

# Install guide for my personal Arch setup

Load the correct keyboard layout

```
loadkeys [euro|de-latin1]
```

Check for internet connectivity and time settings. If they are not working correctly, refer to [the arch wiki](#)

```
ip link
ping wiki.kuchenmampfer.de
timedatectl
```

## Disk partitioning and formatting

Find the correct device (I will assume it is `/dev/sda` here) and partition it

```
fdisk -l
fdisk /dev/sda
```

Here, make two partitions:

1. The EFI-partition `/dev/sda1` to be mounted under `/boot` with the Size of 1GB
2. The Main partition `/dev/sda2` which contains all of our btrfs subvolumes

Format the EFI-partition as FAT32

```
mkfs.fat -F 32 /dev/sda1
```

Encrypt the main partition

```
cryptsetup -c aes-xts-plain -s 256 luksFormat /dev/sda2
cryptsetup luksOpen /dev/sda2 luksroot
```

Format the main partition as btrfs, mount it and create the subvolumes `root1`, `root2`, and `home`.

See [the arch wiki](#) for more details

```
mkfs.btrfs /dev/mapper/luksroot
mount -o noatime,ssd,compress=lzo /dev/mapper/luksroot /mnt
btrfs subvolume create /mnt/root1
btrfs subvolume create /mnt/root2
btrfs subvolume create /mnt/home
umount /mnt
```

Mount the EFI-Partition, one of the root subvolumes and the home subvolume

```
mount -o noatime,ssd,compress=lzo,subvol=root1 /dev/mapper/luksroot /mnt
mount --mkdir /dev/sda1 /mnt/boot
mount --mkdir -o noatime,ssd,compress=lzo,subvol=home /dev/mapper/luksroot /mnt/home
```

# Initialize the new system

Install an initial set of packages into the `/mnt` directory tree. `amd-ucode` only needs to be installed on one of the root partitions.

```
pacstrap -K /mnt base linux linux-firmware amd-ucode btrfs-progs networkmanager openssh neovim git gcc sudo
man-db man-pages texinfo os-prober
```

Generate `/etc/fstab`

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

Chroot into the new environment

```
arch-chroot /mnt
```

Uncomment the desired locales in `/etc/locale.gen`

```
...
#de_CH ISO-8859-1
de_DE.UTF-8 UTF-8
#de_DE ISO-8859-1
...
```

```
#en_SG ISO-8859-1
en_US.UTF-8 UTF-8
#en_US ISO-8859-1
...
```

Set the time zone, synchronize the clock and generate locales

```
ln -sf /usr/share/zoneinfo/Europe/Berlin /etc/localtime
hwclock --systohc
locale-gen
```

save the hostname, the locale and the keyboard layout to their config files

```
echo <hostname> > /etc/hostname
echo "LANG=de_DE.utf8" > /etc/locale.conf
echo "KEYMAP=euro" > /etc/vconsole.conf
```

# Configure the bootstrap process to correctly deal with encryption

Install grub

```
pacman -Syu grub grub-btrfs efibootmgr
```

Write the UUIDs of all devices to a temporary file in order to read them from inside nvim in the next step

```
blkid > /tmp/blocks
```

Edit Hooks in `/etc/mkinitcpio.conf`

```
...
HOOKS=(base udev autodetect microcode modconf kms keyboard keymap consolefont block encrypt
filesystems fsck)
...
```

Edit `/etc/default/grub`

...

```
GRUB_CMDLINE_LINUX="cryptdevice=UUID=<UUID of /dev/sda2>:luksroot root=UUID=<UUID of /dev/mapper/luksroot> lang=euro locale=de_DE.UTF8"
```

...

```
GRUB_DISABLE_OS_PROBER=true
```

On `root2`, mount `root1`, so `os-prober` can detect it

```
mount -o noatime,ssd,compress=lzo,subvol=root1 /dev/mapper/luksroot /mnt
```

Apply these configurations and install grub

```
mkinitcpio -p linux
```

```
grub-mkconfig -o /boot/grub/grub.cfg
```

```
grub-install --target=x86_64-efi --efi-directory=/boot --bootloader-id=grub
```

## Further setup

Copy the old `etc/pacman.conf` over

Install the repository for the `klassy` kde theme engine and install it, see [software.opensuse.org](https://software.opensuse.org)

```
key=$(curl -fsSL https://download.opensuse.org/repositories/home:paul4us/Arch/$(uname -m)/home_paul4us_Arch.key)
fingerprint=$(gpg --quiet --with-colons --import-options show-only --import --fingerprint <<< "${key}" | awk -F: '
$1 == "fpr" { print $10 }')
```

```
pacman-key --init
```

```
pacman-key --add - <<< "${key}"
```

```
pacman-key --lsign-key "${fingerprint}"
```

```
pacman -Sy home_paul4us_Arch/klassy
```

Install some more essential packages and enable their services

```
pacman -S plasma-meta sddm sddm-kcm fish fwupd konsole dolphin flatpak flatpak-kcm
```

```
systemctl enable NetworkManager
```

```
systemctl enable sddm
```

```
systemctl enable bluetooth
```

## Set root password

```
passwd
```

## Create my user

```
useradd -m -g users -G wheel -s /usr/bin/fish MYUSERNAME  
passwd MYUSERNAME
```

## Modify `/etc/sudoers`:

```
...  
%wheel ALL=(ALL:ALL) ALL  
...
```

## Enable sddm autologin in `/etc/sddm.conf.d/autologin.conf`

```
[Autologin]  
User=MYUSERNAME  
Session=plasma
```

# Finally

- Repeat all necessary steps with the `root2` subvolume
- Copy everything over into the `home` directory
- Install `ame` from the AUR

```
# as MYUSERNAME  
cd Skripte/ame  
git pull  
makepkg -si
```

- Install packages `ame ins $(cat packages.pacman)`
- Install flatpaks `flatpak install $(cat installed.flatpaks)`, but remember to first remove runtimes
- set up tailscale and Uni-VPN
- ensure that things work: Passwordmanager, SSH, Ansible, Applications, ...

---

Revision #10

Created 16 September 2024 10:59:27 by Kuchenmampfer

Updated 22 September 2024 10:36:25 by Kuchenmampfer