

Installer Alpine Linux dans un chroot

Table of Contents

- [Installer Alpine Linux dans un chroot](#)
- [Prérequis](#)
- [Installation de ALpine](#)
 - [Mettre en place APK](#)
 - [Mettre en place le chroot](#)
- [Configuration du système](#)
 - [Entrer dans le chroot](#)
 - [Scripts d'initialisation](#)
 - [Configuration des droits root](#)
- [Pour aller plus loin](#)
 - [Installation de packages supplémentaires](#)
 - [Dépannage](#)

Cet article explique comment configurer un environnement de construction Alpine dans un chroot sous une distribution Linux différente, telle que Arch, Debian, Fedora, Gentoo ou Ubuntu. Une fois dans l'environnement chroot, vous pouvez créer, déboguer et exécuter des packages alpins. Ce guide peut également être utilisé pour installer Alpine Linux à partir d'un livecd non-Alpine Linux tel que Ubuntu ou un CD de secours Système.

Cet exemple d'installation d'Alpine Linux dans un chroot fonctionnera avec la dernière version. Mais il est également possible de créer un chroot avec les versions Edge ou plus anciennes d'Alpine Linux pour tester les backports.

Prérequis

Pour la base Alpine Linux, on n'a besoin que d'environ 6 Mo d'espace libre; bien que pour construire des paquets, on aura besoin d'au moins 500 Mo.

Les variables ci-dessous:

- **`${chroot_dir}`** = Doit pointer vers le répertoire chroot
- **`${mirror}`** = Doit être remplacé par l'un des miroirs Alpine Linux disponibles.

Installation de ALpine

Mettre en place APK

Conseil: Pour utiliser une installation 32 bits. remplacez **x86_64** par **x86** dans la commande ci-dessous.

Avertissement: On aura besoin de la version 2.6.22 ou ultérieure du noyau pour utiliser `apk-tools-static`

Télécharger le dernier paquet statique apk (remplacer `${version}` par la version actuelle):

```
wget ${mirror}/latest-stable/main/x86_64/apk-tools-static-${version}.apk
```

Les paquets **.apk** ne sont que des archives compressées, décompresser en utilisant:

```
tar -xzf apk-tools-static-*.apk
```

Installer l'installation de la base alpine sur le chroot

```
./sbin/apk.static -X ${miroir}/last-stable/main -U --allow-untrusted --root  
${chroot_dir} --initdb add alpine-base
```

Mettre en place le chroot

Configurer des appareils dans le chroot



La création manuelle de périphériques n'est pas nécessaire si on monte `/dev` des hôtes dans le chroot décrit plus tard.

```
mknod -m 666 ${chroot_dir}/dev/full c 1 7  
mknod -m 666 ${chroot_dir}/dev/ptmx c 5 2  
mknod -m 644 ${chroot_dir}/dev/random c 1 8  
mknod -m 644 ${chroot_dir}/dev/urandom c 1 9  
mknod -m 666 ${chroot_dir}/dev/zero c 1 5  
mknod -m 666 ${chroot_dir}/dev/tty c 5 0
```

Si on a besoin d'un accès au disque SCSI:

```
mknod -m 666 ${chroot_dir}/dev/sda b 8 0  
mknod -m 666 ${chroot_dir}/dev/sda1 b 8 1  
mknod -m 666 ${chroot_dir}/dev/sda2 b 8 2  
mknod -m 666 ${chroot_dir}/dev/sda3 b 8 3
```

```
mknod -m 666 ${chroot_dir}/dev/sda4 b 8 4
mknod -m 666 ${chroot_dir}/dev/sda5 b 8 5
mknod -m 666 ${chroot_dir}/dev/sda6 b 8 6
mknod -m 666 ${chroot_dir}/dev/sdb b 8 16
mknod -m 666 ${chroot_dir}/dev/sdb1 b 8 17
mknod -m 666 ${chroot_dir}/dev/sdb2 b 8 18
mknod -m 666 ${chroot_dir}/dev/sdb3 b 8 19
mknod -m 666 ${chroot_dir}/dev/sdb4 b 8 20
mknod -m 666 ${chroot_dir}/dev/sdb5 b 8 21
mknod -m 666 ${chroot_dir}/dev/sdb6 b 8 22
```

Un fichier **resolv.conf** est nécessaire pour la résolution de nom:

```
cp /etc/resolv.conf ${chroot_dir}/etc/
mkdir -p ${chroot_dir}/root
```

On peut également créer un nouveau **resolv.conf** en utilisant les serveurs OpenDNS (ou tout autre):

```
echo -e 'nameserver 208.67.222.222\nnameserver 2620:0:ccc::2' >
${chroot_dir}/etc/resolv.conf
```

Configurer le miroir APK (remplacer \${branche} par le dernier nom de branche stable, par exemple, v3.3):

```
mkdir -p ${chroot_dir}/etc/apk
echo "${mirror}/${branch}/main" > ${chroot_dir}/etc/apk/repositories
```

Configuration du système

Entrer dans le chroot

Important: À ce stade, Alpine a été installé avec succès sur le répertoire chroot. Mais avant de se chrooter, il faut monter /proc et /sys dans le chroot:

```
mount -t proc none ${chroot_dir}/proc
mount -o bind /sys ${chroot_dir}/sys
```

Si on ne veut pas créer vous-même des fichiers de périphérique spéciaux, on peut également monter le répertoire du périphérique hosts sur le chroot:

```
mount -o bind /dev ${chroot_dir}/dev
```

On peut maintenant chrooter:

```
chroot ${chroot_dir} /bin/sh -l
```

Scripts d'initialisation

Pour que le système soit réellement amorçable, il faut ajouter des scripts d'initialisation aux niveaux d'exécution appropriés:

```
rc-update add devfs sysinit
rc-update add dmesg sysinit
rc-update add mdev sysinit

rc-update add hwclock boot
rc-update add modules boot
rc-update add sysctl boot
rc-update add hostname boot
rc-update add bootmisc boot
rc-update add syslog boot

rc-update add mount-ro shutdown
rc-update add killprocs shutdown
rc-update add savecache shutdown
```

Configuration des droits root

Si vous utilisez Alpine en tant que système de construction natif, il faut s'assurer que chroot peut exécuter chmod. Ajouter ce qui suit dans `/etc/sysctl.conf`

```
kernel.grsecurity.chroot_deny_chmod = 0
```

Puis lancer la commande suivante

```
sysctl -p
```

Pour aller plus loin

Installation de packages supplémentaires

Alpine Linux possède un excellent méta-package pour la construction de packages Alpine à partir du

code source disponible, `alpine-sdk`. Pour installer, lancer:

```
apk add alpine-sdk
```

Dépannage

en cas d'avertissement du type **AVERTISSEMENT**: Ignorer `APKINDEX.xxxx.tar.gz`

Il faut s'assurer que `${chroot_dir}/etc/apk/repo` est valide à l'intérieur du processus `chroot`:

```
apk update
```

From:

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

Permanent link:

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

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

