltDelBurstAction=reboot-force
##DefaultStartLimitIntervalSec=10s
##DefaultIOAccounting=no
##DefaultTasksMax=15%
```

#### /etc/init.d

- is now empty
- SystemV Skripte können nicht aktiviert werden
- Falls nötig paket insserv-compat nachinstallieren

#### Netzwerk

#### SuSE-release

- Datei SuSE-release ist entfernt worden

#### Floppy-Disk

- Floppy Disk Support ist entfernt worden

## Filesysteme

- ReiserFS - Support ist entfernt worden
- Der Installer blockiert eine Migration, wenn
  - ReiserFS entdeckt wird

## Finger

- Den Befehl gibt es nicht mehr

## cronjobs

```
Einige cronjobs unter /etc/cron
(hourly,daily,weekly,monthly) sind jetzt als Timer Units implementiert worden.
snapper-timeline.timer
backup-sysconfig.timer
backup-rpmbd.timer
check-battery.timer
snapper-cleanup.timer
systemd-tmpfiles-clean.timer
btrfs-balance.timer
fstrim.timer
##btrfs
```

## Neues btrfs Layout für /var

```
SLES 12:
verschiedene Subvolumes unter /var (mit COW)

SLES 15:
nur noch subvolume /var -> ohne COW eingerichtet
```

## Fazit

- Veraltete Befehle sind entfernt worden

## Installation / Administration

### Working with modules/extensions

#### General

- modules extend functionality
- extensions also, but require an additional license (registration key)

#### Modules

Add functionality to your system (starting from SLES 15)

All modules are available for you installaton

#### list modules/extensions with SUSEConnect

```
sudo SUSEConnect --list-extensions
```

#### Extensions

They hold specific software-extension like high availability. You have to buy a license to use them.

#### Activate module "Desktop" - so that it is possible to install Gnome

```
SUSEConnect -p sle-module-desktop-applications/15.2/x86_64
```

## Documentation

- Basics about extensions: <https://documentation.suse.com/sles/15-SP1/html/SLES-all/art-modules.html>

## Working with SUSEConnect

### Register SLES with email and license key

```
## Command -r registration code (from suse.com)
sudo SUSEConnect -r xxxxxxxxxxxx -e name@domain
```

## Unregister SLES with email and license key

```
## unregister/deregister system / otherwise, you cannot reuse the key
sudo SUSEConnect -d -e name@domain
```

## Systemctl

### systemctl Beispiele

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

## Wie heisst der Dienst / welche Dienste gibt es ? (nur wenn der service aktiviert ist).
systemctl list-units -t service
## für apache
systemctl list-units -t service | grep apache
## die Abkürzung
systemctl -t service | grep apache
## alle Dienste egal ob aktiviert oder nicht
systemctl list-unit-files -t service | grep ssh

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

## Dienst deaktivieren (nach Booten nicht starten)
systemctl disable apache2
systemctl is-enabled
disabled
echo $?
1 # 1 wenn nicht aktiviert

## Rebooten des Servers
## verweist auf systemctl
reboot
systemctl reboot
shutdown -r now

## Halt (ohne Strom ausschalten)
halt
systemctl halt
shutdown -h now

## Poweroff
poweroff
systemctl poweroff
```

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

```
## z.B. für Apache2
systemctl cat apache2.service
```

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

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

```
lrwxrwxrwx 1 root root 13 Jan  6 20:47 runlevel1.target -> rescue.target
lrwxrwxrwx 1 root root 17 Jan  6 20:47 runlevel2.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jan  6 20:47 runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jan  6 20:47 runlevel4.target -> multi-user.target
lrwxrwxrwx 1 root root 16 Jan  6 20:47 runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13 Jan  6 20:47 runlevel6.target -> reboot.target
```

## Welche Dienste sind aktiviert/deaktiviert

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

## Dienste bearbeiten

```
systemctl edit sshd.service
## Dann eintragen
[Unit]
Description=Jochen's ssh-server
## Dann speichern und schliessen (Editor)
```

```
## nur falls es nicht funktioniert !
## systemctl daemon-reload
systemctl status
```

## Targets (wechseln und default)

```
## Default runlevel/target auslesen
systemctl get-default
## in target wechseln
systemctl isolate multi-user
## Default target setzen (nach start/reboot)
systemctl set-default multi-user
```

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

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

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

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

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

## Systemctl - Cheatsheet - Ref

- <https://www.linuxtrainingacademy.com/systemd-cheat-sheet/>

## Beispiel / Dienste starten / restarten

```
[Unit]
StartLimitInterval=400
StartLimitBurst=3

[Service]
```



```
Restart=on-failure
RestartSec=5
```

## Systemd - dokumentation

- <https://www.freedesktop.org/software/systemd/man/systemd.unit.html#>

## firewalld

### Install firewalld

- firewalld is installed on SLES by default
- It uses either nft or iptables as backend
  - There is a change here between service packs

### Is firewalld running ?

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

### Command to control firewalld

- firewall-cmd

## Zones documentation

man firewalld.zones

### Zones available

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

### Active Zones

```
firewall-cmd --get-active-zones
## in our case empty
```

### Add Interface to Zone = Active Zone

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

### Show information about all zones that are used

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

### Default Zone

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

### Show services / Info

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

### Adding/Removing a service

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

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

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

## Best way to add a new rule

```
## Step1: do it persistent -> written to disk
firewall-cmd --add-port=82/tcp --permanent

## Step 2: + reload firewall
firewall-cmd --reload
```

## Enable / Disabled icmp

```
firewall-cmd --get-icmptypes
## none present yet
firewall-cmd --zone=public --add-icmp-block-inversion --permanent
firewall-cmd --reload
```

## Working with rich rules

```
## Documentation
## man firewallld.richlanguage

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

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

##

## list only the rich rules
firewall-cmd --zone=public --list-rich-rules

## persist all runtime rules
firewall-cmd --runtime-to-permanent
```

## References

- <https://www.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/>

## supportconfig

## Installation and usage

```
zypper install -y supportutils
zypper install -y yast2-support
yast2 support
```

### Ref:

- <https://documentation.suse.com/de-de/sles/15-SP1/html/SLES-all/cha-adm-support.html>

## Upgrade

### Upgrade SLES 15

#### Schritt 1:

- <https://documentation.suse.com/sles/15-SP4/single-html/SLES-upgrade/index.html#cha-upgrade-offline>

#### Preparation

##### Substep 1:

- <https://www.suse.com/releasenotes/index.html>
- Matrix mit Links zu den Release Notes
- Release Notes studieren (Wichtige Pakete und Paketquellen durchführen)
  - [https://documentation.suse.com/package-lists/sle/15-SP4/package-changes\\_SLE-15-SP3-GA\\_SLE-15-SP4-GA.txt](https://documentation.suse.com/package-lists/sle/15-SP4/package-changes_SLE-15-SP3-GA_SLE-15-SP4-GA.txt)
  - [https://documentation.suse.com/package-lists/sle/15-SP4/module-changes\\_SLE-15-SP3-GA\\_SLE-15-SP4-GA.txt](https://documentation.suse.com/package-lists/sle/15-SP4/module-changes_SLE-15-SP3-GA_SLE-15-SP4-GA.txt)

##### Substep 2:

- Sichern von /etc,/var, /home (Empfehlung von SUSE)
- Sicherung installierte Pakete und Paketquellen durchführen

```
zypper lr -e repositories.bak
rpm -qa --queryformat '%{NAME}\n' > installed-software.bak
```

```
## wiederherstellen
zypper ar repositories.bak.repo
zypper install $(cat installed-software.bak)
```

##### Substep 3:

- Patch current version
- zypper patch

#### Optional: media\_upgrade

```
To force the installer to only install packages from the DVD and not from network sources, add the boot option
media_upgrade=1.
```

## Installation Datenbanken

### postgresql

#### Walkthrough

```
zypper search-packages postgresql
## also installs the client packages as dependencies
zypper install -y postgresql14-server
systemctl start postgresql

## empty
ls -la /var/lib/pgsql

systemctl start postgresql
## Do not forget to enable it
systemctl enable postgresql
```

```
## now we have data files
ls -la /var/lib/pgsql/data
```

## Filestructure

- <https://www.postgresql.org/docs/current/storage-file-layout.html>

## Administrationsuser konfigurieren

```
sudo passwd postgres
su - postgres
psql
## so kommen wir wieder raus
\q
```

## Walkthrough Datenbank

```
su - postgres
createuser -P -d testu
createdb -O testu testdb
psql testdb
```

```
## in psql
## Jetzt kann man eine Tabelle in der Datenbank anlegen:
```

```
CREATE TABLE books ( id int, name varchar(80), publisher varchar(80), date_published date );
```

```
## Die Liste der Relationen kann man sich anzeigen lassen:
testdb=# \dt
```

```
Liste der Relationen
Schema | Name  |  Typ  | Eigentümer
-----+-----+-----+-----
public | books | Tabelle | postgres
```

Mit INSERT fügt man Daten in die Tabelle ein:

```
INSERT INTO books VALUES ('1', 'Reference openSUSE Leap 15.1', 'SUSE LLC', '2019-05-25');
```

Die Tabellenstruktur liest man mit:

```
\d books
```

```
Tabelle »public.books«
  Spalte  |          Typ          | Sortierfolge | NULL erlaubt? | Vorgabewert
-----+-----+-----+-----+-----
id        | integer               |              |               |
name      | character varying(80) |              |               |
publisher | character varying(80) |              |               |
date_published | date                 |              |               |
```

## Ob der Eintrag der Daten in die Tabelle geklappt hat, zeigt das SELECT-Statement:

```
SELECT * FROM books;
```

```
id |          name          | publisher | date_published
---+-----+-----+-----
 1 | Reference openSUSE Leap 15.1 | SUSE LLC | 2019-05-25
```

```
## tabelle löschen
drop table books;
\q
```

```
## in bash
dropdb testdb;
```

## Authentifizierung von remote

```
vi /var/lib/pgsql/data/postgresql.conf
## listen_addresses
listen_addresses='*'
```

```
## vi /var/lib/pgsql/data/pg_hba.conf
cat pg_hba.conf | grep -A 1 -i " ipv4"
## IPv4 local connections:
host    all             all             10.163.24.0/24          trust
```

```
systemctl restart postgresql
firewall-cmd --add-service=postgresql
firewall-cmd --add-service=postgresql --permanent
```

```
## vom anderen Rechner verbinden
## client tools installieren
zypper install -y postgresql14
## Achtung erfolgt ohne passwords
psql -h <ip-des-postgres-servers> -U postgres
```

## Ref:

- <https://de.opensuse.org/PostgreSQL>

## mariadb

### Walktrough

```
zypper install -y mariadb
systemctl start mariadb
systemctl enable mariadb
```

## Daten, wo ?

```
## Erst nach dem ersten Start unter
## /var/lib/mysql
##
```

## Grundabsichern

```
mysql_secure_installation
```

## Einige Kommandos

```
mysql>
show databases;
use mysql;
select * from user;
select * from user \G
pager less
select * from user \G
nopager
```

## MariaDB nach aussen öffnen

```
## vi /etc/my.cnf.d/z_settings.cnf
[mysqld]
listen-address=0.0.0.0
```

```
systemctl restart mariadb
firewall-cmd --add-service mysql
firewall-cmd --add-service mysql --permanent

echo "CREATE USER ext1@'%' identified by 'password'" | mysql
echo "GRANT ALL ON *.* TO ext1@'%" | mysql
```

## MariaDB Client

```
mysql -u ext1 -p -h <ip-des-mariadb-servers>
mysql -u ext1 -p -h 10.163.24.108
```

## Management with zypper and rpm

### Zypper commands

#### Pakete für ping und nslookup ohne Server-Module zu installieren

```
zypper install -y bind-utils
zypper install -y iputils

zypper search test_
## Pakete installieren von modulen die nicht noch nicht installiert sind
## Cool: findet auch Pakete, wenn sie im community repo sind
zypper search-packages iperf
zypper source-install apache2-mod_nss
```

```
## Programme finden, die nicht installiert sind
zypper install command-not-found
cnf iperf
```

```
## installierte Pakete mit Version anzeigen
zypper search --only-installed -s
```

```
## Nach Paketen mit wildcard suchen
zypper search yast*
zypper search yast-
```

### Show available patterns

```
## patterns = groups of software packages, that can be easily installed
zypper search patterns-
```

### Zypper cheatsheet

- <https://en.opensuse.org/images/3/30/Zypper-cheat-sheet-2.pdf>

### Find out installed packages with version

```
rpm -qa
zypper search --installed-only -s
```

## Verzeichnisse und Dateitypen

### Filesysteme

#### Btrfs:

```
Bigger files are possible
snapshots
subvolumes
Raid
more methods to achieve Data Integrity
Special Tools needed (btrfs Tools)
```

```
df does not reflect size properly
Mounting Snapshots
Safer because of COW (copy on Write)
-> Disabled on SLES 15
```

## XFS

```
Good for bigger files (Performance)
Bad Performance on small files
So it is mainly used for databases
Schneller Filesystem check
```

## Ext4:

```
Verbesserung von ext3
Journalbasiert, abwärtskompatibel
Rocksolid
Quicker but not so many Features for Data Integrity
```

## Linux Dateisysteme - Welche gibt es ?

- ext2/ext3
- ext4
- btrfs
- zfs
- xfs
- reiserfs

## Vergleich der wichtigsten Filesysteme

	Ext3	Ext4	XFS	Btrfs
Production-Ready	✓	✓	✓	teilweise
Max. Dateisystem-Größe	16 TiB	1 EiB	16 EiB	16 EiB
Max. Datei-Größe	2 TiB	1 EiB	8 EiB 8 EiB	
Online vergrößern	✓	✓	✓	✓
Online verkleinern	-	-	-	✓
Offline vergrößern	✓	✓	-	-
Offline verkleinern	✓	✓	-	-
Discard (ATA Trim)	✓	✓	✓	✓
Metadaten CRC	✓	✓	✓	✓
Daten CRC	-	-	-	✓
Snapshots/Clones/Internal RAID/Compression	-	-	-	✓

## Das Journal-Filesystem

- Alle Änderungen werden in ein Journal geschrieben, bevor sie auf Festplatte geschrieben werden
- Crashed das System z.B. durch Stromausfall, kann mit Hilfe des Journals wieder ein konsistenter Stand hergestellt werden.
- --
- Filesysteme mit Journal
- --
- ext3/ext4
- XFS
- reiserfs

## Der Klassiker ext2/ext3 bzw. ext4

- ext2 (kein Journal)
- Erstes Filesystem von Linux überhaupt
- ext3 identisch mit ext2, jedoch mit Journaling
- ext4 Weiterentwicklung von ext3

## Vergrößern / Verkleinern von Dateisystemen

### Dateisysteme und ihre Möglichkeiten

Dateisystem	Online	Online	Offline	Offline
	<i>Vergrößern</i>	<i>Verkleinern</i>	<i>Vergrößern</i>	<i>Verkleinern</i>
ext2/ext3/ext4	ja	nein	ja	ja
ReiserFS	nein	nein	ja	ja
JFS	ja	nein	nein	nein
XFS	ja	nein	nein	nein
NTFS	nein	nein	ja	ja
FAT	nein	nein	ja	ja

- Vergrößern ext3/ext4 seit Kernel 2.6.

### Vorgehen beim Verkleinern / Vergrößern

- Vor dem Verkleinern einer Partition bzw. eines Logical Volume immer zuerst das Dateisystem verkleinern
- Vor dem Vergrößern des Dateisystems immer zuerst das Logical Volume bzw. die Partition vergrößern

### Verkleinern/Vergrößern - gute Referenz

- [https://wiki.ubuntuusers.de/Dateisystemgr%C3%B6%C3%9Fe\\_%C3%A4ndern/](https://wiki.ubuntuusers.de/Dateisystemgr%C3%B6%C3%9Fe_%C3%A4ndern/)

## Verzeichnisaufbau

### /etc

- Verzeichnis für Konfigurationsdateien

### /dev

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

### /mnt

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

### /media

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

### /opt

- Große Softwarepakete (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
```

## **sudo**

### **sudo unter SLES sicher einrichten**

#### **Walkthrough**

```
## Auskommentieren, damit password des Nutzers abgefragt und nicht root
## Defaults targetpw # ask for the password of the target user i.e. root

## Attention: Comment out, because otherwise everybody is allowed to do anything
## ALL ALL=(ALL) ALL # WARNING! Only use this together with 'Defaults targetpw'!

## Jeder, der in der Gruppe wheel ist, hat alle sudo root-Rechte
%wheel ALL=(ALL) ALL
```

### **Spezifische Rechte für Benutzer**

```
groupadd externer
useradd -m -s /bin/bash externer
passwd externer
```

```
##vi /etc/sudoers.d/externer
%externer ALL=(ALL) /usr/bin/systemctl restart *,/usr/bin/chown
```

```
chmod 440 /etc/sudoers.d/externer
```

```
## Testen
su - externer
sudo systemctl restart sshd
```

## **vim**

### **vim**

### **vim installieren (falls nicht installiert)**

```
zypper install vim
```

## Zeilennummern aktivieren für meinen User

```
cd
vi .vimrc
## eitrage
set number
```

## Wichtigste Aktionen

```
1. # Öffnen eine neuer Datei mit vi
vi dateiname

2. # Schreiben in der Datei
i # <- i-Taste drücken

3. # Es erscheint unten in der Zeile
# -- INSERT --

4. # Nun können Sie etwas hineinschreiben

5a. Beenden ohne Speichern (wenn geänderter Inhalt vorhanden ist)
ESC + :q! # ESC Taste drücken, dann : und q! und enter

5b. Oder: Speichern und schliessen
ESC + :x # ESC Taste drücken, dann : und w und enter
```

## Virtual Mode

```
v Zeichenweise markieren einschalten
V Zeilenweise markieren einschalten (SHIFT + v)
STRG + v Blockweise markieren

## mit Cursortasten auswählen / markieren
## Dann:
x # Löschen des markierten Bereichs
```

## Zeilen löschen im Normalmodus (Interaktiver 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](http://www.atmos.albany.edu/daes/atmclasses/atm350/vi_cheat_sheet.pdf)

## timers

### create a simple time with systemctl

### Schritt 1: script erstellen und testen

```
vi /usr/local/bin/scriptv2.sh
```

```
#!/bin/bash
LOGTO=/var/log/scriptv2.log
echo "script script-ng schreibt was ins log...."
env
date >> $LOGTO
env >> $LOGTO
```

```
chmod u+x /usr/local/bin/scriptv2.sh
scriptv2.sh
```

## Schritt 2: Service erstellen und testen

```
systemctl edit --force --full scriptv2.service
```

```
[Unit]
Description=simple script for testing timer
[Service]
Type=oneshot
ExecStart=/usr/local/bin/scriptv2.sh
StandardOutput=journal
```

```
systemctl status scriptv2
systemctl start scriptv2
systemctl status scriptv2
```

## Schritt 3: Timer erstellen und testen

```
systemctl edit --force --full scriptv2.timer
```

```
[Unit]
Description=Timer for scriptv2
[Timer]
OnCalendar=*:0/5

[Install]
WantedBy=timer.target
```

```
systemctl enable scriptv2.timer

systemctl status scriptv2.timer
systemctl start scriptv2.timer
systemctl status scriptv2.timer

systemctl list-timers
```

## Reference:

- `man systemd.timer`

## Unix-Tools

### Beispiel - nur befehl von history auslesen mit unixtools

```
## -s ' ' ersetzt alle doppelten leerzeichen durch eines
## cut -d -> setzt das Trennzeichen fest
## -f5- zeigt die Spalte 5 und alle weiteren Spalten an
history | tr -s ' ' | cut -d ' ' -f5-
```

## nfs-server und lvm

### LVM logical volume erweitern

## Beispiel filesystem vergrößern

```
lvresize --resizefs -L +128MiB /dev/vg_data/lv_data
```

## Prerequisites

- Volume Group needs to be big enough

```
parted /dev/sdb mkpart lvtest3 5200MB 8200MB
pvcreate /dev/sdb4
vgextend vg_data /dev/sdb4
```

## Snapshot erstellen

```
## Erstellt ein Snapshot in der gleichen Volume Group
## muss die gleiche volume group sein
## -s -> snapshot erstellen
## "snapshot" is not allowed as name, as it is reserved
lvdisplay
lvcreate -L 2G -s -n mysnapshot /dev/vg_data/lv_data
```

## Snapshot restore (zurückspielen)

```
## show the data like ls in logical volume
lvs
```

```
## Explaining lvs
s : for snapshot, "o" meaning origin for the original logical volume copied to the snapshot;
w : for writeable meaning that your snapshot has read and write permissions on it;
i : for "inherited";
a : for "allocated", meaning that actual space is dedicated to this logical volume;
o : (in the sixth field) meaning "open" stating that the logical volume is mounted;
s : snapshot target type for both logical volumes
```

```
## Now add something in folder to test it
cd /mnt/lvtest
## Putting file test into folder
touch test
cd ../../
```

```
## To restore it, you need to mount the backup
lvconvert --merge /dev/vg_data/lv_data /dev/vg_data/snapshot
## you need to unmount firstly
umount /dev/vg_data/lv_data
lvchange -a n /dev/vg_data/lv_data
lvchange -a y /dev/vg_data/lv_data
mount /dev/vg_data/lv_data /mnt/lvtest
cd /mnt/lvtest
ls -la
### test should not be there anymore
```

## Reference

- <https://devconnected.com/lvm-snapshots-backup-and-restore-on-linux/>
- <https://linuxconfig.org/create-and-restore-manual-logical-volume-snapshots>

## NFS-Server unter SLES 15 einrichten

### Server

#### Step 1: Server installieren

```
zypper install nfsserver
```

### Step 3: Firewall einrichten

```
firewall-cmd --permanent --add-service=nfs
firewall-cmd --permanent --add-service=mountd
firewall-cmd --permanent --add-service=rpc-bind
firewall-cmd --reload
```

## Autoyast

### autoyast

### Walkthrough

```
zypper install autoyast2 autoyast2-installation
yast clone_system
ls -la /root/autoinst.xml
```

### Installation

```
## put file on tftp server e.g.
## in kernel params
AutoYaST=URL
```

### Ref:

- <https://doc.opensuse.org/documentation/leap/autoyast/html/book-autoyast/Invoking.html>

## nfs / Zeitserver

### Chrony

### Server

```
chronyc
(daemon not running from the beginning)
chronyd

systemctl status chronyd

yast ntp-client

systemctl start chronyd
systemctl status chronyd
systemctl enable chronyd

chronyc activity
chronyc
> activity
```

### Konfigurieren

```
yast ntp-client
```

### Start ntp-server

- manual
- ohne daemon synchronisieren
- now and at start

### ohne Daemon synchronisieren

- synchronisierung erfolgt ohne daemon im interval (default 5 min)
- Intervall läßt sich einstellen.

### Konfigurationsquelle (ntp-server über)

```
statisch.  
dynamisch: erfolgt über dhcp
```

## Überprüfen

```
systemctl status chronyd  
date  
timedatectl
```

## Documentation

### What is journal ordered in ext3/ext4

- [https://people.redhat.com/rpeterso/KB/ext3\\_tune.html](https://people.redhat.com/rpeterso/KB/ext3_tune.html)

### Maximum blocksize of ext4 and tests

```
## To change block size of ext4, partition needs to be unmounted  
umount /mnt/platte  
blockdev --getbsz /dev/sdb1  
  
## not possible  
blockdev --setbsz 8192 /dev/sdb1  
  
## you can try  
mkfs.ext4 -b 8192 /dev/sdb1  
### answer  
### 8192 to big for ext4 (4096) max  
### proceed anyway
```

### Ref:

- [https://ext4.wiki.kernel.org/index.php/Ext4\\_Disk\\_Layout#:~:text=With%20the%20default%20block%20size,disk%20in%20little%20endian%20order.](https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout#:~:text=With%20the%20default%20block%20size,disk%20in%20little%20endian%20order.)
- Blocksizes up to 64KiB are possible (see Dokument)

### Journal - Groesse anpassen

```
## /etc/systemd/journal.conf  
  
[Journal]  
SystemMaxUse=50M  
SystemMaxFileSize=10M  
  
### Server in AD einbinden  
  
* https://www.suse.com/support/kb/doc/?id=000018831
```