Utilisation de PPA sous Ubuntu

Léo Baudouin

7 octobre 2013

Pour des ajouts, corrections, suggestions ou pour des questions vous pouvez me contacter à l'adresse : `leo.baudouin@univ-bpclermont.fr` ou au bureau **3116**.

1 Utilisation d'un PPA

Un PPA (Personal Package Archives) permet d'utiliser facilement les paquets Debian personnels compilés de façon automatique.

Exemple d'utilisation :

```
user:~$ sudo apt-add-repository ppa :<user>/<repository>
```

```
user:~$ sudo apt-get update
```

```
user:~$ sudo apt-get install <package>
```

2 Création d'un compte sur Launchpad

2.1 Création d'une clé PGP

Créez votre clé PGP (Pretty Good Privacy) en utilisant l'interface *Mots de passe et clés* d'Ubuntu. Faites *Fichier >Nouveau >Clé PGP* ou en ligne de commande : `user:~$ gpg --gen-key`

2.2 Partage de votre clé publique

Trois méthodes permettent de partager votre clé publique :

1. Dans *Mots de passe et clés* rendez-vous dans le menu *Distant >Synchroniser et publier des clés...*. Il se peut que cette méthode soit bloquée sur le réseau universitaire.
2. Rendez-vous sur un des sites suivants : `http://keyserver.ubuntu.com`, `http://keyserver.pgp.com` ou `http://pool.sks-keyservers.net` et entrez votre clé publique. Pour créer un fichier contenant votre clé publique vous devez faire :
   `user:~$ gpg --output mykey.asc --export -a <user>` en remplaçant *user* par l'identifiant de votre clé, ou une partie de votre nom. Envoyez le fichier **mykey.asc** sur le site de votre choix.
3. En console :
   `user:~$ gpg --send-keys --keyserver <server> <public-key>`

Vous devez attendre une dizaine de minutes pour que votre clé apparaisse sur les serveurs, passez donc à la suite.

2.3 Signature du code de conduite

Rendez-vous sur `http://launchpad.net` et créez un compte utilisateur. Sur votre page perso, cliquez sur l'icône  en dessous de la section **Signed Ubuntu Code of Conduct**.

1. Enregistrez votre clé PGP :
   - Récupérez l'empreinte de votre clé dans *Mots de passe et clés >Mes clés personnelles* puis double cliquez sur votre clé PGP et regardez les détails.
   - Collez la chaîne de 10 × 4 caractères.

– Vous allez recevoir un mail crypté avec votre clé publique, et vous allez utilisez votre clé privée pour le lire. Pour cela enregistrez le contenu du message dans un fichier, puis exécutez :

```
user:~$ gpg --decrypt myMail.txt
```

Vous pouvez également configurer Thunderbird, Evolution, KMail ou autre pour lire vos mails cryptés.

2. Téléchargez et lisez le code de conduite.
3. Signez le avec votre clé PGP.

```
user:~$ gpg --clearsign UbuntuCodeofConduct-2.0.txt
```

Si vous avez plusieurs clés, précisez l'utilisateur :

```
user:~$ gpg -u <user> --clearsign UbuntuCodeofConduct-2.0.txt
```

Copiez ensuite le contenu du fichier *UbuntuCodeofConduct-2.0.txt.asc* dans le champ correspondant sur le site de Launchpad.

3 Préparation

3.1 Organisation des dossiers

Créez un dossier <name>-<version> et placez y vos sources
Exemple :

```
user:~$ mkdir -p ~/packaging/libExample-0.1.0  && cd ~/packaging/libExample-0.1.0
```

L'architecture de votre dossier n'est pas important en soi, mais utiliser un modèle standard permet à d'autre de se repérer facilement dans votre projet.

```
libExample-0.1.0/
    include/
         myinclude.h
    src/
         mysrc.cpp
    CMakeLists.txt
```

Vous pouvez également utiliser d'autre méthode que CMake pour compiler (Makefile, configure,...)

3.2 Première version

Compressez votre dossier dans une tarball avec gunzip en respectant la notation suivante :
<name>_<version>.orig.tar.gz

```
user:~/packaging$ tar -zcvf libExample_0.1.0.orig.tar.gz libExample-0.1.0/
```

3.3 Configuration

Un dossier **debian** doit être créé dans votre dossier de travail afin d'y ajouter tous les fichiers permettant de configurer votre paquet.

```
user:~/packaging$ mkdir -p libExample-0.1.0/debian && cd libExample-0.1.0/debian
```

Dans ce dossier vous devez créer quatre fichiers :

1. Le fichier **changelog** va contenir les modifications apportées à votre projet ainsi que le numéro de version et la version d'Ubuntu pour laquelle votre projet est destinée.

```
libExemple (0.1.0) precise; urgency=low

  * Initial upload!

 -- Your Name <your.email@some.where>  Wed, 02 Oct 2013 14:32:28 +0200
```

Note : pour obtenir la date faites simplement : `user:~$ date -R`

2. Le fichier **control** va contenir les informations sur votre paquet

```
Source: libExample
Section: devel
Priority: optional
Maintainer: Your Name <your.email@some.where>
Build-Depends: cmake, build-essential
Homepage: http://www.yourhomepage.org

Package: libexample
Architecture: any
Depends:
Description: First test-package
 Long description of libExample
 It can span multiple lines!
```

3. Le fichier **copyright** informera sur le type de droits utilisés pour ce projet

```
###############################################################################
#                        libExample                                           #
###############################################################################

Copyright (C) 2013 Your Name

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

4. Le fichier **rules** donne les informations de compilation de votre projet

```
#!/usr/bin/make -f

BUILDDIR = build_dir

# secondly called by launchpad
build:
mkdir $(BUILDDIR);
cd $(BUILDDIR); cmake -DCMAKE_INSTALL_PREFIX=../debian/tmp/usr ..
make -C $(BUILDDIR)

# thirdly called by launchpad
binary: binary-indep binary-arch

binary-indep:
# nothing to be done

binary-arch:
cd $(BUILDDIR); cmake -P cmake_install.cmake
mkdir debian/tmp/DEBIAN
dpkg-gencontrol -plibExample
dpkg --build debian/tmp ..

# firstly called by launchpad
clean:
rm -f build
rm -rf $(BUILDDIR)

.PHONY: binary binary-arch binary-indep clean
```

Note : Vous devez installer les fichiers (exécutable, bibliothèque, ...) de votre projet dans le dossier **../debian/tmp/usr/** afin qu'ils apparaissent dans le fichier Debian, le code ci dessus permet d'utiliser la variable *CMAKE_INSTALL_PREFIX* dans le CMakeLists.txt.

4 Création des fichiers nécessaire à Launchpad

Pour le premier envoi, vous devez utiliser **debuild** avec les options suivante :

```
user:~/packaging/libExample-0.1.0$ debuild -S -sa -k0xxxxxxx
```

Remplacez les 'x' par l'empreinte de votre clé PGP, ce sont les 8 caractères après le '/' sur la ligne 'pub' en faisant :

```
user:~$ gpg --fingerprint
```

Pour les prochaines fois vous devrez utiliser les options *-S -sd* pour **debuild**.

5 Test de compilation

Afin de tester si Launchpad va réussir à compiler votre projet, vous pouvez faire une simulation de compilation en utilisant **pbuilder**.

```
user:~/packaging/$ sudo pbuilder build libExample_0.1.0.dsc
```

Si tout ce passe correctement vous aurez un fichier .deb dans le dossier `/var/cache/pbuilder/result`.

Contrôlez si ce fichier Debian correspond bien à vos attentes :

```
user:~$ gdebi-gtk /var/cache/pbuilder/result/libExample_0.1.0_amd64.deb
```

6 Envoi de votre projet

Launchpad a besoin du fichier *.changes* que **debuild** a précédemment créé.

```
user:~/packaging/$ dput ppa :<user>/<repository> libExample_0.1.0_source.changes
```

Il faudra plusieurs dizaines de minutes avant que Launchpad compile votre projet.

7 Mises à jour

Modifiez votre fichier **changelog** puis exécutez les commande **debuild** et **dput**.

```
user:~/packaging/libExample-0.1.0$ debuild -S -sd -k0xxxxxxx
```

```
user:~/packaging/$ dput ppa :<user>/<repository> libExample_0.1.0_source.changes
```

Informations obtenues sur : `http://www.simonschneegans.de/?p=346`

Document réalisé avec LaTeX