debmirror

Serveur de dépôt mirroir

Un serveur mirroir, est une copie exacte d'un serveur de dépôt. Il peut être utile d'en créer si vous n'avez pas accès à Internet de manière permanente, ou pour limiter la bande passante consommées par l'installation de logiciels depuis Internet. Debmirror est un script qui permet de télécharger et maintenir un mirroir Debian. Il utilise ftp, http, hftp ou rsync.

Debmirror effectue 3 étapes :

1. Téléchargement des paquets et des sources
2. nettoyage des fichiers inconnus
3. les paquets et sources sont scannés pour construire une liste.

OPTIONS

–mirrordir emplacement ou seront placé les paquets

–debug mode débugage

–progress, -p affiche une barre de progressions pendant le téléchargement

–verbose, -v mode verbeux

–source inclure les sources dans le miroir (défaut)

–nosource ne pas inclure les sources

–md5sums, -m utiliser md5sums pour déterminer si les fichiers sont correctes

–passive télécharge en mode passif

–host=remotehost, -h spécifie l'hôte distant (défaut : ftp.debian.org)

–user=remoteusername, -u Spécifie l'utilisateur distant, utile pour les proxys

–passwd=remoteuserpassword password de l'utilisateur

–method=ftp|hftp|http|rsync, -e Méthode pour le téléchargement des paquets

–proxy=http ://user :pass@url :port/ Serveur proxy a utiliser

–timeout=seconds, -t timeout à utiliser pour les opérations réseau (défaut 300 sec)

–root=directory, -r directory Spécifier le dossier sur l'hôte distant défaut "/debian"

–dist=foo [,bar,...], -d foo spécifier la distribution (sarge, etch, lenny, sid)

–section=foo [,bar,...], -s foo Spécifier la section défaut : main,contrib,non-free,main/debian-installer

–arch=foo [,bar,...], -a foo Architecture, défaut : i386.

–postcleanup nettoie le miroir local après le mirroring et uniquement s'il n'y a pas eu d'erreur

–cleanup fait un nettoyage des fichiers et dossiers inconnus

–nocleanup ne nettoie pas le miroir local

–ignore=regex ne jamais télécharger de fichier dont les noms de fichiers correspondent à regex, peut être spécifié plusieurs fois

–exclude-deb-section=regex ne jamais télécharger de fichier dont la section correspond à regex [games, doc, oldlibs, science,...] peut être spécifié plusieurs fois

–limit-priority=regex limite le téléchargement dont la priorité correspond à regex [required, extra, optional, etc...]

–include=regex ne pas exclure les fichiers correspondant à regex peut être spécifié plusieurs fois

–skippackages ne pas re-télécharger les paquets et fichiers sources, utile si vous savez qu'ils sont à jour.

–getcontents télécharge les fichiers Contents.arch.gz

–max-batch=number télécharge jusqu'à max-batch nombre de fichiers et ignore le reste

–rsync-batch=number télécharge jusqu'à max-batch nombre de fichiers avec chaque appel rsync

–ignore-missing-release ne plante pas si un fichier Release manque.

–ignore-release-gpg ne plante pas sir Release.gpg est manquant

–dry-run mode simulation

–rsync-options=options spécifier les options rsync alternatives à utiliser

–ignore-small-errors n'indique pas les erreurs de type missing ou broken

–pdiff=use|mirror|none si Release contient des entrées pour pdiff, debmirror va tenter de mettre a jour les fichiers sources et les paquets mais ne les inclu pas dans le mirroir.

Debmirror utilise gpgv pour vérifier les Release et Release.pgp en utilisant par défaut /.gnupg/trustedkeys.gpg. Pour ajouter les bonnes clés, il faut les importer comme ceci : gpg –keyring /usr/share/keyrings/debian-archive-keyring.gpg –export | gpg –import ou télécharger les clé depuis le serveur : gpg –keyserver keyring.debian.org –recv-keys key ID cette key ID peut être trouvée dans le message d'erreur gpgv dans debmirror

Installer un dépôt local

Un serveur web doit déjà exister (ex : apache)
Création d'un utilisateur système dédié, dans le groupe du serveur web pour que ce dernier ait les accès
adduser –system –home /var/www/mirror –gid 33 –no-create-home mirror
puis on crée un répertoire pour stocker les fichiers
mkdir /var/www/mirror
chown mirror :www-data ./mirror
chmod u+rwx,g+rxs-w,o-rwx ./mirror
Installation des programmes nécessaires (inclure un serveur http ou ftp) :
aptitude install debmirror debian-keyring
les opérations suivante sont à effectuer sous l'utilisateur mirror :
su mirror
récupération des clé
gpg –keyring /usr/share/keyrings/debian-role-keys.gpg –export | gpg –import gpg –keyring /usr/share/keyrings/debian-archive-keyring.gpg –export | gpg –import gpg –no-default-keyring -a –keyring /usr/share/keyrings/debian-archive-keyring.gpg –export 55BE302B | gpg –no-default-keyring –keyring /.gnupg/trustedkeys.gpg –import -
récupération des paquets (ici on ne récupère que la section debian-security)
debmirror –verbose –arch=i386,amd64 –dist=lenny/updates –source –method=rsync –host=mirror.ens-lyon.fr –root= :debian-security –section=main,contrib,non-free,main/debian-installer -ignore-release-gpg –cleanup /var/www/mirror/debian-security/
pour les clients utiliser dans /etc/apt/sources.list
deb http ://server/mirror/debian-security lenny/updates main contrib non-free
deb-src http ://server/mirror/debian-security lenny/updates main contrib non-free
Exemple de fichier de lancement de debmirror

```
# ! /bin/sh
ARGS=" -progress \
-arch=i386,amd64 \
-dist=lenny/updates \
-source \
-method=rsync \
-host=mirror.ens-lyon.fr \
-root=:debian-security \
-section=main,contrib,non-free,main/debian-installer \
-ignore-release-gpg \
-cleanup \
/var/www/mirror/debian-security/"
debmirror $ARGS
```

Attention aux droits et faire un cron pour lancer périodiquement ce script.