

Installer Archlinux Dans un chroot

Table of Contents

- [Installer Archlinux Dans un chroot](#)
- [Installation et utilisation de pacman](#)
 - [Création de l'environnement chroot](#)
 - [Méthode A: Utilisation de l'image d'amorçage \(recommandé\)](#)
 - [Méthode B: Utilisation de l'image LiveCD](#)
 - [Utilisation de l'environnement chroot](#)
 - [Initialisation du trousseau de clés pacman](#)
 - [Sélection d'un miroir et téléchargement des outils de base](#)
 - [Choses à vérifier avant de redémarrer](#)
- [Dépannage](#)
 - [Cachedir](#)
 - [Hôte basé sur Debian](#)
 - [/dev/shm](#)
 - [/dev/pts](#)
 - [lvmetad](#)
 - [Hôte basé sur Fedora](#)

Cet article présente la construction d'un système Archlinux, pour faire en sorte que **pacman** travaille directement sur le système hôte ou d'exécuter un système Arch à l'intérieur du système hôte, l'installation réelle étant exécutée à partir du système Arch. Le système imbriqué est contenu dans un chroot.

Installation et utilisation de pacman

Pacman peut être compilé sur la plupart des distributions Linux et utilisé directement sur le système hôte pour amorcer Arch Linux. Les scripts d'arch-install doivent s'exécuter sans problème directement à partir des sources téléchargées de toute distribution récente.

Certaines distributions fournissent un paquet pour **pacman** et/ou **arch-install-scripts** dans leurs dépôts officiels, qui peuvent être utilisés à cette fin. Depuis février 2019, Gentoo est connu pour fournir le package **pacman**, tandis qu'**Alpine Linux** et **Fedora** fournissent des scripts **pacman** et **arch-install**.

Création de l'environnement chroot

Deux méthodes pour configurer et entrer le chroot sont présentées ci-dessous, du plus facile au plus compliqué. Sélectionner une seule des deux méthodes. Ensuite, passer au chapitre suivant.

Méthode A: Utilisation de l'image d'amorçage (recommandé)

Télécharger l'image d'amorçage d'un miroir dans /tmp.



on peut également télécharger la signature (même URL avec .sig ajouté) et la vérifier avec GnuPG.

Extraire l'archive:

```
tar xzf <path-to-bootstrap-image>/archlinux-bootstrap-*-x86_64.tar.gz
```

Sélectionner un serveur de référentiel en modifiant
/tmp/root.x86_64/etc/pacman.d/mirrorlist.

Créer le chroot

- Si bash 4 ou une version ultérieure est installée et que unshare supporte les options **-fork** et **-pid**:

```
/tmp/root.x86_64/bin/arch-chroot /tmp/root.x86_64/
```

- Sinon, exécuter les commandes suivantes:

```
mount --bind /tmp/root.x86_64 /tmp/root.x86_64
cd /tmp/root.x86_64
cp /etc/resolv.conf etc
mount -t proc /proc proc
mount --make-rslave --rbind /sys sys
mount --make-rslave --rbind /dev dev
mount --make-rslave --rbind /run run    # (assuming /run exists on the
system)
chroot /tmp/root.x86_64 /bin/bash
```

Méthode B: Utilisation de l'image LiveCD

Il est possible de monter l'image racine du dernier support d'installation d'Arch Linux, puis de chrooter dessus. Cette méthode présente l'avantage de fournir une installation Arch Linux fonctionnelle au sein du système hôte sans qu'il soit nécessaire de la préparer en installant des packages spécifiques.



Avant de continuer, il faut s'assurer que la dernière version de squashfs est installée sur le système hôte. Sinon, des erreurs telles que celles-ci sont à prévoir: **ERREUR FATAL abandonné: uncompress_inode_table: la lecture du bloc a échoué.**

L'image racine se trouve sur l'un des miroirs sous `arch/x86_64/`. Le format squashfs n'étant pas éditable, supprimer l'image racine et la monter.

- Pour effacer l'image racine, exécutez

```
unsquashfs airootfs.sfs
```

- Avant de pouvoir y accéder, configurer des points de montage et copier le `resolv.conf` pour la mise en réseau.

```
mount --bind squashfs-root squashfs-root
mount -t proc none squashfs-root/proc
mount -t sysfs none squashfs-root/sys
mount -o bind /dev squashfs-root/dev
mount -o bind /dev/pts squashfs-root/dev/pts ## important pour pacman (pour
la vérification de la signature)
cp -L /etc/resolv.conf squashfs-root/etc ## cela est nécessaire pour
utiliser la mise en réseau dans le chroot
```

- Maintenant, tout est prêt à être intégré dans le nouvel environnement Arch.

```
chroot squashfs-root bash
```

Utilisation de l'environnement chroot

L'environnement de démarrage est vraiment barebone (pas de nano, pas de ping, pas de cryptage, pas de lvm). Par conséquent, il faut configurer **pacman** afin de télécharger le reste de la base et, si nécessaire, base-devel.

Initialisation du trousseau de clés pacman

Avant de commencer l'installation, les clés pacman doivent être configurées:

```
pacman-key --init
pacman-key --populate archlinux
```

Conseil: L'installation et l'exécution du `hasged` doivent être effectuées sur le système hôte, car il n'est pas possible d'installer des packages avant d'initialiser **pacman keyring** et que **systemd** le détecte s'exécutant dans un chroot et ignore la demande d'activation. Pour générer suffisamment d'entropie sur un serveur sans tête distant exécuter plusieurs fois en boucle `ls -Ra /` dans une autre console (TTY, terminal, session SSH ...) (au moins cinq ou six exécutions à partir de l'hôte se sont suffisantes)

Sélection d'un miroir et téléchargement des outils de base

Après avoir sélectionné un miroir, actualiser les listes de paquets et installer les paquets souhaités: base, base-devel, partitioned, etc.



Comme il n'y a pas encore d'éditeur de texte, quitter arch-chroot et éditer la liste miroir à l'aide de l'éditeur de texte de l'hôte.

Choses à vérifier avant de redémarrer

Avant de redémarrer, se chrooter dans le système nouvellement installé.

Créer un utilisateur avec un mot de passe pour pouvoir se connecter via ssh. La connexion racine est désactivée par défaut depuis OpenSSH-7.1p2.

Définir un mot de passe root afin de pouvoir basculer sur root via su ultérieurement:

```
passwd
```

Installer ssh et l'activer pour qu'il démarre automatiquement au démarrage.

Configurer la connexion réseau pour qu'elle démarre automatiquement au démarrage.

Configurer un chargeur de démarrage et le configurer de manière à utiliser la partition de swap précédemment attribuée en tant que partition racine.

Dépannage

Cachedir

Si lors de l'installation des packages avec pacman, on a l'erreur suivante:

```
impossible de déterminer le point de montage de cachedir  
/var/cache/pacman/pkg
```

Pour résoudre ce problème, exécuter

```
mount --bind <répertoire livecd-ou-bootstrap> <répertoire livecd-ou-  
bootstrap>
```

avant de procéder au chroot.

Hôte basé sur Debian

/dev/shm

Sur certains systèmes hôtes basés sur Debian, pacstrap peut générer l'erreur suivante:

```
# pacstrap /mnt base

==> Creating install root at /mnt
mount: mount point /mnt/dev/shm is a symbolic link to nowhere
==> ERROR: failed to setup API filesystems in new root
```

En effet, dans certaines versions de Debian, /dev/shm pointe vers /run/shm alors que dans le chroot basé sur Arch, /run/shm n'existe pas et que le lien est rompu. Pour corriger cette erreur, créer un répertoire /run/shm:

```
mkdir /run/shm
```

/dev/pts

Lors de l'installation de archlinux-2015.07.01-x86_64 à partir d'un hôte Debian 7, l'erreur suivante a empêché pacstrap et arch-chroot de fonctionner:

```
# pacstrap -i /mnt

mount: mount point /mnt/dev/pts does not exist
==> ERROR: failed to setup chroot /mnt
```

Apparemment, c'est parce que ces deux scripts utilisent une fonction commune. chroot_setup () [1] s'appuie sur les nouvelles fonctionnalités d'util-linux, incompatibles avec Debian 7 (voir FS # 45737).

La solution pour pacstrap consiste à exécuter manuellement ses différentes tâches, mais en utilisant la procédure habituelle pour monter les systèmes de fichiers du noyau sur le répertoire cible ("newroot"):

```
newroot=/mnt
mkdir -m 0755 -p "$newroot"/var/{cache/pacman/pkg,lib/pacman,log}
"$newroot"/{dev,run,etc}
mkdir -m 1777 -p "$newroot"/tmp
mkdir -m 0555 -p "$newroot"/{sys,proc}
mount --bind "$newroot" "$newroot"
mount -t proc /proc "$newroot/proc"
```

```
mount --rbind /sys "$newroot/sys"
mount --rbind /run "$newroot/run"
mount --rbind /dev "$newroot/dev"
pacman -r "$newroot" --cachedir="$newroot/var/cache/pacman/pkg" -Sy base
base-devel ... ## ajoutez les paquetages que vous voulez
cp -a /etc/pacman.d/gnupg "$newroot/etc/pacman.d/"      ## copier le
trousseau de clés
cp -a /etc/pacman.d/mirrorlist "$newroot/etc/pacman.d/" ## copy mirrorlist
```

Au lieu d'utiliser arch-chroot, utiliser simplement chroot "\$newroot".

lvmetad

Si quand on veut créer des volumes logiques LVM à partir d'un environnement archlinux-bootstrap-2015.07.01-x86_64 sur un hôte Debian 7 cela génère l'erreur suivante:

```
# lvcreate -L 20G lvm -n root

/run/lvm/lvmetad.socket: connect failed: No such file or directory
WARNING: Failed to connect to lvmetad. Falling back to internal scanning.
/dev/lvm/root: not found: device not cleared
Aborting. Failed to wipe start of new LV.
```

(La création d'un volume physique et d'un groupe de volumes a fonctionné malgré /run/lvm/lvmetad.socket: la connexion a échoué: aucun fichier ou répertoire de ce type n'est affiché.)

Cela peut être facilement contourné en créant les volumes logiques en dehors du chroot (à partir de l'hôte Debian). Ils sont alors disponibles une fois de plus chrootés.

De plus, si le système que utilisé a lvm, on peut avoir le résultat suivant:

```
# grub-install --target=i386-pc --recheck /dev/main/archroot

Installing for i386-pc platform.
/run/lvm/lvmetad.socket: connect failed: No such file or directory
WARNING: Failed to connect to lvmetad. Falling back to internal scanning.
/run/lvm/lvmetad.socket: connect failed: No such file or directory
WARNING: Failed to connect to lvmetad. Falling back to internal scanning.
/run/lvm/lvmetad.socket: connect failed: No such file or directory
WARNING: Failed to connect to lvmetad. Falling back to internal scanning.
/run/lvm/lvmetad.socket: connect failed: No such file or directory
WARNING: Failed to connect to lvmetad. Falling back to internal scanning.
/run/lvm/lvmetad.socket: connect failed: No such file or directory
WARNING: Failed to connect to lvmetad. Falling back to internal scanning.
```

Debian n'utilise pas par défaut **lvmetad**. Il faut éditer /etc/lvm/lvm.conf et définir use_lvmetad sur 0:

```
use_lvmetad = 0
```

Cela déclenchera plus tard une erreur au démarrage dans la phase initrd. Par conséquent, il faut le appliquer les étapes sont les suivantes:

- Après avoir installé le système, vérifier **Mkinitcpio** et les paramètres du chargeur de démarrage.
- Il faudra peut-être modifier le fichier `/etc/mdadm.conf` afin de refléter les paramètres RAID (le cas échéant).
- Il faudra peut-être modifier vos CROCHETS et vos MODULES en fonction des exigences LVM et RAID: `MODULES = "dmmod" HOOKS = "base udev mdadmudev ...` bloquer les systèmes de fichiers `lvm2 ...`
- Il faudra probablement générer de nouvelles images **initrd** avec `mkinitcpio`.
- Définir `use_lvmetad = 0` dans `/etc/lvm/lvm.conf`.
- Mettre à jour les paramètres de chargeur de démarrage.
- Définir `use_lvmetad = 1` dans `/etc/lvm/lvm.conf`.

Hôte basé sur Fedora

Sur les hôtes basés sur Fedora et les clés USB actives, on peut rencontrer des problèmes lorsqu'on utilise **genfstab** pour générer **fstab**. Supprimer les entrées en double et l'option "seclabel" où elle apparaît, car elle est spécifique à **Fedora** et empêchera le système de démarrer normalement.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=debian:linux-chroot-archlinux>

Last update: **2025/02/19 10:59**

