ARCH LINUX

POST-INSTALL GUIDE

Everything a modern user needs — drivers, audio, GPU, battery, fingerpri-nt auth, DE tweaks, containers & beyond.

Plasma Desktop Edition · 2025

SECTION 1

01

System Foundation

pacman · AUR · mirrors · locale · firmware · kernel

SECTION 1 · SYSTEM FOUNDATION

First Boot Checklist

1. **Update mirrors with reflector**

```
reflector --country Poland,Germany --sort rate --save /etc/pacman.d/mirrorlist
```

2. **Full system update**

```
sudo pacman -Syu
```

3. **Install base-devel + git (for AUR)**

```
sudo pacman -S base-devel git
```

4. **Install yay (AUR helper)**

```
git clone https://aur.archlinux.org/yay.git && cd yay && makepkg -si
```

5. **Enable parallel downloads**

```
Edit /etc/pacman.conf → ParallelDownloads = 5
```

SECTION 1 · SYSTEM FOUNDATION

Locale, Timezone & Hostname

Locale

Edit /etc/locale.gen

Uncomment your locale

(e.g. en_US.UTF-8)

Run: sudo locale-gen

Set: /etc/locale.conf

LANG=en_US.UTF-8

Timezone

sudo timedatectl set-timezone

Europe/Warsaw

Enable NTP sync:

sudo timedatectl

set-ntp true

Hostname

sudo hostnamectl

set-hostname mymachine

Edit /etc/hosts:

127.0.1.1 mymachine

.localdomain mymachine

SECTION 1 · SYSTEM FOUNDATION

Kernel Selection & CPU Microcode

Available Kernels

linux	Standard — always keep this
linux-lts	LTS — stability over features
linux-zen	Desktop-tuned, lower latency
linux-hardened	Security-hardened, restrictive
linux-rt	Real-time kernel (audio prod.)

CPU Microcode Updates

CRITICAL — install before anything else!

```
# Intel:
sudo pacman -S intel-ucode
# AMD:
sudo pacman -S amd-ucode
```

Firmware (fwupd)

```
sudo pacman -S fwupd
sudo fwupdmgr refresh
sudo fwupdmgr get-updates
sudo fwupdmgr update
```

SECTION 2

02

Display & GPU

NVIDIA · AMD · Intel · Wayland · multi-monitor

SECTION 2 · DISPLAY & GPU

GPU Drivers — The Right Way

NVIDIA

```
sudo pacman -S nvidia nvidia-utils
nvidia-settings lib32-nvidia-utils
```

- Use proprietary driver — nvidia-open for Turing+
- Add nvidia_drm.modeset=1 to kernel params
- For Wayland: KWIN_DRM_USE_EGL_STREAMS=1
- Check: nvidia-smi

AMD

```
sudo pacman -S mesa lib32-mesa
vulkan-radeon lib32-vulkan-radeon
libva-mesa-driver
```

- Open source (amdgpu) — works out of the box
- Wayland native — no quirks needed
- ROCm for compute: sudo pacman -S rocm-opencl-runtime
- Check: glxinfo | grep renderer

Intel

```
sudo pacman -S mesa lib32-mesa
vulkan-intel lib32-vulkan-intel
intel-media-driver
```

- Use intel-media-driver (iHD) for Gen8+
- Avoid xf86-video-intel — modesetting is better
- Wayland works perfectly on recent gens
- Check: vainfo

SECTION 2 · DISPLAY & GPU

Wayland vs X11 — and Multi-Monitor

Feature	Wayland (KWin)	X11 (Xorg)
Screen tearing	✓ None (composited)	⚠ Needs compositing
NVIDIA support	⚠ Needs modeset=1	✓ Mature
Screen sharing	✓ PipeWire/xdg-portal	✓ Native
HiDPI / mixed DPI	✓ Per-display scaling	⚠ Global only
Gaming (Steam)	✓ GameScope works	✓ Traditional

Multi-Monitor Quick Setup

```
System Settings → Display & Monitor → Arrange displays drag-and-drop
For CLI: kscreen-doctor output.HDMI-A-1.enable output.HDMI-A-1.mode.1920x1080@60
Autorandr (X11): sudo pacman -S autorandr && autorandr --save home
```

SECTION 3

03

Audio

PipeWire · ALSA · Bluetooth audio · EQ · Pro Audio

SECTION 3 · AUDIO

PipeWire — The Modern Audio Stack

Installation

```
sudo pacman -S pipewire pipewire-alsa
  pipewire-pulse pipewire-jack
  wireplumber

systemctl --user enable --now pipewire
systemctl --user enable --now pipewire-pulse
systemctl --user enable --now wireplumber
```

Why PipeWire?

- Replaces PulseAudio AND JACK simultaneously
- Ultra-low latency (pro audio ready)
- Bluetooth aptX-HD / LDAC support

Essential GUI Tools

`pavucontrol`	— Volume control per app
`easyeffects`	— Parametric EQ, compressor, noise suppressor
`helvum`	— PipeWire patchbay (visual routing)

Bluetooth Audio

```
sudo pacman -S bluez bluez-utils pulseaudio-bluetooth
systemctl enable --now bluetooth
# In /etc/bluetooth/main.conf: AutoEnable=true  |  Enable=Source,Sink,Media,Socket
```

SECTION 4

04

Power & Battery

TLP · power-profiles-daemon · thermald · hibernate

SECTION 4 · POWER & BATTERY

Battery Life & Thermal Management

TLP — Battery Optimizer

```
sudo pacman -S tlp tlp-rdw
sudo systemctl enable --now tlp

# For ThinkPads also install:
sudo pacman -S tp_smapi acpi_call

# Check status:
sudo tlp-stat -b
```

power-profiles-daemon (KDE native)

⚠ Use either TLP OR power-profiles-daemon — not both!
KDE Plasma uses power-profiles-daemon by default.
For laptops: TLP gives finer control.
Check: powerprofilesctl list

thermald — Intel Thermal Control

```
sudo pacman -S thermald
```

Hibernate / Suspend-then-Hibernate

1. Create swap file ≥ RAM size 2. Add resume= to kernel params (UUID of swap partition)
3. Add resume hook to /etc/mkinitcpio.conf HOOKS: (... resume ...) 4. mkinitcpio -P
For suspend-then-hibernate: systemctl edit systemd-suspend-then-hibernate.service → HibernateDelaySec=30min

SECTION 5

05

Input & Fingerprint

fprintd · libinput · gestures · keyboard layouts · IME

SECTION 5 · INPUT & FINGERPRINT

Fingerprint Authentication with fprintd

1 Install fprintd

```
sudo pacman -S fprintd
sudo pacman -S libfprint  # driver library
```

2 Enroll your fingerprint

```
fprintd-enroll  # scan finger 5 times
fprintd-verify   # test it
```

3 Configure PAM (sudo)

```
# /etc/pam.d/sudo — ADD at top:
auth sufficient pam_fprintd.so
```

4 Configure PAM (login / SDDM)

```
# /etc/pam.d/sddm — ADD at top:
auth sufficient pam_fprintd.so
```

5 Configure PAM (polkit)

```
# /etc/pam.d/polkit-1 — ADD at top:
auth sufficient pam_fprintd.so
```

✓ TIP: For KDE Wallet auto-unlock with fingerprint, install kwallet-pam and add pam_kwallet5.so to PAM stack.

SECTION 5 · INPUT & FINGERPRINT

Touchpad Gestures & Keyboard Magic

Touchpad Gestures (libinput-gestures)

```
yay -S libinput-gestures
libinput-gestures-setup install
libinput-gestures-setup start
libinput-gestures-setup autostart

# Config ~/.config/libinput-gestures.conf:
gesture swipe up 3 xdotool key super
gesture swipe left 3 xdotool key alt+Right
```

Keyboard Layout & Input Method

Add layout in KDE:	System Settings → Input Devices → Keyboard
CLI permanent layout:	localectl set-x11-keymap pl
Caps→Ctrl (power move):	setxkbmap -option ctrl:nocaps
fcitx5 (CJK IME):	sudo pacman -S fcitx5 fcitx5-qt fcitx5-gtk
IBus (alternative IME):	sudo pacman -S ibus
Custom compose key:	System Settings → Keyboard → Advanced

Gaming Mice & Special Peripherals

Logitech: yay -S piper (GUI) + sudo pacman -S ratbagd && sudo systemctl enable ratbagd
Razer: yay -S openrazer-meta polychromatic | Corsair: yay -S ckb-next
Wacom tablets: sudo pacman -S xf86-input-wacom + KDE System Settings → Drawing Tablet

SECTION 6

Networking

NetworkManager · VPN · Firewall · DNS-over-HTTPS · SSH

SECTION 6 · NETWORKING

NetworkManager, VPN & Firewall

NetworkManager (the must-have)

```
sudo pacman -S networkmanager nm-connection-editor
nm-applet network-manager-applet
sudo systemctl enable --now NetworkManager

# CLI management:
nmcli device wifi list
nmcli device wifi connect "SSID" password "pass"
```

Firewall — nftables / ufw

```
sudo pacman -S ufw
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow ssh
sudo ufw enable
sudo systemctl enable ufw

# GUI: sudo pacman -S gufw
```

VPN

```
# WireGuard:
sudo pacman -S wireguard-tools
# OpenVPN:
sudo pacman -S openvpn networkmanager-openvpn
# Import .ovpn: nmcli connection import type
  openvpn file my.ovpn
```

DNS-over-HTTPS / Privacy

```
# systemd-resolved with DoT:
sudo pacman -S systemd-resolvconf
# /etc/systemd/resolved.conf:
[Resolve]
DNS=9.9.9.9
DNSOverTLS=yes
FallbackDNS=1.1.1.1
```

SECTION 7

07

Security

disk encryption · SSH hardening · AppArmor · Yubikey · audit

SECTION 7 · SECURITY

System Hardening Essentials

SSH Hardening (/etc/ssh/sshd_config)

```
PermitRootLogin no
PasswordAuthentication no
PubkeyAuthentication yes
Port 2222    # change default!
AllowUsers yourusername
```

AppArmor (MAC security)

```
sudo pacman -S apparmor
# Kernel param: lsm=landlock,lockdown,yama,
  integrity,apparmor,bpf
sudo systemctl enable --now apparmor
aa-status    # check profiles
```

LUKS Full-Disk Encryption (if not done at install)

```
# Already encrypted? Check:
lsblk -f | grep crypto
# Add 2nd LUKS passphrase (backup):
sudo cryptsetup luksAddKey /dev/sdXn
# Backup LUKS header (CRITICAL):
sudo cryptsetup luksHeaderBackup /dev/sdXn --header-backup-file luks-
header.img
```

Yubikey / Hardware Token

```
sudo pacman -S yubikey-manager pam-u2f
# Generate u2f config:
mkdir -p ~/.config/Yubico
pamu2fcfg > ~/.config/Yubico/u2f_keys
# /etc/pam.d/sudo add:
auth required pam_u2f.so
```

SECTION 8

08

Plasma DE Customization

themes · fonts · KWin effects · SDDM · widgets · Konsole

SECTION 8 · PLASMA DE CUSTOMIZATION

KDE Plasma — Deep Tuning

Fonts & Rendering

1. `sudo pacman -S ttf-jetbrains-mono nerd-fonts-jetbrains-mono`
2. `sudo pacman -S noto-fonts noto-fonts-emoji noto-fonts-cjk`
3. `/etc/fonts/local.conf → enable subpixel hinting`
4. KDE Settings → Fonts → Force DPI: 96 (or 192 for HiDPI)
5. `yay -S ttf-ms-fonts  # Microsoft compat fonts`

KWin Effects & Performance

1. System Settings → Desktop Effects → enable Blur, Wobbly
2. `KWin scripts: bismuth (tiling) — yay -S kwin-bismuth`
3. Disable compositing temporarily: Alt+Shift+F12
4. `Latency: kwriteconfig5 --file kwinrc --group Compositing --key LatencyPolicy low`
5. Gaming mode: toggle via kwin dbus to disable effects

`SDDM Theme: yay -S sddm-theme-sugar-candy-git → /etc/sddm.conf.d/sddm.conf → [Theme] Current=sugar-candy`

SECTION 8 · PLASMA DE CUSTOMIZATION

Shell, Terminal & Essential Tools

ZSH + Oh-My-Zsh + Powerlevel10k

```
sudo pacman -S zsh
chsh -s /bin/zsh

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

# Powerlevel10k theme:
yay -S zsh-theme-powerlevel10k
```

Must-Have ZSH Plugins

`zsh-autosuggestions`	— Fish-like autocomplete
`zsh-syntax-highlighting`	— Color-coded commands
`zsh-history-substring-search`	— ↑↓ search history
`fzf`	— Fuzzy finder for everything
`thefuck`	— Fix mistyped commands
`autojump / zoxide`	— Smart cd replacement

Yakuake (drop-down terminal)

sudo pacman -S yakuake
Autostart in KDE Session Settings.
Press F12 to toggle. Configure via System Tray.

CLI Power Tools

```
sudo pacman -S bat eza fd ripgrep btop htop
nemo  # File manager alt  |  mc  # Midnight Commander
tmux  # Terminal multiplexer  |  neovim  # Editor
```

SECTION 9

09

Dev & Containers

Docker · Podman · Flatpak · Distrobox · Virt-Manager

SECTION 9 · DEV & CONTAINERS

Docker, Podman, Distrobox & KVM

Docker

```
sudo pacman -S docker docker-compose
sudo systemctl enable --now docker
sudo usermod -aG docker $USER
# Log out and back in!
docker run hello-world
```

Podman (rootless Docker alt)

```
sudo pacman -S podman podman-compose
# No daemon needed, rootless by default
podman run hello-world

# Docker compatibility alias:
alias docker=podman
```

Distrobox — Any distro inside Arch

```
sudo pacman -S distrobox
# Create Ubuntu 24.04 box:
distrobox create --name ubuntu \
  --image ubuntu:24.04
distrobox enter ubuntu
# Install Ubuntu apps — they appear
# in your KDE app launcher!
```

KVM / QEMU (full VMs)

```
sudo pacman -S qemu-full virt-manager
  libvirt dnsmasq
sudo systemctl enable --now libvirtd
sudo usermod -aG libvirt $USER
# Enable nested VT-x/AMD-V:
echo 'options kvm_intel nested=1' \
  | sudo tee /etc/modprobe.d/kvm.conf
```

SECTION 9 · DEV & CONTAINERS

Flatpak, Snap & Application Ecosystem

Flatpak Setup

```
sudo pacman -S flatpak
flatpak remote-add --if-not-exists flathub
  https://dl.flathub.org/repo/flathub.flatpakrepo

# Discover apps: flathub.org
# Install example:
flatpak install flathub com.spotify.Client

# KDE Discover can manage Flatpaks
# natively — just enable it!
```

Must-Have Flatpak Apps

Bottles:	Run Windows .exe apps (Wine wrapper)
Flatseal:	Manage Flatpak permissions
Heroic Games Launcher:	Epic Games + GOG on Linux
Obsidian:	Markdown knowledge base
Thunderbird:	Email with CalDAV sync
Signal Desktop:	Secure messaging
Kdenlive:	Video editor (KDE-native)

Gaming: Steam + Proton

```
sudo pacman -S steam  # Enable multilib first in /etc/pacman.conf!
Steam → Settings → Compatibility → Enable Steam Play for all titles → Proton Experimental
sudo pacman -S gamemode lib32-gamemode  # Performance mode during games
yay -S proton-ge-custom  # Better compatibility for some titles
```

SECTION 10

10

System Maintenance

backups · pacman orphans · journal · crash recovery · snapper

SECTION 10 · SYSTEM MAINTENANCE

Backups, Snapshots & System Health

Timeshift / Snapper (BTRFS Snapshots)

```
sudo pacman -S snapper snap-pac
# Creates auto snapshots on pacman events
snapper -c root create-config /
# GUI: yay -S snapper-gui
```

Restic — Encrypted Cloud Backups

```
sudo pacman -S restic
restic -r s3:s3.amazonaws.com/bucket init
restic -r /mnt/backup backup /home/user
restic -r /mnt/backup snapshots
```

Pacman Maintenance

```
# Remove orphan packages:
sudo pacman -Rns $(pacman -Qdtq)
# Clean package cache (keep 2 versions):
sudo paccache -rk2
# Check for broken symlinks:
sudo pacman -Dk
# Mirror update:
sudo reflector --save /etc/pacman.d/mirrorlist
```

System Logs & Journal

```
# Limit journal size:
journalctl --disk-usage
sudo journalctl --vacuum-size=500M
# /etc/systemd/journald.conf:
SystemMaxUse=500M
# Real-time logs:
journalctl -f -p err
# Check boot errors:
journalctl -b -p err
```

SECTION 10 · SYSTEM MAINTENANCE

When Things Go Wrong — Recovery Guide

⚠ System won't boot / broken after update? Here's your survival kit.

1. Boot from Arch ISO → chroot in

```
mount /dev/sdXn /mnt
mount /dev/sdXp /mnt/boot  # if separate
arch-chroot /mnt
```

2. Downgrade a broken package

```
sudo pacman -U /var/cache/pacman/pkg/
  package-old-version.pkg.tar.zst
# Or: yay -S downgrade && downgrade pkg
```

3. Regenerate initramfs

```
mkinitcpio -P
# After kernel/hooks changes
# Then reboot
```

4. Reinstall GRUB

```
grub-install /dev/sdX
grub-mkconfig -o /boot/grub/grub.cfg
```

Pro Tips

- Before ANY major update: snapper create --description 'pre-update' or timeshift --create
- Read /var/log/pacman.log to see what changed: grep 'upgraded' /var/log/pacman.log | tail -20
- Arch news feed: https://archlinux.org/news — ALWAYS check before upgrading (subscribe via RSS!)

BONUS · PERIPHERALS & MISCELLANEOUS

Printing, Scanning & Other Peripherals

Printing (CUPS)

```
sudo pacman -S cups cups-pdf system-config-printer
hplip  # HP printers
brlaser  # Brother printers (AUR: yay -S brlaser)
sudo systemctl enable --now cups
# Access: http://localhost:631
```

Scanning (SANE)

```
sudo pacman -S sane
yay -S simple-scan  # GUI frontend
# Find your scanner:
scanimage -L
# Test scan:
scanimage --device=<device> > scan.pnm
# Network scanner: sudo pacman -S sane-airscan
```

Notifications & KDE Connect

```
sudo pacman -S kdeconnect
# Sync phone notifications, clipboard,
# files, battery — install KDE Connect app
# on your Android/iPhone

# KDE notification center:
# System Settings → Notifications
```

Night Light / f.lux alternative

```
# KDE built-in Night Color:
# System Settings → Display → Night Color
# Enable, set 3200K at night

# Or use redshift:
sudo pacman -S redshift
redshift -l 52.2:21.0 -t 6500:3200
# (Warsaw lat:lon)
```

QUICK REFERENCE · CHEAT SHEET

Essential Commands — Keep This Handy

Full update:	`sudo pacman -Syu && yay -Syu`
Install pkg:	`sudo pacman -S <pkg>` / `yay -S <pkg>`
Search pkg:	`pacman -Ss <term>` / `yay -Ss <term>`
Remove + deps:	`sudo pacman -Rns <pkg>`
Remove orphans:	`sudo pacman -Rns $(pacman -Qdtq)`
List installed:	`pacman -Q` / `pacman -Qe (explicit)`
File owner:	`pacman -Qo /path/to/file`
Pkg info:	`pacman -Qi <pkg>`

Enable service:	`sudo systemctl enable --now <svc>`
Service status:	`systemctl status <svc>`
Live logs:	`journalctl -f`
Boot errors:	`journalctl -b -p err`
Disk usage:	`df -h` / `du -sh *` / `ncdu`
Check RAM:	`free -h` / `vmstat 1`
CPU info:	`lscpu` / `cat /proc/cpuinfo`
GPU info:	`glxinfo | grep renderer` / `vainfo`

You're Done.
Now Break Things.

(And fix them — that's how you actually learn Arch.)

Arch Wiki →	wiki.archlinux.org — your bible
Arch Forums →	bbs.archlinux.org — community
Arch News →	archlinux.org/news — check before updates!
r/archlinux →	reddit.com/r/archlinux

BONUS · AUTOMATION & CRON

Cron Automation — Set It and Forget It

Weekly Full System Backup (Restic)

```
0 3 * * 0   # Every Sunday at 3:00 AM
```

```
#!/bin/bash
export RESTIC_PASSWORD_FILE=/root/.restic-pass
restic -r /mnt/backup/system backup /home /etc /root \
  --exclude=/home/*/.cache \
  --exclude=/home/*/.local/share/Trash \
  --tag weekly-auto
restic -r /mnt/backup/system forget \
  --keep-weekly 4 --keep-monthly 6 --prune
restic -r /mnt/backup/system check
```

Daily pacman Cache Cleanup

```
30 4 * * *  # Every day at 4:30 AM
```

```
#!/bin/bash
# Keep last 2 versions of each package
/usr/bin/paccache -rk2

# Remove all uninstalled packages
/usr/bin/paccache -ruk0

# Log results
echo "$(date): cache cleaned" >> /var/log/paccache.log
```

BTRFS Snapshot (before updates)

```
# Handled by snap-pac hook automatically!
# But add a manual weekly snapshot:
```

```
#!/bin/bash
# /etc/cron.weekly/btrfs-snapshot
SNAP_DIR="/.snapshots"
DESC="weekly-auto-$(date +%Y%m%d)"

snapper -c root create \
  --description "$DESC" \
  --cleanup-algorithm number

# Keep only last 5 weekly snapshots
snapper -c root cleanup number
```

Mirror Update (monthly)

```
0 5 1 * *   # 1st of every month at 5 AM
```

```
#!/bin/bash
reflector --country Poland,Germany,Czech \
  --sort rate --latest 20 \
  --save /etc/pacman.d/mirrorlist
echo "$(date): mirrors updated" \
  >> /var/log/reflector.log
```

BONUS · AUTOMATION & CRON

How to Set Up Cron — and systemd Timers

cronie — Cron Daemon Setup

```
# Install and enable cron daemon
sudo pacman -S cronie
sudo systemctl enable --now cronie

# Edit your crontab (user-level):
crontab -e

# Edit root crontab:
sudo crontab -e

# Or drop scripts into system dirs:
/etc/cron.hourly/     # every hour
/etc/cron.daily/      # daily ~midnight
/etc/cron.weekly/     # every week
/etc/cron.monthly/    # every month

# Cron syntax quick ref:
# MIN HOUR DOM MON DOW COMMAND
#  0   3    *   *   0  /scripts/backup.sh
#          ^--- 3AM every Sunday (DOW=0)

# Always test before adding to cron:
bash -x /etc/cron.weekly/my-script.sh
```

systemd Timers — Modern Alternative

```
# /etc/systemd/system/mybackup.service
[Unit]
Description=My weekly backup

[Service]
Type=oneshot
ExecStart=/usr/local/bin/backup.sh

# /etc/systemd/system/mybackup.timer
[Unit]
Description=Weekly backup timer
[Timer]
OnCalendar=Sun 03:00
Persistent=true
[Install]
WantedBy=timers.target

# Enable the timer:
sudo systemctl enable --now mybackup.timer
# List all active timers:
systemctl list-timers
```

More Useful Automations

- **Orphan cleanup:** `pacman -Rns $(pacman -Qdtq) --noconfirm`
- **Journal vacuum:** `journalctl --vacuum-size=500M`
- **TRIM SSD:** `systemctl enable fstrim.timer  # built-in!`

✓ fstrim.timer jest wbudowany w systemd — wystarczy go włączyć. Wykonuje TRIM na SSD raz w tygodniu.