ARCH LINUX + UEFI + SECURE BOOT + LVM + INTEL + GNOME

IMPORTANT: MAKE SURE THAT SECURE BOOT WITH NO KEYS ENROLLED IS SET

1. NETWORK & CONNECTIVITY BY WI-FI

```
ip link
iwctl
device list
station wlan0 scan
station wlan0 get-networks
station wlan0 connect “Network ID”
exit
ping 1.1.1.1
timedatectl

ls /usr/share/kbd/consolefonts/ | grep ter-120b
if yes: setfont ter-120b
if not: pacman -S kbd terminus-font
then  : setfont ter-120b
```

2. UEFI & KEYBOARD & LOCALE

```
ls /sys/firmware/efi/efivars && efibootmgr

loadkeys br-abnt2
nano /etc/locale.gen # unmark LANG=pt_BR.UTF-8
locale-gen
cat /etc/locale.conf
echo 'LANG=pt_BR.UTF-8' > /etc/locale.conf
```

3. PARTITIONING & FORMATTING & MOUNTING NVME LVM

```
parted /dev/nvme0n1 mklabel gpt
parted /dev/nvme0n1 mkpart ESP fat32 1MiB 1025MiB
parted /dev/nvme0n1 set 1 esp on
parted /dev/nvme0n1 mkpart CRYPTO 1025MiB 100%

mkfs.fat -F32 /dev/nvme0n1p1

cryptsetup luksFormat --type luks2 /dev/nvme0n1p2
cryptsetup open --allow-discards /dev/nvme0n1p2 cryptroot

pvcreate /dev/mapper/cryptroot
vgcreate vg0 /dev/mapper/cryptroot

lvcreate -L 3G -n boot vg0

mkfs.ext4 /dev/vg0/boot

lvcreate -l 100%FREE -n root vg0
```

```
mkfs.btrfs -f -L ROOT /dev/vg0/root

mount /dev/vg0/root /mnt

btrfs subvolume create /mnt/@
btrfs subvolume create /mnt/@home
btrfs subvolume create /mnt/@log
btrfs subvolume create /mnt/@pkg
btrfs subvolume create /mnt/@snapshots

umount /mnt

mount -o subvol=@ /dev/vg0/root /mnt
mkdir -p /mnt/{home,var/log,var/cache/pacman/pkg,var/snapshots}

mount -o subvol=@home /dev/vg0/root /mnt/home
mount -o subvol=@log /dev/vg0/root /mnt/var/log
mount -o subvol=@pkg /dev/vg0/root /mnt/var/cache/pacman/pkg
mount -o subvol=@snapshots /dev/vg0/root /mnt/var/snapshots

mkdir -p /mnt/boot
mount /dev/vg0/boot /mnt/boot

mkdir -p /mnt/boot/efi
mount /dev/nvme0n1p1 /mnt/boot/efi

4. BASE SYSTEM INSTALLATION FOR INTEL CHIPSET

reflector --country BR,CA,ES,NO,PT,US --protocol https --latest 15 --score 10 --
delay 1 --sort rate --save /etc/pacman.d/mirrorlist

pacstrap -K /mnt base intel-ucode linux-firmware linux-lts linux-lts-headers
vulkan-intel vulkan-tools sudo sbctl networkmanager nano mkinitcpio-utils
mkinitcpio lvm2 efibootmgr dkms cryptsetup btrfs-progs apparmor

genfstab -U /mnt > /mnt/etc/fstab

5. CHROOT CONFIGURATION

arch-chroot /mnt

ln -sf /usr/share/zoneinfo/America/Araguaina /etc/localtime

hwclock --systohc

nano /etc/locale.gen # unmark pt_BR.UTF-8 UTF-8

locale-gen

echo 'LANG=pt_BR.UTF-8' > /etc/locale.conf
echo 'KEYMAP=br-abnt2' > /etc/vconsole.conf
echo 'sofos' > /etc/hostname
```

```
passwd # root
useradd -m -g users -G wheel -s /bin/bash archer
passwd archer
EDITOR=nano visudo
%wheel ALL=(ALL:ALL) ALL # uncomment for enable sudo for “archer” user

6. BOOTLOADER

a. bootctl config

bootctl install

nano /etc/mkinitcpio.conf # make sure that’s setup is according to

MODULES=(i915) # insert the appropriate chipset driver
BINARIES=()
FILES=()

HOOKS=(base systemd autodetect microcode modconf kms keyboard sd-vconsole sd-
encrypt lvm2 filesystems fsck)

COMPRESSION="zstd"
COMPRESSION_OPTIONS=(-3)

nano /etc/mkinitcpio.d/linux-lts.preset # verify if the preset is according to

PRESETS=('default' 'fallback')

default_image="/boot/initramfs-linux-lts.img"
fallback_image="/boot/initramfs-linux-lts-fallback.img"
fallback_options="-S autodetect"

mkinitcpio -p linux-lts

blkid /dev/nvme0n1p2 # catch PARTUUID code

nano /boot/loader/entries/arch-lts.conf

title   Arch Linux LTS
linux   /vmlinuz-linux-lts
initrd  /intel-ucode.img
initrd  /initramfs-linux-lts.img
options rd.luks.name=<PARTUUID>=cryptroot rd.lvm.vg=vg0 root=/dev/vg0/root
rootfstype=btrfs rootflags=subvol=@ rw zswap.enabled=0 apparmor=1 security=apparmor
nvme_core.default_ps_max_latency_us=0 loglevel=3

cp /boot/loader/entries/arch-lts.conf /boot/loader/entries/arch-lts-fallback.conf

nano /boot/loader/entries/arch-lts-fallback.conf # only insert fallback name

title   Arch Linux Fallback
linux   /vmlinuz-linux-lts
initrd  /intel-ucode.img
```

```
initrd  /initramfs-linux-lts-fallback.img
options rd.luks.name=<PARTUUID>=cryptroot rd.lvm.vg=vg0 root=/dev/vg0/root
rootfstype=btrfs rootflags=subvol=@ rw zswap.enabled=0 apparmor=1 security=apparmor
nvme_core.default_ps_max_latency_us=0 loglevel=3

nano /boot/loader/loader.conf

default arch-lts.conf
timeout 2
console-mode max
editor no

b. sbctl config

sbctl status
sbctl create-keys
sbctl verify

# sign the EFI’s and Kernel’s according to below example

sbctl sign -s /boot/EFI/BOOT/BOOTX64.EFI
sbctl sign -s /boot/EFI/systemd/systemd-bootx64.efi
sbctl sign -s /boot/vmlinuz-linux-lts

sbctl verify
sbctl enroll-keys
sbctl list-enrolled-keys
sbctl status

c. hooks config

nano /etc/sysctl.d/99-sysctl.conf

kernel.kptr_restrict = 2
kernel.dmesg_restrict = 1
kernel.randomize_va_space = 2
fs.protected_hardlinks = 1
fs.protected_symlinks = 1

sysctl --system

mkdir -p /etc/pacman.d/hooks/

nano /etc/pacman.d/hooks/99-secureboot.hook

[Trigger]
Operation = Upgrade
Type = Package
Target = systemd
Target = linux-lts

[Action]
Description = Signing EFI binaries and kernel for Secure Boot
```

```
When = PostTransaction
Exec = /bin/sh -c '(sbctl sign -s /boot/EFI/BOOT/BOOTX64.EFI 2>/dev/null || true)
&& (sbctl sign -s /boot/EFI/systemd/systemd-bootx64.efi 2>/dev/null || true) &&
(sbctl sign -s /boot/vmlinuz-linux-lts 2>/dev/null || true)'

d. mkinitcpio initramfs generate

ls /boot | grep lts

initramfs-linux-lts-fallback.img
initramfs-linux-lts.img
vmlinuz-linux-lts

ls /sys/firmware/efi/efivars && efibootmgr

lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT

btrfs subvolume list /

systemctl enable apparmor.service
systemctl enable NetworkManager.service

mkinitcpio -P # to check about any error

exit # arch-chroot environment logoff

swapoff -a

umount -R /mnt

reboot + F2 + Secure Boot + Key Management # verify keys or import keys if you did
not has used sbctl config # + F10

7. POST-INSTALLATION CONFIGURATION

login as “root”

nmcli general status
nmcli device status
nmcli device wifi list
nmcli device wifi connect "SSID" --ask

pacman -S kbd terminus-font
setfont ter-120b

a. gnome desktop install

pacman -S adwaita-icon-theme bluez bluez-libs bluez-obex bluez-utils colord evince
eog gdm gnome-backgrounds gnome-boxes gnome-calculator gnome-calendar gnome-
characters gnome-clocks gnome-color-manager gnome-console gnome-control-center
gnome-disk-utility gnome-firmware gnome-info-collect gnome-keyring gnome-logs
gnome-menus gnome-music gnome-online-accounts gnome-power-manager gnome-session
gnome-settings-daemon gnome-shell gnome-system-monitor gnome-terminal gnome-text-
```

```
editor gnome-themes-extra gnome-tweaks gnome-usage gnome-user-docs gnome-user-share
gnome-weather grilo-plugins gst-plugins-base gst-plugins-base-libs gvfs gvfs-afc
gvfs-dnssd gvfs-goa gvfs-google gvfs-gphoto2 gvfs-mtp gvfs-nfs gvfs-onedrive gvfs-
smb gvfs-wsdd localsearch loupe man-db man-pages mutter nautilus network-manager-
applet orca rygel sushi system-config-printer tecla tinysparql wayland-utils xdg-
desktop-portal-gnome xdg-user-dirs-gtk xkeyboard-config yelp yelp-tools yelp-xsl

systemctl enable gdm.service
systemctl status gdm.service

reboot and login as created “user”

sudo systemctl start gdm.service # necessary only if login was failed

b. complementary applications and services to gnome # copy & paste

sudo pacman -S collision decibels extension-manager eyedropper file-roller firefox
firefox-i18n-pt-br foliate font-manager fragments gimp gimp-help-pt_br gnome-
browser-connector gnome-shell-extension-appindicator gnome-shell-extension-arc-menu
gnome-shell-extension-caffeine gnome-shell-extension-dash-to-panel gnome-shell-
extension-vitals gnome-shell-extension-weather-oclock gthumb libreoffice-still
libreoffice-still-pt-br mpv pdfarranger ptyxis seahorse shotwell showtime snapshot

sudo pacman -S alsa-utils apparmor aspell aspell-en aspell-pt at-spi2-core audio-
convert avahi bashtop bat bind-tools cronie cryptsetup cups cups-filters cups-pdf
curl dconf ethtool eza fail2ban fastfetch fd ffmpeg firewalld foomatic-db foomatic-
db-engine foomatic-db-ppds fwupd fzf git glances grc gutenprint hspell htop
hunspell hwinfo inxi iproute2 less libcamera libssh libssh2 libvncserver libvoikko
libwireplumber lsof meson mokutil ninja nmap nodejs npm nss-mdns ntfs-3g nuspell
p7zip pacman pacman-contrib parted pipewire pipewire-libcamera pipewire-pulse
powertop pwgen python-pyqt5 python-pyqt6 qt5-wayland qt6-wayland ripgrep rpcbind
smartmontools speech-dispatcher speedtest-cli tlp tree unzip upower v4l-utils
v4l2loopback-utils wget wireless_tools wireplumber zip zram-generator zsh zsh-
autocomplete zsh-autosuggestions zsh-completions zsh-history-substring-search zsh-
lovers zsh-syntax-highlighting

sudo pacman -Rns power-profiles-daemon # for notebook only

sudo systemctl mask systemd-rfkill.service systemd-rfkill.socket

sudo wget -O /usr/local/bin/yt-dlp
https://github.com/yt-dlp/yt-dlp/releases/latest/download/yt-dlp && sudo chmod a+rx
/usr/local/bin/yt-dlp

c. install fonts

sudo pacman -S inter-font ttf-hack ttf-fira-sans ttf-fira-mono ttf-ibm-plex noto-
fonts noto-fonts-emoji ttf-dejavu ttf-liberation ttf-carlito ttf-caladea terminus-
font ttf-material-icons ttf-material-symbols-variable ttf-meslo-nerd

sudo chown -R root:root /usr/share/fonts
sudo find /usr/share/fonts -type d -exec chmod 755 {} \;
sudo find /usr/share/fonts -type f -exec chmod 644 {} \;
sudo fc-cache -fv
```

d. services activation

```
sudo systemctl enable apparmor.service
sudo systemctl enable avahi-daemon.service
sudo systemctl enable bluetooth.service
sudo systemctl enable cronie.service
sudo systemctl enable cups.service
sudo systemctl enable firewalld.service
sudo systemctl enable fstrim.timer
sudo systemctl enable tlp.service

chsh -s /bin/zsh $USER

reboot

sh -c "$(wget
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh -O -)"

git clone --depth=1 https://github.com/romkatv/powerlevel10k.git "${ZSH_CUSTOM:-
$HOME/.oh-my-zsh/custom}/themes/powerlevel10k"

nano ~/.zshrc

ZSH_THEME="powerlevel10k/powerlevel10k"
```

e. system setup

1. fstab: sudo nano /etc/fstab # use findmnt -a for partitions sets
2. locale.gen locale.conf locale-gen
3. gnome-shell-extensions
4. gnome-tweaker
5. gnome-control-center
6. zram-generator