

Arch Linux

The guide is for both - server and desktop.

It includes:

Server & Desktop

- UEFI
- systemd-boot
- LVM on LUKS
- NetworkManager
- zram
- doas/sudo

Desktop

- Xorg
- KDE / Plasma
- SDDM/xinit

Download the ISO

<https://www.archlinux.org/download/>

Verify the ISO image

Check the two files in the same folder with the following command(s):

- for Arch users
 - `pacman-key -v archlinux-<version>-x86_64.iso.sig`
- other GnuPG systems
 - `gpg --keyserver pgp.mit.edu --keyserver-options auto-key-retrieve -verify archlinux-<version>-x86_64.iso.sig`
- and check the sha256sum with the following command
 - `sha256sum archlinux-<version>-x86_64.iso`

Another method to verify the authenticity of the signature is to ensure that the public key's fingerprint is identical to the key fingerprint of the [Arch Linux developer](#) who signed the ISO-file. See [Wikipedia:Public-key_cryptography](#) for more information on the public-key process to authenticate keys.

Initial setup

If using a US keyboard:

```
ls /usr/share/kbd/keymaps/**/*.map.gz
loadkeys us
```

Check if system is under UEFI:

```
ls /sys/firmware/efi/efivars
```

Connect to wifi if needed

```
iwctl
device list
station DEVICE_NAME scan
station DEVICE_NAME get-networks
station DEVICE_NAME connect SSID
```

Enable NTP and set timezone

```
timedatectl set-ntp true
timedatectl set-timezone Pacific/Auckland
```

Test Connection

```
ping techsaviours.org -c 1
```

Format disk/s and create partitions

Format your disks and create GPT table.

```
cfdisk /dev/sd*
```

Typical partitions look like this:

Partitions	Space	Type
/dev/sda1 (boot)	512M	EFI System
/dev/sda2 (root)	xG	Linux Filesystem (ext4,...)
/dev/sdb1 (home) (optional)	xG	Linux Filesystem (ext4,...)

As an option, the home partition - /dev/sdb1, if you want to use another hard drive

LVM on LUKS

Create LUKS

root

```
cryptsetup luksFormat --type luks2 --cipher aes-xts-plain64 --key-size 512
/dev/sda2
```

```
cryptsetup open /dev/sda2 root
```

home (Optional) Second disk (/dev/sdb1)

```
cryptsetup luksFormat --type luks2 --cipher aes-xts-plain64 --key-size 512  
/dev/sdb1  
cryptsetup open /dev/sdb1 home
```

Create LVM

Preparing the physical volumes, volume groups and logical volumes

root

```
pvcreate /dev/mapper/root  
vgcreate vg0 /dev/mapper/root  
lvcreate -l 100%FREE vg0 -n root
```

home (optional)

```
pvcreate /dev/mapper/home  
vgcreate vg1 /dev/mapper/home  
lvcreate -l 100%FREE vg1 -n home
```

Format filesystems and mount

root

```
mkfs.ext4 /dev/vg0/root  
mount /dev/vg0/root /mnt
```

boot

```
mkfs.fat -F32 /dev/sda1  
mkdir /mnt/boot  
mount /dev/sda1 /mnt/boot
```

home (optional)

```
mkfs.ext4 /dev/vg1/home  
mkdir /mnt/home  
mount /dev/vg1/home /mnt/home
```

Install the base packages

```
pacstrap /mnt base base-devel linux-hardened linux-hardened-docs linux-hardened-headers linux-firmware nano networkmanager lvm2 openssh
```

If you encounter some issues, e.g. if you are using an older ISO, first run `pacman -Sy archlinux-keyring && pacman-key --init && pacman-key --populate archlinux`.

Configure the system

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

Timezone

```
ln -sf /usr/share/zoneinfo/Pacific/Auckland /etc/localtime  
hwclock --systohc
```

Uncomment your location. For example: `en_US.UTF-8 UTF-8`

```
nano /etc/locale.gen
```

```
echo "LANG=en_US.UTF-8  
LC_ADDRESS=en_US.UTF-8  
LC_IDENTIFICATION=en_US.UTF-8  
LC_MEASUREMENT=en_US.UTF-8  
LC_MONETARY=en_US.UTF-8  
LC_NAME=en_US.UTF-8  
LC_NUMERIC=en_US.UTF-8  
LC_PAPER=en_US.UTF-8  
LC_TELEPHONE=en_US.UTF-8  
LC_TIME=en_US.UTF-8  
LC_ALL=en_US.UTF-8" >> /etc/locale.conf
```

```
locale-gen
```

Keyboard layout

```
ls /usr/share/kbd/keymaps/**/*.map.gz  
nano /etc/vconsole.conf
```

```
KEYMAP=YOUR_KEYBOARD
```

Hostname

```
echo "arch" > /etc/hostname
```

Host file

```
echo "127.0.0.1 localhost  
::1 localhost  
127.0.1.1 arch.localdomain arch" >> /etc/hosts
```

root password

```
passwd
```

Create an initial ramdisk

```
nano /etc/mkinitcpio.conf
```

```
HOOKS=(base udev autodetect microcode modconf kms keyboard keymap  
consolefont block filesystems fsck encrypt lvm2)
```

```
mkinitcpio -P
```

Bootloader

```
bootctl install
```

```
echo "title Arch Linux  
linux /vmlinuz-linux-hardened  
initrd /initramfs-linux-hardened.img  
options cryptdevice=UUID=$(blkid -s UUID -o value /dev/sda2):root  
root=/dev/vg0/root rw" >> /boot/loader/entries/arch.conf
```

```
echo "title Arch Linux (fallback initramfs)  
linux /vmlinuz-linux-hardened  
initrd /initramfs-linux-hardened-fallback.img  
options cryptdevice=UUID=$(blkid -s UUID -o value /dev/sda2):root  
root=/dev/vg0/root rw" >> /boot/loader/entries/arch-fallback.conf
```

Microcode

Depends on your CPU - [*AMD*](#) or [*Intel*](#) - choose one of the following commands:

```
pacman -S intel-ucode
```

```
pacman -S amd-ucode
```

doas

Allow members of group wheel to run commands:

```
echo "permit persist :wheel" >> /etc/doas.conf
chown -c root:root /etc/doas.conf
chmod -c 0400 /etc/doas.conf
```

The persist feature is disabled by default [...] This feature is new and potentially dangerous, in the original doas, a kernel API is used to set and clear timeouts. This API is openbsd specific and no similar API is available on other operating systems.

Sudo user?

```
pacman -Rsn opendoas
pacman -S sudo
```

Enable wheel for your sudo user.

```
visudo
```

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

or

```
echo "alias sudo='doas'"
alias sudoedit='doas rnano'" >> ~/.bashrc
ln -s $(which doas) /usr/bin/sudo
```

Add user

Change USER to your name.

```
useradd -m -G wheel -s /bin/bash USER
passwd USER
```

zram

Module

```
echo "zram" >> /etc/modules-load.d/zram.conf
```

Modprobe

```
echo "options zram num_devices=1" >> /etc/modprobe.d/zram.conf
```

Udev

```
echo 'KERNEL=="zram0", ATTR{disksize}="4GB" RUN="/usr/bin/mkswap  
/dev/zram0", TAG+="systemd"' >> /etc/udev/rules.d/99-zram.rules
```

Fstab

```
echo "# swap  
/dev/zram0 none swap defaults 0 0  
" >> /etc/fstab
```

Enable services

```
systemctl enable --now NetworkManager.service  
systemctl enable --now sshd.service
```

(Optional) Add key for home partition

If you have decided to use an additional partition or drive, you can also use a key instead of entering the passphrase over and over again. This way it only stays for root to enter the passphrase.

```
mkdir /etc/luks-keys/  
dd bs=512 count=4 if=/dev/urandom of=/etc/luks-keys/home.bin  
chmod -cR 0400 /etc/luks-keys/  
cryptsetup luksAddKey /dev/sdb1 /etc/luks-keys/home.bin  
echo "home /dev/sdb1 /etc/luks-  
keys/home.bin" >> /etc/crypttab
```

Reboot

```
exit
```

```
umount -R /mnt  
reboot
```

(Optional) Connect to wifi if needed

```
nmcli d wifi list  
nmcli dev wifi connect SSID password 'password'
```

Congratulation ☐ The server part is done! Continue with [KDE](#) if you want to install a desktop environment. Also create a [backup](#).

2021/11/27 20:24

From:

<http://wiki.techsaviours.org/> - Your Digital Privacy DIY Solutions | TECH
SAVIOURS .ORG

Permanent link:

http://wiki.techsaviours.org/en/desktop/operating_systems/arch_linux

Last update: **2022/10/24 08:24**

