Live Build : création d'une image iso live personnalisée

- Objet : Apprendre à utiliser `live-build` afin de se faire une iso live aux petits oignons (sous Jessie).
- Niveau requis :
  avisé
- Suivi :
  à corriger, à placer
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.
- Topic de suivi sur le forum : https://debian-facile.org/viewtopic.php?id=14147

Introduction

`live-build` est un ensemble de scripts qui permet de créer une image iso live personnalisée de Debian. Il n'est pas forcément évident de prendre cet outil en main, d'où le présent tuto !

Les avantages :

1. Disposer de tous les logiciels désirés sur l'image iso
2. Reproduire son environnement de travail (bureau, icônes, thème...)
3. Réinstaller son système sans avoir à créer de sauvegarde (l'iso étant elle-même la sauvegarde !)
4. Porter son système vers une autre architecture : 32 ou 64 bits (chose que l'on ne peut pas faire avec une sauvegarde...)

Alléchant, n'est-ce pas ?

Installation et documentation

On installe le jouet et sa doc (tant qu'à faire) :

```
apt-get install live-manual live-build
```

Pour consulter la doc, visitez le répertoire `/usr/share/doc/live-manual/`, vous la trouverez dans les formats suivants :

- html
- odt
- pdf
- epub
- txt

Faites en sorte de l'avoir toujours sous la main en créant un ou plusieurs raccourcis sur votre bureau par exemple, c'est une vrai mine d'or !

Bon, allez, au boulot !

Exemple représentatif

Le but à atteindre : obtenir un image iso légère pour faire de la navigation internet, avec *votre* profil iceweasel/firefox

1. Copie des exemples des 3 scripts principaux

`live-build` fournit essentiellement 3 commandes principales :

1. `lb config` qui permet de définir les options de configuration globales de l'iso live que vous voulez créer.
2. `lb build` qui se charge des différentes étapes de construction de ladite iso.
3. `lb clean` qui nettoie le tout avant de passer à une nouvelle iso.

Pour simplifier les choses les mainteneurs du paquet nous fournissent des exemple tout prêts que l'on n'a plus qu'à copier et éditer en fonction de nos convenances.

On commence d'abord par créer un répertoire de travail dans lequel nous créons un premier répertoire de test où nous nous plaçons avec la commande cd :

```
cd ~
```

```
mkdir Live_Build_Work
```

```
cd Live_Build_Work
```

```
mkdir test_1
```

```
cd test_1
```

Maintenant qu'on est dedans, on va copier les 3 scripts dans un répertoire `auto` :

```
mkdir auto
```

```
cp /usr/share/doc/live-build/examples/auto/* auto/
```

2. Édition du script ''config''

On édite maintenant le script `config` de la manière suivante :

```
editor auto/config
```

`editor` lance votre éditeur de texte en mode console par défaut. À priori, si vous ne l'avez jamais défini manuellement, il s'agit de nano.

Pour en définir un autre :

```
update-alternatives --config editor
```

Vous pouvez bien sur mettre directement nano à la place de `editor`, ou bien un autre éditeur de texte que vous maîtrisez (vim, `emacs`...).

config

```
#!/bin/sh

lb config noauto \
    --architectures i386 \
    --linux-flavours "586 686-pae" \
    --linux-packages "linux-image" \
    --ignore-system-defaults \
    --bootappend-live "boot=live components autologin username=toto"
    "${@}"
```

Pour une image 64 bit, supprimez la ligne –`linux-flavours` et remplacez `i386` par `amd64`.

Cela nous donnera une iso live “passe-partout”, car en 32 bit avec deux noyaux :

1. un 586 pour les processeurs 32 bit simple coeur, et
2. un 686-pae pour les 32 bits double coeur qui fonctionnera également avec les processeurs 64 bits.

Le nom de l'utilisateur sera `toto` (vous pouvez éventuellement remplacer par autre chose si vous n'aimez pas ce prénom).

La langue du système sera l'anglais, et le clavier sera en `QWERTY`. Pour franciser tout ça, voir : les options de boot.

3. Définition des paquets supplémentaires

Afin d'installer les paquets de votre choix dans votre iso live il suffit d'éditer le fichier `config/packages-list/live.list.chroot` ainsi :

```
editor config/packages-list/live.list.chroot
```

Et d'ajouter à la fin du fichier :

live.list.chroot

```
linux-image-586
linux-image-686-pae
task-lxde-desktop
icewesael-l10n-fr
```

Lorsque vous serez décidé à faire une iso 64 bit, veillez à remplacer `linux-image-586` et `linux-image-686-pae` par `linux-image-amd64`.
Vous pouvez aussi les écrire les uns à la suite des autres, séparés par un espace.

Et si vous avez l'intention d'en mettre plusieurs, vous pouvez les répartir en plusieurs fichiers ayant comme extension : `.list.chroot`.

Par exemple : `bureautique.list.chroot`, `system.list.chroot`...

4. Ajout de fichiers de configuration pour l'utilisateur live

Les fichiers de configuration d'un utilisateur contenus dans son répertoire personnel sont les fichiers cachés (commençant par un point : `.`) ou bien se trouvant dans un répertoire caché (même chose). Les fichiers de configurations de votre profile iceweasel/firefox se trouvent donc dans le répertoire caché `.mozilla`.

Afin que l'utilisateur live (notre cher `toto`) en profite il va falloir créer un répertoire `/etc/skel` à l'intérieur du répertoire `config/includes.chroot` et copier notre `.mozilla` à l'intérieur.

```
mkdir -p /config/includes.chroot/etc/skel
```

et

```
cp -r ~/.mozilla config/includes.chroot/etc/skel/
```

5. Création de l'iso

Ça y est, on a plus qu'à lancer la procédure de création (ça prendra environ une heure) :

```
lb build
```

Alors, c'était dur ?

Plus de configurations

Maintenant qu'on a découvert l'utilisation basique de `live-build`, on va voir quelques possibilités supplémentaires.

Moi je veux pas Lxde ! Je veux le même bureau que sur mon ordi !!!

Oui oui, ok, c'est bon, ça arrive, pas la peine de s'énerver... Tout d'abord, on nettoie :

```
lb clean
```

```
lb config
```

Avant de relancer `lb config`, il est bon de jeter un oeil à `auto/config` pour voir si on à rien à changer...

1. Éditer la liste des paquets supplémentaires

Si vous utilisez une police, des icônes, un thème, ou encore des plugins particuliers, il faut penser à les ajouter au `live.list.chroot`.

Par exemple, avec Xfce :

live.list.chroot

```
task-french-desktop
task-xfce-desktop
faenza-icon-theme
fonts-cantarell
murrine-themes
xfce4-plugin-whiskermenu
```

Ajouter les applications de votre choix bien entendu.

2. Importer les fichiers de configuration

Certaines application ont leur propre `.bidule`. Cependant, un grand nombre d'entre elles, ainsi que la plupart des configurations relatives au bureau (y compris le fond d'écran) sont dans le répertoire `.config` (vous pouvez allez jeter un coup d'oeil dedans si vous ne me croyez pas).

```
cp -r ~/.config config/includes.chroot/etc/skel/
```

Terminé !

```
lb build
```

Et si je veux installer des paquets .deb supplémentaires ?

C'est d'un facilité déconcertante : il suffit de les placer dans `config/pacakges.chroot`. Pensez à vérifier tout de même que les dépendances seront satisfaites...

Et pour que ça installe la même chose que sur l'iso ?

C'est simple : dans le script `config` rajoutez la ligne :

```
    --debian-installer live \
```

Vous verrez, lors de l'installation, après le partitionnement, au lieu de la traditionnelle installation du

sytème de base puis du choix des logiciels, l'installateur copiera le contenu de l'iso.

C'est pas merveilleux ?!

Attention : lorsque vous serez amené à créer un nouvel utilisateur après avoir installé ce système, il prendra pour squelette le contenu du répertoire `/etc/skel/` ! Ce qui signifie que le nouvel utilisateur aura accès à votre profil iceweasel/firefox contenant tous vots mots de passe (à moins que vous ne les ayez protégé par un mot de passe principal). Même chose pour votre profil chromium si vous utilisez ce navigateur.

Prenez donc soin de faire le ménage dans le répertoire `/etc/skel` après avoir un installé votre système custom.

Je trouve ça lourd de renommer à chaque fois l'iso quand c'est fini

Aucun problème : dans le fichier `config/build`, remplacez `live-image` à la ligne 9 par : `ma-super-debian-que-j-ai-fais-moi-et-qui-dechire-tout`.

Les options de la commande config

Il y a beaucoup de chose à faire ici. Voyez plutôt :

```
man lb_config
```

On peut ajouter les options en ligne de commande, comme ceci (à chaque fois qu'on lance cette commande, les options se cumulent) :

```
lb config --une-option un_parametre
```

Ou bien tout mettre dans le fichier `auto/config` ; une option par une ligne, suivie d'un anti-slash ; comme ceci :

```
    --une-option un_parametre \
```

Parmi les options utiles :

- `–apt-source-archives false`
  N'inclue pas les dépôts source durant le bootstrap ou le chroot. Cela évite de gaspiller de la connexion pour rien, surtout qu'apt-get update est lancé plus d'une dizaine de fois pendant toute la procédure de `lb build` ! (Observer la sorite de votre terminal ou bien le fichier `build.log`, et vous verrez...)

- `–apt-recommends false`
  Permets de ne pas installer les paquets recommandés. Pratique lorsque l'on veut faire un install minimale, uniquement avec les logiciels désirés. Cela permet également de réduire la taille de l'image iso pour ceux qui voudraient la graver sur un CD pour une raison X ou Y (impossibilité de booter sur une clé USB ou un DVD par exemple).

- `–debian-installer true`

Utiliser l'installateur standard (celui qui ne copie pas l'iso).

- `–distribution stretch`
  Je pense que ça se passe de commentaire. Les plus téméraires d'entre vous pourraient (et je dis bien “pourraient”) être tentés de mettre `sid`...

- `–security true`, `–updates true` et `–backports true`
  Permets d'inclure les dépôts security, updates et backports durant la construction de l'iso. Pratique pour inclure un noyau plus récent.
  Ajoutez des paquets au fichier `live.list.chroot` pour en profiter, sinon, cela ne sert à rien !

- `–archive-areas main contrib non-free`
  Inclure les parties main, contrib et non-free des dépôts. Cela permet de mettre des firmwares non libre dans l'iso.
  Même remarque que précédemment...

Les options de boot

L'option `–bootappend-live` de la commande config à elle seule vous permets d'accéder à certains paramétrages très intéressants.
Voici quelques options utiles que vous pouvez ajouter à la suite dans votre fichier `config` (tout sur une ligne) :

- `lang=fr_FR.UTF-8 locales=fr_FR.UTF-8 keyboard-layouts=fr keyboard-model=pc105 timezone=Europe/Paris utc=yes`
  Définit la langue du système, l'agencement du clavier et le fuseau horaire.

- `hostname=ordi`
  Permet de définir le nom d'hôte (`ordi` en l'occurence).

- `toram`
  Charge la totalité du système dans la RAM. Cela permet de pouvoir faire fonctionner le système même si le support (CD, clé USB) est retiré. Pratique non ?

- `swap=true`
  Pour utiliser les partitions swap détectées.

Importation de fichiers et de répertoires

Il faut savoir que tout fichier que vous placez dans le répertoire `config/includes.chroot` se retrouvera à la racine du système de votre image iso. On peut donc importer, par exemple, un fichier de configuration global contenu dans `/etc`, comme `/etc/bash.bashrc`. Il faudra alors faire comme suit :

```
mkdir config/includes.chroot/etc
```

```
cp /etc/bash.bashrc /config/includes.chroot/etc/
```

Le répertoire `/etc/skel` - celui que l'on a peuplé plus haut - sert de “squelette” lors la création de

l'utilisateur live : tout ce qui a été déposé à cet emplacement se retrouvera dans son répertoire personnel. C'est pour cela qu'il fallait créer un répertoire `/etc/skel` à l'intérieur de `includes.chroot`. Il ne faut donc pas créer de répertoire `/home/toto` comme on pourrait se l'imaginer, mais utiliser la méthode décrite plus haut.

Les hooks

Les hooks sont des scripts qui s'effectuent à un moment précis d'une procédure dans un but de personnalisation.

Vos hooks doivent se trouver dans le répertoire `config/hooks` (il est déjà rempli avec un certain nombre de hooks par défaut après l'exécution de `lb config`.
Ils doivent :

1. commencer par un numéro, afin qu'ils soient exécuter dans un ordre précis (commencez par `1000-` puis `1001-`...) ;
2. porter un extension `.hook.chroot` ou `.hook.binary` selon l'étape à laquelle vous voulez qu'ils s'exécutent.

Liens

- Tuto pour Debian Wheezy : https://debian-facile.org/doc:install:live-build

Tutos sur d'autres sites (attention, certains ne fonctionnent plus sous jessie !) :

- http://linuxmao.org/tiki-index.php?page=live-build&structure=Accueil+Tutos&redirectpage=live-build
- https://yeuxdelibad.net/Logiciel-libre/Debian/Creer_sa_propre_distribution_avec_live-build.html
- http://blog.handylinux.org/article235/partage-d-experience-live-build-et-paquets-debian
- http://www.blaess.fr/christophe/2011/09/02/creation-dun-systeme-live-linux-personnalise/