

Linux

- [Boot](#)
- [special distro behaviour](#)
- [Shell](#)
- [Shell scripting](#)
- [Network](#)
- [Permissions, Groups](#)
- [DE Desktop environments](#)
- [Different issues and settings](#)
- [CLI tools](#)
 - [find, search, grep](#)
 - [Cronjob, tar, autostart, sudo, ln](#)
 - [CLI Editor](#)
 - [user management](#)
 - [disk/file management](#)
 - [tmux](#)
 - [basic](#)
- [special tools](#)
- [New Server Setup](#)

Boot

add boot entry

- <https://linuxconfig.org/booting-a-ms-windows-os-using-grub>
- Boot to your Linux
- Open a Terminal window
- Type: `sudo os-prober`
- If the results identify a Windows 10 install, type `sudo update-grub` which will update the GRUB entries and add it
- Reboot your computer and test to see if Windows 10 is a new option to select.

remove boot entries

If you can boot into Linux in UEFI mode, use `efibootmgr`:

- `efibootmgr` or `efibootmgr -v` to list entries;
- `efibootmgr -b XXXX -B` to delete an entry by number.

If you can boot into Windows in UEFI mode, use `bcdedit`:

- `bcdedit /enum firmware` to list entries;
- `bcdedit /delete {GUID}` to delete an entry by its "identifier".

special distro behaviour

Ubuntu

Browser graphic issue:

- <https://askubuntu.com/questions/1413703/graphics-issue-with-google-chrome>

application shortcut

- in folder `/usr/share/applications` or `~/.local/share/applications`
- `.desktop` file example:

```
[Desktop Entry]
Name=Visual Studio Code
Comment=Code Editing. Redefined.
GenericName=Text Editor
Exec=/usr/share/code/code --unity-launch %F
Icon=vscode
Type=Application
StartupNotify=false
StartupWMClass=Code
Categories=TextEditor;Development;IDE;
MimeType=text/plain;inode/directory;application/x-code-workspace;
Actions=new-empty-window;
Keywords=vscode;

[Desktop Action new-empty-window]
Name=New Empty Window
Exec=/usr/share/code/code --new-window %F
Icon=vscode
```

manjaro

vmware screen resolution bug

```
sudo sh -c '(crontab -l; echo "@reboot sleep 5 && systemctl restart vmttoolsd") | sort -u | crontab -'
```

old, not working/necessary

- `sudo pacman -S open-vm-tools`
- `sudo pacman -Su xf86-input-vmmouse xf86-video-vmware mesa gtk2 gtkmm`
- `sudo echo needs_root_rights=yes >>/etc/X11/Xwrapper.config`
- `sudo systemctl enable vmttoolsd`
- `sudo systemctl start vmttoolsd`
- Now, for instance if you've rebooted from the login screen and the resolution is not adapting (Or even after starting the vmttoolsd service nothing happens only then) type in `sudo systemctl restart vmttoolsd`

or create autostart desktop entry:

- `nano /etc/xdg/autostart/restartVmttoolsd.desktop`

enter:

```
[Desktop Entry]
Type=Application
Exec=systemctl restart vmttoolsd
Name=Vmttools restarter
X-GNOME-Autostart-enabled=true
```

Raspberry pi

configs

- use `sudo raspi-config`
- fix ip address
- edit `sudo nano /etc/dhcpd.conf`

```
interface eth0
static ip_address=192.168.0.250/24
static routers=192.168.0.254
static domain_name_servers=192.168.0.254
```

- reboot

mount exfat USB

- <https://pimylifeup.com/raspberry-pi-exfat/>

```
sudo apt-get update
sudo apt-get upgrade

sudo apt-get install exfat-fuse
sudo apt-get install exfat-utils
```

- the pi should detect and mount the exfat usb drive automatically
- check your mounted drives

```
sudo cat /proc/mounts
```

- format drive to exFAT if needed: `mkfs.exfat /dev/sd**`

Mounting an exFAT Drive Manually from Terminal:

- Mounting an exFAT drive utilize the `-t exfat` argument to tell the mount command to recognize the file system as exFAT.

To begin, we must first create a folder where we will mount our desired drives. Create this new exfat folder by running the following command on the Raspberry Pi.

```
sudo mount -t exfat /dev/sd** /media/data1
```

auto mount on boot for exFAT drives

- identify the the disk (/dev/sdb1) first

```
df -h #disk-free, displays available disk space
```

Most external drives will be references under the `/dev/sd**` filesystem name.

- to find UUID and chek the type of drive (should be exFAT): `sudo blkid /dev/sd**`
- edit the fstab file via `sudo nano /etc/fstab` and add a line like:

```
UUID=<uuid> /media/data1 exfat defaults,auto,umask=000,users,rw 0 0
```

linux mint (cinnamon)

- disable moving windows with `alt` + mouse click(and hold)
 - System Settings -> Windows -> Behaviour -> Special key to move and resize windows

Fedora

Updates

- check for updates `dnf check-update`
- summary of available updates `dnf updateinfo` or `dnf updateinfo list`
- upgrade all packages `sudo dnf upgrade`
- reboot afterwards `sudo shutdown -r now`

different locale

- to use a different local in a e.g. browser use: `LANGUAGE=de_AT google-chrome`

Shell

bash basics

default text editor

- for inline shell call: `sudo EDITOR=nano crontab -e`
- add to `~/.bashrc` or `~/.zshrc`:
 - `export EDITOR=nano`
- for default root editor, edit also `/root/.bashrc` or `/root/.zshrc`

```
echo "export VISUAL=nano\nexport EDITOR=nano" >> ~/.bashrc
echo "export VISUAL=nano\nexport EDITOR=nano" >> ~/.zshrc

sudo sh -c "echo -e \"export VISUAL=nano\nexport EDITOR=nano\" >> ~/.bashrc"
sudo sh -c "echo -e \"export VISUAL=nano\nexport EDITOR=nano\" >> ~/.zshrc"
```

login message:

- for local user edit: `nano ~/.profile`
- for global: `/etc/profile` or `/etc/bashrc` (only for bash shell)

add for example:

```
#shell login message
echo -e "\n\n You are working on: \t\e[41m>>> subdomain.my-server.com <<<\e[49m\n\n"
```

custom shell prompt:

- for local user edit: `nano ~/.profile`
- for global: `/etc/profile` or `/etc/bashrc` (only for bash shell)

add for example:

```
#shell login message
alias PS1="\e[0;32m[\u@\h \w]\$ \e[0m"
```

- `\u` - user
- `\h` - host
- `\w` - current path

```
[test@ubuntu1 ~]$ <- in green
```

PS3 for select prompt

https://www.thegeekstuff.com/2008/09/bash-shell-take-control-of-ps1-ps2-ps3-ps4-and-prompt_command/

Terminal colors

- `\e[0;32m`
- `\e[` - Begin color changes
- `0;32m` - Specify the color code
- `[\u@\h \w]\$` - This is the code for your normal BASH prompt (username@hostname Workingdirectory \$)
- `\e[0m` - Exit color-change mode

The first number in the color code specifies the typeface:

- `0` - Normal
- `1` - Bold (bright)
- `2` - Dim
- `4` - Underlined

The second number indicates the color you want:

color	background	text
black	<code>\e[0;40m</code>	<code>\e[0;30m</code>
Red	<code>\e[0;41m</code>	<code>\e[0;31m</code>
Green	<code>\e[0;42m</code>	<code>\e[0;32m</code>
Brown	<code>\e[0;43m</code>	<code>\e[0;33m</code>
Blue	<code>\e[0;44m</code>	<code>\e[0;34m</code>
Purple	<code>\e[0;45m</code>	<code>\e[0;35m</code>
Cyan	<code>\e[0;46m</code>	<code>\e[0;36m</code>

color	background	text
Light gray	\e[0;47m	\e[0;37m

output redirection

- `2>&1` redirect StdErr to StdOut
- `2>&1 |` redirect and Pipe, same as `|&` (not in sh)

Syntax	visible in terminal		visible in file		existing
	StdOut	StdErr	StdOut	StdErr	file
<code>></code>	no	yes	yes	no	overwrite
<code>>></code>	no	yes	yes	no	append
<code>2></code>	yes	no	no	yes	overwrite
<code>2>></code>	yes	no	no	yes	append
<code>&></code>	no	no	yes	yes	overwrite
<code>&>></code>	no	no	yes	yes	append
<code> tee</code>	yes	yes	yes	no	overwrite
<code> tee -a</code>	yes	yes	yes	no	append
<code> & tee</code>	yes	yes	yes	yes	overwrite
<code> & tee -a</code>	yes	yes	yes	yes	append

zsh

all in one script

```
if test -f ~/.zshrc; then
  echo "Installing zsh with oh my zsh and powerlevel10kTheme"

  sudo apt update && sudo apt install curl git zsh
  sh -c "$(curl -fsSL
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)" "" --unattended
  git clone --depth=1 https://github.com/romkatv/powerlevel10k.git ${ZSH_CUSTOM:-$HOME/.oh-my-
zsh/custom}/themes/powerlevel10k
```

```
sed -i '/ZSH_THEME="/c\ZSH_THEME="powerlevel10k/powerlevel10k"' ~/.zshrc
fi
```

To customize prompt, run `p10k configure` or edit `~/.p10k.zsh`.

install zsh

- ZSH (shell with more features)
 - `sudo apt install zsh` or `sudo dnf install zsh` (on Fedora)
- Oh-My-ZSH
 - `sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"`

Oh My Zsh [Cheatsheet](#)

install powerlevel 10k theme

- install font for "powerlevel10k" <https://github.com/romkatv/powerlevel10k#meslo-nerd-font-patched-for-powerlevel10k>
- **choose font Meslo Nerd in preferences of the terminal or editor**
- install "powerlevel10k"-theme for ohmyzsh <https://github.com/romkatv/powerlevel10k>
`git clone --depth=1 https://github.com/romkatv/powerlevel10k.git ${ZSH_CUSTOM:-$HOME/.oh-my-zsh/custom}/themes/powerlevel10k`
- Set `ZSH_THEME="powerlevel10k/powerlevel10k"` in `~/.zshrc`.
 - `sed -i '/ZSH_THEME="/c\ZSH_THEME="powerlevel10k/powerlevel10k"' ~/.zshrc`
- Type `p10k configure` if the configuration wizard doesn't start automatically.

edit for visual studio code

- Press `command + shift + P` in your VSCode window.
- type `settings.json` and click "Open User Settings (JSON)" it.
- insert or add:

```
{
  "terminal.integrated.fontFamily": "MesloLGS NF",
  "terminal.integrated.fontSize": 12,
  "terminal.integrated.defaultProfile.linux": "zsh"
}
```

terminal emulator

terminator

- install: `sudo apt install terminator`

documentation

- Create more terminals by:
 - horizontal split: `Ctrl-Shift-o`
 - vertical split: `Ctrl-Shift-e`
- Shift focus to:
 - next terminal: `Ctrl-Shift-n`
 - previous terminal: `Ctrl-Shift-p`
- New tab: `Ctrl-Shift-t`
- New window: `Ctrl-Shift-i`
- Close terminal or tab:
 - `Ctrl-Shift-w`
 - or right mouse click -> Close
- Close window with all it's terminals and tabs: `Ctrl-Shift-q`
- Reset zoom: `Ctrl-0`

Shell scripting

cheatsheets:

- <https://devhints.io/bash>

bash basics

`$?` last result

conditions

- check if file exists:

```
[ -f /path/to/file ] && do something  
if [ -f /path/to/file ]; then do something; fi
```

find biggest folders of current directory

```
du -hs * | sort -rh | head -5
```

Network

settings

set static ip address

- `sudo nano /etc/network/interfaces`:

```
auto ens18
iface ens18 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    gateway 192.168.0.1
    dns-nameservers 89.207.128.252 89.207.130.252
```

- `sudo reboot`

check local network settings

[linux commands](#)

- `ifconfig -a`
 - alias `nmcli dev list iface eth0 | grep IP4`
- `ip a` [hints](#)

restart network

- `sudo systemctl restart NetworkManager.service`
- `sudo nmcli networking off` and `sudo nmcli networking on`
- `nmtui` a terminal user interface
- `sudo ifdown -a && sudo ifup -a`

dnsmasq

- wildcard usbdomains: <https://saturncloud.io/blog/wildcard-subdomains-with-dnsmasq-a-guide-for-software-engineers/>

network analyse

<https://armann-systems.com/wiki/netzwerkd Diagnose-tools-unter-linux-ein-umfassender-leitfaden/>

check open ports

```
ss -ltup
```

netstat

- install with `sudo apt install net-tools`
- show all active tcp connections: `netstat -tn`
- scan all ports of ip: `nmap -p 1-65535 192.168.1.1`
- check a specific port on the system (and get infos about process):

```
sudo netstat -nlp | grep PORT_NUMBER  
ps -Flww -p THE_PID
```

nmap

missing

proxy settings

global settings in `/etc/environment`

```
http_proxy=http://www-zproxy.myproxy.com:80/  
https_proxy=http://www-zproxy.myproxy.com:80/  
ftp_proxy=ftp://www-zproxy.myproxy.com:80/  
no_proxy=localhost,myproxy.com,.myproxy.com,10.0.0.0/8,127.0.0.1  
soap_use_proxy=on
```

Permissions, Groups

TODO

<https://www.pluralsight.com/blog/it-ops/linux-file-permissions>

groups

- add primary group (restart is necessary): `sudo usermod -g primarygroupname username`
- add group: `sudo usermod -a -G secondarygroupname username`

desc:

- `-g` primary group assigned to the users
- `-G` Other groups the user belongs to
- `-a` Add the user to the supplementary group(s)

chmod

chown

DE Desktop environments

gnome

taskbar changes

- gnome extensions
 - install browser extension (in not snap browser)
 - install `chrome-gnome-shell` or `gnome-browser-connector` via `apt-get`
 - install extensions:
 - network stats <https://extensions.gnome.org/extension/4308/network-stats/>
 - system monitor
 - <https://extensions.gnome.org/extension/3010/system-monitor-next/>
 - requires `sudo apt install libgtop2-dev`
- install `dconf editor` in ubuntu store or:
- Shorten the dock launcher `gsettings set org.gnome.shell.extensions.dash-to-dock extend-height false`
- Move Show Applications button to the left `gsettings set org.gnome.shell.extensions.dash-to-dock show-apps-at-top true`
- Autohide and show on mouse over: `gsettings set org.gnome.shell.extensions.dash-to-dock require-pressure-to-show false` `gsettings set org.gnome.shell.extensions.dash-to-dock intellihide true`

Gnome Taskbar extensions

- install manager: `sudo apt install gnome-shell-extension-manager`
- extensions
 - System-Monitor-Next (<https://extensions.gnome.org/extension/3010/system-monitor-next/>)
 - requires `sudo apt install libgtop2-dev`

other interesting extensions

- radio: <https://extensions.gnome.org/extension/836/internet-radio/>
 - requires `sudo apt install gir1.2-gst-plugins-base-1.0`

Different issues and settings

time

time sync

- simple time check: `date`
- more details: `timedatectl`
- activate time sync: `sudo timedatectl set-ntp true`
 - necessary: `apt install systemd-timesyncd`

CLI tools

find, search, grep

find

use cases:

- `find ./subfolder -name sample.txt` - Search a file with specific name
- `find ./subfolder -name *.txt` - Search a file with pattern
- `find ./GFG -name sample.txt -exec rm -i {} \;` - find and delete a file with confirmation
- `find ./subfolder -empty` - Search for empty files and directories
- `find ./subfolder -perm 664` - Search for file with entered permissions
- `find ./ -type f -name "*.txt" -exec grep 'some Phrase' {} \;` - Search text within multiple files

more examples

```
find /path -name *.txt
find /path -type f -name test.txt
find /path -name failed*.* -type f
find /path -type f -not -name "*.html"
find / -name "file.txt" -size +4M
find /dev/ -type b -name "sda*"
find /*file*
```

grep

```
grep [OPTION]... PATTERNS [FILE]...
```

- options
 - `-B <numb>` - show numb lines before match
 - `-A <numb>` - show numb lines after match
 - `-i` - ignore case distinctions in patterns and data
 - `-r`, `--recursive` - like `--directories=recurse`
 - `-v`, `--invert-match` - To display only the lines that do not match a search pattern

- `--exclude-dir=<foldername>` - exclude folder from search
- `-n`, `--line-number` - Prefix each line of output with the 1-based line number within its input file.

examples

- `grep -ir --exclude-dir=vendor skeleton .` - find all occurrences of "skeleton" in the current working dir
- `grep -i "some string" path/**/*.log` - search string in log files

sed

```
sed -i 's/SEARCH_REGEX/REPLACEMENT/g' INPUTFILE
```

- `-i` - By default, sed writes its output to the standard output. This option tells sed to edit files in place. If an extension is supplied (ex `-i.bak`), a backup of the original file is created.
- `s` - The substitute command, probably the most used command in sed.
- ``//`` - Delimiter character. It can be any character but usually the slash (/) character is used.
- `SEARCH_REGEX` - Normal string or a regular expression to search for.
- `REPLACEMENT` - The replacement string.
- `g` - Global replacement flag. By default, sed reads the file line by line and changes only the first occurrence of the `SEARCH_REGEX` on a line. When the replacement flag is provided, all occurrences are replaced.
- `INPUTFILE` - The name of the file on which you want to run the command.

Cronjob, tar, autostart, sudo, In

cronjob

- execute all listed cronjobs: `crontab -l | grep -v '^#' | cut -f 6- -d ' ' | while read CMD; do eval $CMD; done`
- send output to mail address
 - `MAILTO="empfänger@adresse.de"`
 - <https://de.godaddy.com/blog/behalten-sie-ihre-cron-jobs-unter-kontrolle/>

tar

usage: `tar [OPTION...] [FILE]...`

- options:
 - `-c` Ein neues Archiv erzeugen.
 - `-d` Dateien im Archiv und im Dateisystem miteinander vergleichen.
 - `-f` Archiv in angegebene Datei schreiben / Daten aus angegebener Datei lesen.
Diese Option muss die letzte sein, da die nachfolgende Zeichen als Datei interpretiert werden. Z.B. würde `-cfv` zu einer Fehlermeldung führen. Korrekt wäre `-vcf`.
 - `-x` Dateien aus einem Archiv extrahieren.
- create archive:
 - unzipped
 - `tar -czf archiv.tar.gz file` Create archiv.tar.gz
 - zipped
 - `tar -cf archiv.tar path/` Create archiv.tar with all subdir
 - `tar -cf archiv.tar datei_1.txt *.pdf` Create archiv.tar with 1 file and all pdfs
- anhängen
 - `tar -rf archiv.tar datei_1.txt` Add 1 specific file to archiv (uncompressed)
- extract archive: ACHTUNG - tar überschreibt existierende automatisch
 - `tar -xf archiv.tar` Extract files
 - `tar -xzf archiv.tar.gz -C /` Extract gzip archives
 - `tar -xzf archiv.tar.gz -C /path` Extract gzip archives to /path
- anzeigen `tar -tvf archiv.tar` Show which files are in archiv.tar

autostart

- create new file
 - user only `sudo nano ~/.config/autostart/<some_name>.desktop`
 - global `sudo nano /etc/xdg/autostart/<some_name>.desktop`
- insert content like:

```
[Desktop Entry]
Type=Application
Name=Musterprogramm
Exec=Auszuführendes Kommando
```

sudo

add user to sudo

```
su -
usermod -aG sudo <username>
```

no password sudo

- edit sudoers file with `sudo visudo`
- add line at **end of file** (important for not be overridden):

```
username ALL=(ALL) NOPASSWD:ALL
```

links

- `ln -s /Pfad/zur/Datei /Pfad/zum/symLink`
 - `-s` (symbolic) erstellt einen symbolischen Link statt eines Hardlinks.
 - `-f` (force) aktualisiert den Link und entfernt existierende Ziele.
 - `-i` (interactive) fragt nach, bevor Ziele entfernt werden (setzt -s voraus).
 - `-r` (relative) erstellt symbolische Links relativ zum Link-Speicherort.

CLI Editor

nano

cli parameters/flags

- `-c`, `--const` - Constantly display the cursor position and line number on the statusbar. Note that this overrides `-U`.
- `-U`, `--quickblank` - Do quick statusbar blanking. Statusbar messages will disappear after 1 keystroke instead of 25. Note that `-c` overrides this.
- `-S`, `--smooth` - Enable smooth scrolling. Text will scroll line-by-line, instead of the usual chunk-by-chunk behavior.
- `-m`, `--mouse` - Enable mouse support, if available for your system. When enabled, mouse clicks can be used to place the cursor, set the mark (with a double click), and execute shortcuts. The mouse will work in the X Window System, and on the console when gpm is running.

set defaults by editing `/etc/nanorc` or in `~/.nanorc`: (some recommendations)

```
## Constantly display the cursor position in the status bar or minibar.
set constantshow

## Display line numbers to the left (and any anchors in the margin).
set linenumbers

## Enable mouse support, if available for your system.  When enabled,
## mouse clicks can be used to place the cursor, set the mark (with a
## double click), and execute shortcuts.  The mouse will work in the
## X Window System, and on the console when gpm is running.
set mouse

## Make the Home key smarter: when Home is pressed anywhere but at the
## very beginning of non-whitespace characters on a line, the cursor
## will jump to that beginning (either forwards or backwards).  If the
```

```
## cursor is already at that position, it will jump to the true start
## of the line (the left edge).
# set smarthome

## Spread overlong lines over multiple screen lines.
# set softwrap
```

useful short cuts

shortcuts are displayed on the bottom, maximize to see more

- `Ctrl` + `k` cut selection or current line
- `Ctrl` + `u` paste at cursor position
-

vim

tutorials

- [VimTutor](#)
- <https://danielmiessler.com/study/vim/>

vim modes

- command mode (default)
You can switch to any mode from this mode. You can't do this in any other modes. To switch from one mode to another, you have to come to Command Mode first and then navigate to the other mode. The commands that you run without any prefix (colon) indicate that you're running the command in command mode.
- `i` - insert mode
This mode is used to edit the contents of the file. You can switch to insert mode by pressing `i` from command mode. You can use the `Esc` key to switch back to command mode.
- `:` - command-line mode
You can use this mode to play around with some commands. But the commands in this mode are prefixed with a colon (`:`). You can switch to this mode by pressing `:` (colon) in command mode.
- `v` - visual mode

You use this mode to visually select some text and run commands over that section of code. You can switch to this mode by pressing `v` from the command mode.

base commands

- create/edit and save/quit
 - `:edit sample.txt` create/edit (new file)
 - `:w` - save and continue
 - `:q` - quit
 - `:q!` - quit without save
 - `:wq` - save and quit
- modes
 - `i` - insert
 - `v` - visual
 - `ESC` - return to command mode (default start point)
- undo/redo
 - `u` - undo
 - `U` - undo a whole line
 - `3u` - undo 3 times
 - `:undolist` - list available undo options
 - `CTRL + R` - redo

movement

- general

```
^
k[]   Hint: The h key is at the left and moves left.
< h   l >[] The l key is at the right and moves right.
j[]   The j key looks like a down arrow.
v
```

- jump to lines
 - `gg` - first line
 - `G` - end of file
 - `42G` - to line 42
- counted motions
 - `2w` move cursor to words forward
 - `3e` move cursor to the end of the third word
 - `0` move to line start

text editing

- motion list
 - `w` - until the start of the next word, EXCLUDING its first character.
 - `e` - to the end of the current word, INCLUDING the last character.
 - `$` - to the end of the line, INCLUDING the last character.
- inserting
 - `i` - insert before cursor
 - `A` - append after the line
- deletion
 - `x` - delete current character
 - `dw` - delete from cursor up to the next word
 - `de` - delete from cursor up to the end of the current word
 - `d$` - delete from cursor to the end of the line
 - `d0` - delete to the beginning of line
 - `dd` - delete whole line
 - `d2w` - delete two words
 - `d2d` - delete two lines
- copy/cut/paste
 - `y` - copy selected text
 - `yy` - copy current line
 - `y5y` - copy 5 lines
 - `d` - cut selected text
 - `p` - paste copied text
 - `:t.` - duplicate current line
 - `yy`, `p` - duplicate current line

find and replace

searching only

- `/` search forward (forward slash)
- `?` search backward
- case insensitiv: append `\c`

example: `/Linux\c` + Enter

- after first search press `n` to search for the next occurrence or uppercase `N` to search backwards

find and replace

```
:[range]s/{pattern}/{string}/[flags]
```

- `[range]` indicates that you can pass the range of lines. The range is separated by a comma.
 - Pass `%` to find and replace in all lines

- `5,10` To find and replace between lines 5 to 10
- `.` current line
- `$` the last line of the file
- `{pattern}` indicates the pattern to find the text. You can pass regex patterns here.
- `{string}` is the string to replace in the found text.
- `[flags]` indicates if you wish to pass any additional flags. By default, this does a case-sensitive search.
 - `i` case-insensitive search
 - `c` confirm before replacing
 - `g` indicates making the change globally

example

`:2,3s/Hi/Hello and Welcome/gci` -> replace with Hello and Welcome (y/n/a/q/l/^E/^Y)?

- `y` - Replace the match
- `n` - Skip the match
- `a` - Substitutes the match and all remaining occurrences of the match
- `q` or `Esc` - Quit substitution
- `l` - Replace the match and quit
- `CTRL+Y` - Scroll the screen down
- `CTRL+E` - Scroll the screen up

user management

create user

- `useradd john`
- set password `passwd john`
- create user with root privileges: `useradd -ou 0 -g 0 john`
 - check if id are set correctly:

```
$ grep john /etc/passwd  
john:x:0:0:./home/john:/bin/sh
```

- add user to root group `usermod -a -G root john`
- delete user `userdel john`

disk/file management

disk

show disk size

- check disk space: `df`
 - `-h` - human readable

show folder size

- show folder size: `sudo du -shc ./*`

find biggest folders of current directory

```
du -hs * | sort -rh | head -5
```

file movements

rsync

Links

- <https://wiki.ubuntuusers.de/rsync/>

options

- `a` - contains **highly recommended**
 - `-r` copy subfolder
 - `-l` copy symbolic links
 - `-p` keep permissions of source files
 - `-t` keep timestamps of source files

- `-g` keep group permissions of source files
- `-o` keep user permissions of source files (only root)
- `-D` keep device permissions of source files (only root)
- `v` - verbose
- `h` - human readable
- `z` - with compression
- `-e` - remote shell auswählen
 - `-e 'ssh -p 222'` ändert ssh port auf e.g. 222

local to remote server

```
rsync [option] [source] user@hostname-or-ip:[destination path]
rsync -avhz /home/source-rsync/ user@your-remote-server.com:/home/dest-rsync/
```

remote to local server

```
rsync -avhz user@your-remote-server.com:/home/dest-rsync/ /home/source-rsync/
```

scp

```
scp <Optionen> <Quellpfad> <Zielpfad>
```

- Quellpfad: `<Nutzer>@<Host>:<Verzeichnis/Datei.Endung>`
- optionen:
 - `-C` Kompression nutzen (compression)
 - `-p` Datei-Attribute auf Zieldateien übertragen (permissions)
 - `-r` Verzeichnisse rekursiv kopieren (recursive)
 - `-v` Erweiterte Ausgabe anzeigen (verbose)
 - `-q` Ausgabe unterdrücken (quiet)
 - `-3` Daten durch lokales System senden (third party)

example:

```
scp -r srv01-local_draab:/var/lib/docker/volumes/wireguard_wireguard-config/_data/
/home/danielraab/wireguard/
```

tmux

links

[tutorial](#) [cheatsheet](#)

configs

edit file `~/.tmux.conf` as you like

-> [gpakosz github](#)

•

keyboard shortcuts

`Ctrl + b` `?` - show keyboard shortcut overview

pane

- `Ctrl+b "` — split pane horizontally.
- `Ctrl+b %` — split pane vertically.
- `Ctrl+b` `arrow key` — switch pane.
- `Ctrl+b z` - maximize pane (and undo maximize)
- Hold `Ctrl+b`, don't release it and hold one of the `arrow keys` — resize pane.

windows

- `Ctrl+b c` — (c)reate a new window.
- `Ctrl+b w` — List all windows.
- `Ctrl+b n` — move to the (n)ext window.
- `Ctrl+b p` — move to the (p)revious window.

modes

CTRL+b then [Enter copy mode. q Exit copy mode. SPACE Start text selection in copy mode.
ENTER Copy the selected text. ESC Clear the selected text and exit the copy mode. CTRL+b then]
Paste the copied text. h Move the cursor left. j Move the cursor down. k Move the cursor up. l Move

the cursor right. w Move the cursor one word forward. b Move the cursor one word backward. CTRL+u Scroll up half a page. CTRL+d Scroll down half a page. PgUp Scroll up full page. PgDn Scroll down full page.

configs

adaption for zsh powershell10k

Your tmux has two issues.

It cannot display 256 colors. To fix this, create `~/.tmux.conf` with `set -g default-terminal screen-256color` in it and reboot your machine.

It cannot display non-ascii characters. To fix this, install and enable a UTF-8 locale in your OS and reboot your machine.

You can verify that you've fixed tmux by running the following command:

```
print -P -- '---> %F{70}\u276E\u276F%f <--'
```

basic

suspend and continue

- `Ctrl - z` suspend
- command `fg` to continue

special tools

bpftrace

[homepage docs](#)

```
sudo bpftrace --unsafe -e 'tracepoint:signal:signal_generate /args->sig ==9/ {
printf("SIGKILL: target pid=%d sender pid=%d processname=", args->pid, pid);
system("ps -p %d -o comm | sed -n \"2 p\\\"", pid);
}'
```

New Server Setup

```
#!/usr/bin/bash
sudo apt update && sudo apt upgrade
sudo apt -y install fail2ban ufw git curl zsh

cat >> ~/.ssh/authorized_keys << EOF
---pub ssh key---
EOF

echo "vps2" > /etc/hostname

cat >> /etc/hosts << EOF
127.0.0.1      vps2
46.38.233.182 vps2.draab.at vps2
EOF

printf "[sshd]\nenabled = true\nbanaction = iptables-multiport" > /etc/fail2ban/jail.local
systemctl enable fail2ban
ufw allow 222
ufw enable

sed -i -e '/^\(#\|\\\)PermitRootLogin/s/^\.*$/PermitRootLogin no/' /etc/ssh/sshd_config
sed -i -e '/^\(#\|\\\)PasswordAuthentication/s/^\.*$/PasswordAuthentication no/'
/etc/ssh/sshd_config
sed -i -e '/^\(#\|\\\)Port/s/^\.*$/Port 222/' /etc/ssh/sshd_config
sed -i -e '/^\(#\|\\\)KbdInteractiveAuthentication/s/^\.*$/KbdInteractiveAuthentication no/'
/etc/ssh/sshd_config
sed -i -e '/^\(#\|\\\)ChallengeResponseAuthentication/s/^\.*$/ChallengeResponseAuthentication
no/' /etc/ssh/sshd_config
sed -i -e '/^\(#\|\\\)MaxAuthTries/s/^\.*$/MaxAuthTries 2/' /etc/ssh/sshd_config
sed -i -e '/^\(#\|\\\)AllowTcpForwarding/s/^\.*$/AllowTcpForwarding yes/' /etc/ssh/sshd_config
sed -i -e '/^\(#\|\\\)X11Forwarding/s/^\.*$/X11Forwarding no/' /etc/ssh/sshd_config
sed -i -e '/^\(#\|\\\)AllowAgentForwarding/s/^\.*$/AllowAgentForwarding no/' /etc/ssh/sshd_config
sed -i -e '/^\(#\|\\\)AuthorizedKeysFile/s/^\.*$/AuthorizedKeysFile .ssh\/authorized_keys/'
/etc/ssh/sshd_config
sed -i '$a AllowUsers danielraab' /etc/ssh/sshd_config
```

```
curl -sSL https://get.docker.com | sh
usermod -aG docker danielraab
newgrp docker

# - curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.3/install.sh | bash
# - \. "$HOME/.nvm/nvm.sh"
# - nvm install 22

sh -c "$(curl -fsSL
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)" "" --unattended

reboot
```