

## Live-build, pour créer un système à façon

Georges Khaznadar

CLX

2024-09-15

- 1 Paquets à installer
- 2 Un système minimal
- 3 Que fait lb build ?
- 4 Le résultat
- 5 Personnalisation
- 6 Le répertoire auto/
- 7 À quoi ça sert ?

## Paquets à installer

```
apt install live-build
```

# La configuration

```
georgesk@libres:~/lb/base$ lb config
[2024-09-14 17:49:51] lb config
P: Creating config tree for a debian/bullseye/amd64 system
P: Symlinking hooks...
georgesk@libres:~/lb/base$ ls
auto  config  local
```

## Et allons-y ...

```
georgesk@libres:~/lb/base$ sudo lb build
[2024-09-14 17:59:04] lb build
P: live-build 20240810
P: Building for a debian/bullseye/amd64 system
[2024-09-14 17:59:04] lb bootstrap
...
```

## Que fait lb build ?

- ça crée un système Debian dans un sous-répertoire `cache/bootstrap`
- ça « *chroot* » dans `cache/bootstrap`, qui sert d'atelier
- là, rebelote : un nouveau système est créé, nommé `chroot`, pour fabriquer la cible
- les paquets, selon la *configuration*, sont intégrés au système-cible : téléchargés, installés, configurés.
- le système-cible est alors comprimé à l'aide de `mksquashfs` en une version miniaturisée, accessible en *lecture-seule*
- un environnement supplémentaire (`binary`) est créé, qui contiendra les outils pour booter le système-cible
- l'environnement `binary` et la miniature du système cible sont alors assemblés, puis mis en forme comme une image ISO-hybrid

Durée du tout : 14 minutes et 40 secondes, sur cette modeste machine.

## Le résultat

Le fichier bootable, sur DVD-ROM ou clé USB, c'est `live-image-amd64.hybrid.iso`

Ce fichier pèse 261 mégas, pour une distribution bookworm « de base ». Il suffit de le graver sur un DVD, ou de le copier vers un disque/une clé USB. Il contient ce qu'il faut pour *booter* avec ou sans UEFI.

## Contenu du répertoire, à la fin de lb build

```
georgesk@libres:~/lb/base$ ls
auto                config
binary             live-image-amd64.contents
binary.modified_timestamps live-image-amd64.files
cache              live-image-amd64.hybrid.iso
chroot             live-image-amd64.hybrid.iso.zsync
chroot.files       live-image-amd64.packages
chroot.packages.install local
chroot.packages.live
```

## Personnalisation

On peut ajouter des paquets, bien sûr, en plus des paquets du système minimal. Pour cela, il suffit de mentionner leur nom s'ils peuvent venir de l'archive Debian. S'il viennent d'ailleurs, il faut en conserver une copie

## Si on ajoute des paquets de l'archive Debian

On crée un fichier texte sous `config/package-lists/`, nommé par exemple `mes_paquets.list.chroot` : notez bien que l'extension doit bien être `.list.chroot` ; on peut créer plusieurs fichiers de cette sorte, ils seront tous pris en compte. Voici un exemple de contenu...

## Un fichier config/package-lists/mes\_paquets.list.chroot

... pour avoir les paquets nécessaires pour un bureau graphique léger XFCE

```
### ajout d'un bureau XFCE4, avec du français ###  
task-xfce-desktop  
task-french-desktop
```

## Paquets venus d'ailleurs

Ça peut être un paquet de la distribution *testing*, pas encore présent dans le dépôt par défaut, ou encore un paquet non-officiel, structuré en archive `.deb`

Les paquets seront déposés dans le répertoires `config/packages.chroot`

Enfin, s'il s'agit de logiciel *non empaqueté pour Debian*, par exemple un programme qui pourrait tourner avec l'émulateur `wine`, on dépose un arbre de fichiers complet, sous le répertoire `config/includes.chroot_before_packages` ou sous `config/includes.chroot_after_packages`

Puis on lance les commandes : `sudo lb clean` et `sudo lb build`.

Durée de construction, taille de l'image ISO : avec un petit serveur chez OVH (7,50 € par mois), on a un accès au réseau assez fluide : environ 7 MB/s pour les échanges avec les dépôts Debian. La construction y a duré 1 heure 50 minutes, pour une image de 980 mégas.

## Plus de personnalisation ; optimisations

lb build conserve des données en cache, dans un répertoire justement nommé cache/ ; il est possible de relancer lb config; sudo lb build après des modifications (par exemple ajout d'une liste de paquets), et si tout va bien le cache permet de gagner du temps dans quelques étapes.

Il est possible aussi de conserver dans un cache plus universel les paquets téléchargés, à l'aide d'un service comme apt-cacher. Ainsi, si on est obligé d'effacer le répertoire cache/, ou si on démarre un projet en parallèle, on économise des téléchargements.

La documentation disponible dans le [live-manual](#) détaille les nombreuses possibilités de personnalisation de live-build.

## Le répertoire auto/

Il est possible de regrouper les options qui gouvernent live-build dans un fichier `auto/config` :

- l'usage (ou non) d'un dépôts dit *de sécurité*
- la *distribution* à utiliser pour les paquets, les *sections* à utiliser
- des paramètres de *boot* à inclure pour GRUB
- etc.

## Contenu d'un fichier auto/config

```
#!/bin/sh
set -e
lb config noauto \
  --security false --distribution bookworm \
  --archive-areas "main contrib non-free-firmware" \
  --bootappend-live "persistence boot=live components \
    quiet splash locales=fr_FR.UTF-8 keyboard-layouts=fr \
    timezone=Europe/Paris" \
  --apt-indices "false" \
  --apt-http-proxy http://localhost:3142 \
  --apt-recommends "false" \
  --firmware-chroot "true" \
  --iso-application "Freeduc Jbart" \
  "${@}
```

## À quoi ça sert ?

J'ai fabriqué ainsi une **clé USB vive**, qui permet à des élèves d'utiliser un environnement riche durant les cours en lycée, et de pouvoir retrouver le même environnement chez soi.

Rien n'empêche d'imaginer d'autres usages :

- un média nomade pour créer une borne d'information (site web captif, exploré par Firefox en mode pleine page)
- des environnements prêts à l'usage, avec un logiciel qui peut être pointu à configurer : c'est le cas d'une **clé nomade nommée KNOWIMS**, avec le logiciel **Wims** (des dizaines de milliers d'exercices dynamiques, en math, sciences, langues, etc.)
- une clé vive nomade qui fournit des outils de diagnostic/réparation, pour intervenir sur une machine en panne par exemple
- une clé vive nomade venant avec une médiathèque thématique : tourisme autour d'une ville, **parcours possibles dans un musée**, etc.

## Des questions ?



Figure 1: QR-code pour les transparents