

Gestionnaire paquets GNU/Linux

Généralites

Les logiciels

La question cruciale sur un système Linux est comment on gère l'ajout, la suppression et la mise à jour des logiciels du système. Sous Windows on télécharge le .exe et on clique sur suivant mais pas sous GNU/Linux.

Ce problème se pose toujours. On a trois choix pour installer un logiciel sous Linux.

1. Le **gestionnaire de paquets** de la distribution (entièrement automatique).
2. Les **paquets RPM ou DEB** obtenues directement sur des sites fournisseur de paquets (semi-automatique).
3. Compiler à partir des sources du logiciel (entièrement manuel).

Tableau récapitulatif avec la facilité noté de 1 à 5.

Type	Installation	Suppression	Mise à jour	Dernière version
Gestionnaire de paquetage	5	5	5	3
Paquet RPM ou deb	4	4	4	4
Compilation des sources	2	1	2	5

Définition

Un **paquetage** ou paquet une archive comprenant les fichiers informatiques, les informations et procédures nécessaires à l'installation d'un logiciel sur un système d'exploitation. Les paquets les plus courants sont rpm (Red Hat Package Manager) sous les distributions Red Hat et deb pour les distributions Debian.

Un **dépôt** est un espace disponible sur un serveur d'internet accessible par une URL http ou ftp et qui s'intègre au gestionnaire de paquetage afin d'offrir des paquetages supplémentaires. Il existe des dépôts officiellement supportés par la distribution et d'autres non-officiels. Les officiels présentent uniquement les paquets qui sont supportés.

Red Hat fournit un dépôt de paquets officiel nommé Red Hat Network qui est un service payant et qui offre un panel de logiciel qui sont complètement intégré à la distribution et énormément testés pour être en parfaite harmonie avec le système. De plus les logiciels fournis sont supportés par Red Hat.

Un **gestionnaire de paquets** est un outil automatisant le processus d'installation, désinstallation, mise-à-jour de logiciels installés sur un système informatique. Le terme est surtout utilisé pour les systèmes d'exploitation basés sur Unix, tels GNU/Linux. Ces derniers utilisent dans leur majorité un gestionnaire de paquets, souvent fourni en standard. Ils permettent de mettre à disposition simplement des milliers de paquetages lors d'une installation standard. Red Hat/ Fedora utilisent [yum](#),

Debian/Ubuntu utilisent [apt](#) et [apt-get](#) ou aptitude. Ce sont donc des outils d'installations automatiques de paquets rpm ou deb qui téléchargent et installent les logiciels demandés automatiquement.

Conseils

Privilégier dans l'ordre :

1. Le gestionnaire de paquets de la distribution
2. Ensuite les paquets (rpm ou deb) de sites fournisseurs que vous avez téléchargés manuellement
3. Et en dernier lieu, envisager la compilation à partir des sources.

En suivant ces conseils, vous éviterez de vous arracher les cheveux pour installer un logiciel sous Linux.

Convention de nommage des paquets rpm

La convention de nommage des packages rpm est la suivante : **nom-version-release.architecture.rpm**

- nom : représente le nom du package, et est en général le nom du logiciel

Celui-ci peut aussi être décomposé en plusieurs packages. on trouve couramment.

- nom-devel : package de "développement" (include C, bibliothèques) pour permettre la compilation de logiciels.
- libnom : les bibliothèques du logiciel sont séparées du package principal.
- nom-common : partie du logiciel utilisable pour un serveur ou un client.
- nom-server : partie du logiciel pour un serveur.
- nom-client : partie du logiciel pour un client.
- version : telle qu'elle est définie par le ou les développeurs du logiciel.
- release : elle est définie par celui qui fabrique le package.
- L'architecture représente la catégorie de processeur sur laquelle peut être installée le package.
 - ppc : powerpc (mac)
 - sparc : stations sun
 - alpha : pc avec processeurs alpha
 - ...
- i* : processeurs Intel.
 - i386 : la base commune (80386).
 - i486 : ?.
 - i566 : pentium.
 - i686 : pentium II, III, IV.

Tous ces packages contiennent des programmes compilés, donc non portables.

1. Ils ne sont donc pas portables entre les différents processeurs(Intel, powerpc, sparc)
2. Ils dépendent également des bibliothèques utilisées lors de la compilation, et sont donc souvent non

portables entre distributions différentes, voire sur la même distribution, entre versions différentes

Remarque : a priori, le nom de la distribution et la version n'apparaissent pas dans le nom du package (sauf Mandriva, qui finit ses releases par la chaîne "mdk")

il y aussi 2 types d'architecture portable.

- noarch : ces packages contiennent des fichiers de configuration, ou des programmes en langage interprétés (shell, perl, python ...), et peuvent donc être installés partout.
- src : ces "pseudo-packages" contiennent les programmes sources, avec ce qu'il faut pour les recompiler (Makefile, configure) et fabriquer un package.

Convention de nommage des paquets deb

Debian utilise une convention de nommage pour les fichiers de paquet : **nompaquet_version-révisionDebian_plateforme.deb**

- nompaquet : nom du paquet.
- version : version originale du logiciel du paquet.
- révisionDebian : numéro de révision du paquet. Ce numéro est donné par le mainteneur du paquet.
- plateforme : la plateforme cible du paquet (i386, amd64, alpha, etc). Si le paquet n'est pas dépendant d'une plateforme, il est noté all.

Quelques exemples.

- gnash_0.7.1-1_amd64.deb
- mozilla-firefox_1.0.4-2sarge7_i386.deb
- bind-doc_8.4.6-1_all.deb
- iptables_1.3.3-2ubuntu4_powerpc.deb

Gestionnaire de paquets Red Hat : Red Hat Network

Red Hat Network est un service payant de la société Red Hat. C'est le service qui permet de mettre à jour les serveurs Linux Red Hat à l'aide du dépôt officiel.

Enregistrement

Pour bénéficier du service, il faut disposer d'un compte et se connecter sur le site suivant.

<https://www.redhat.com/wapps/sso/rhn/login.html?redirect=http%3A%2F%2Frhn.redhat.com%2Frhn%2FYourRhn.do>

Ensuite, dans l'onglet Systèmes, on voit tous les systèmes qui sont enregistrés au service Red Hat Network.

L'enregistrement au service se réalise sur le serveur lui-même. Il faut bien sûr disposer de suffisamment d'accès achetés au préalable pour pouvoir ajouter un serveur supplémentaire. On a deux choix pour enregistrer :

- Soit on le fait à l'installation. Un menu nous est proposé pour renseigner les identifiants.
- Soit on le fait après l'installation (quand on veut) en se connectant une première fois au service à l'aide de l'icône en haut à droite du bureau GNOME. A ce moment, les informations (login+mot de passe) nous sont demandées et le serveur s'enregistre auprès de RHN. Il suffit de rafraichir la liste des serveurs sur le site Web pour remarquer qu'il est visible dans la liste des systèmes.

Suppression

Cet enregistrement peut être supprimé sur le site Web en cliquant sur le système en question, en cliquant sur le lien "**supprimer le système**" et enfin "**Delete profile**". Cette opération coupe les mises à jour du serveur. Ceci peut être intéressant lorsqu'on réalise un clone d'une machine sur une infrastructure virtualisée VMware ou autre (il faut refaire les enregistrements obligatoirement dans ce cas) ou lorsqu'on veut momentanément donner l'accès à un autre serveur lorsqu'on a pas d'emplacement de libre.

Réenregistrement

Lorsqu'on a désenregistré un serveur, il n'a plus accès aux mises à jour. Pour de nouveau l'ajouter au service et donc à la liste du site Web, il faut entrer la commande suivante en root.

```
rhn_register
```

Cette commande est lançable en commande ou en environnement graphique. Elle relance l'assistant qui demande le login et le mot de passe pour accéder au compte RHN.

Les kernels

Une particularité est appliquée pour le kernel Linux. Il est possible de la mettre à jour via l'utilitaire mais cette manipulation est à réaliser uniquement hors production. Il faut adopter la politique suivante :

- Lors du déploiement d'une machine, faire **toutes les mises à jour y compris celles des kernels (opération manuelle sur le serveur)**.
- Mettre à jour **tous les paquetages y compris les kernels** juste avant l'installation des applications métiers que le serveur hébergera (**opération manuelle sur le serveur**).
- **Ne plus mettre à jour les kernels** à partir du moment où le serveur est en production. (Attention, j'entends par production aussi les serveurs de tests) (**opération hebdomadaire à réaliser via l'outil Web**).

Les mises à jour kernels ne sont faisable que manuellement en se connectant sur le serveur (en graphique ou en commande). Toutes les autres mises à jour sont faisables via l'outil Web Red Hat Network.

Pourquoi cette particularité concernant les kernels ?

Les kernels sont des paquetages particuliers. Ils représentent la partie principale du système d'exploitation. Certains outils souvent externes comme des logiciels de gestion de multi-cheminement fibre channel ou des outils VMware comme les VMware Tools, n'acceptent pas les changements de kernels. Il faut donc les faire au début une bonne fois pour toute et ensuite appliquer ces logiciels et ne plus changer de versions. On peut aussi reconfigurer à chaque fois l'outil pour le nouveau kernel. Ce n'est pas une tâche hyper compliquée non plus, c'est un choix.

Solution

Red Hat connaît bien le problème et propose la solution de positionner le kernel comme un paquetage ignoré par défaut.

- Sous Red Hat 4, il faut configurer cela avec l'interface fournit par `up2date --configure`. On peut aussi ajouter les paquets qu'on veut ignorer dans le mot clé `pkgSkipList` du fichier `/etc/sysconfig/rhn/up2date`

```
pkgSkipList=kernel*;libgcc*
```

- Sous Red Hat 5, il faut configurer le fichier `/etc/yum.conf` et lui ajouter le mot clé suivant.

```
exclude=kernel*
```

Pour faire la mise à jour d'un kernel, il faut donc faire la mise à jour **manuellement** sur le serveur en cochant les paquetages kernel explicitement **“marqués pour être ignorés”**. Si la mise à jour est lancée depuis l'interface Web RHN et que vous cochez tous les paquetages y compris les kernels, sachez que **les kernels ne seront PAS INSTALLES puisque présent dans la liste des paquetages à ignorer. C'est très pratique. Ceci signifie que nous pouvons gérer la totalité des mises à jours de tous les serveurs Linux Red Hat avec une seule interface Web sans risque !**

Dans tout les cas ne **JAMAIS faire les mises à jour kernel sur des SERVEURS EN PRODUCTION**, on ne met à jour que **les autres paquetages**. Les mises à jour kernel doivent se faire au préalable d'une installation d'une application par une société ou autres et dans tout les cas juste après avoir déployé la machine Linux.

Si vous avez mis à jour le kernel, **pas de panique**. Les kernels Red Hat sont stable. Après avoir installé le nouveau kernel, ne pas rebooter la machine et aller dans le fichier `/boot/grub/menu.lst` pour changer la valeur `default` afin de booter par défaut sur le noyau qu'on utilisait jusqu'à présent.

Conseil

Lors de la création du système, je conseil de faire les mises à jour à la main comme ca on peut faire celle du kernel et pour la gestion régulière en production (au moins une fois par semaine) il est très intéressant d'utiliser l'outil Web.

Utilisation du gestionnaire de paquet Red Hat

En commande

Red Hat 4 (up2date)

Installer un paquet.

```
up2date -i nompaket
```

Lister si le paquet nompaket est disponible à l'installation dans le gestionnaire de paquet.

```
up2date -l nompaket
```

Les commandes qui suivent dépendent de la directive `pkgSkipList` présente dans `/etc/sysconfig/rhn/up2date`. Skip list par défaut dans le fichier.

```
pkgSkipList=kernel*;;
```

Mettre à jour.

```
up2date -u
```

La commande ne va pas mettre à jour les paquetages ajoutés dans la liste des paquetages à ignorer (donc les kernels ne s'installeront pas).

```
up2date -fu
```

La commande va tout installer y compris les paquetages de la liste des paquetages à ignorer (donc les kernels s'installeront).

Red Hat 5 ou 6 (yum)

Installer un paquet.

```
yum install nopaquet
```

Supprimer un paquet.

```
yum remove nopaquet
```

Lister si le paquet nopaquet est disponible à l'installation dans le gestionnaire de paquet.

```
yum list nopaquet
```

Liste des repositories disponibles

```
yum repolist
```

Rechercher un paquet disponible

```
yum search smb
```

```
yum info nmap
```

```
yum list
```

Usage des groupes de paquets

```
yum groups list  
yum groups info  
yum groups install "Server with GUI"  
yum groups install "Graphical Administration Tools"
```

Revenir à un état précédent par rapport à l'historique

```
yum history  
yum history undo <number>
```

Les commandes qui suivent dépendent des paramètres entrés dans `/etc/yum.conf`. Le fichier `/etc/yum.conf` a été modifié pour contenir la ligne `exclude=kernel*` ce qui fait qu'on ne peut installer et mettre à jour les kernels que si on spécifie sur la ligne de commande qu'on veut désactiver cette limitation. Cela évite des fausses manipulations. Si une exclusion est mise en place, il n'y a pas lieu de la spécifier sur la ligne de commande.

Mettre à jour le système sauf les kernels.

```
yum update --exclude=kernel*
```

Si pour une raison ou une autre, vous avez à positionné plusieurs paquets en exclusion pour une mise à jour, voici un exemple avec `openldap` et les kernels.

```
yum update --exclude=openldap,kernel*
```

Le séparateur est la virgule.

Pour installer toutes les mises à jour y compris les derniers kernel (sauf si exclusion dans `yum.conf`).

```
yum update
```

Mettre tout le système à jour sans tenir compte des éventuelles exclusions du fichier `/etc/yum.conf` (intègre donc les kernels).

```
yum update --disableexcludes=all
```

Pour lister les mises à jour de sécurité disponibles en retournant un code d'erreur exploitable dans un script si des mises à jour sont disponibles.

```
yum --security check-update
```

Appliquer les mises à jour de sécurité en désactivant celles des kernels.

```
yum --exclude=kernel* --security update
```

Avec les kernels.

```
yum --security update
```

Mise à jour de sécurité automatique avec confirmation automatique (idéal pour un script).

```
yum --exclude=kernel* --security -y update
```

Appliquer une mise à jour relevant d'un bug particulier declare dans le bugzilla. Exemple ici avec les bugs 123, 456 et 789.

```
yum --bz 123 --bz 456 --bz 789 --security update
```

Recherche un paquet qui fournit une commande particulière (ici la commande `spawn`).

```
yum whatprovides */spawn
```

Recherche un paquet qui fournit une commande particulière (ici la commande `ftptop`).

```
yum provides ftptop
```

Produire la liste des dépendances et quel paquet fournit ces dépendances pour le paquet spécifié en paramètre.

```
yum deplist expect
```

En graphique sous GNOME

Il faut lancer l'outil de mise à jour et suivre les instructions. C'est très simple, il n'y a rien de particulier.

Mise à jour via le site Web RHN

Ce site Web permet de mettre à jour les systèmes enregistrés sur Red Hat Network (RHN). On a accès à la liste de tous les serveurs présents, ceux qui ont besoin de mise à jour et ceux qui sont déjà à jour. Pour faire les mises à jour réellement, il suffit de suivre les instructions et les mises à jour vont se programmer toutes seules.

Ce qu'il faut retenir c'est que lorsque sur tous les systèmes est mis en place une exclusion de paquet (ex : le kernel), celle-ci sera appliquée. Si on coche dans les mises à jour ce paquet exclus, il ne sera pas traité.

Dépôts RHEL supplémentaires

Le problème qu'on rencontre assez souvent est que le logiciel que nous voulons installer n'est pas présent dans le dépôt RHN lorsqu'on fait un `yum install`.

On a donc deux solutions pour pallier ce problème. Soit ajouter un nouveau dépôt compatible avec la Red Hat même s'il n'est pas officiel, soit installer les RPM à partir de paquets téléchargés (cf. partie suivante).

EPEL

Un dépôt nommé EPEL (Extra Package for Enterprise Linux) sponsorisé par Fedora (et donc Red Hat) est un dépôt qui a pour volonté d'ajouter un grand nombre de paquets de qualités de manière compatible à une Red Hat Enterprise Linux et même CentOS. L'installation du dépôt EPEL est très simple, il suffit d'installer le paquet.

```
rpm -Uvh epel-release-5-3.noarch.rpm
```

Ajouter sur le proxy de la structure les URL suivantes dans la liste blanche.

```
rpmfind
fedora-epel
epel
ircam.fr
```

Pour avoir les autorisations de passer le proxy sans authentification, il faut ajouter dans le groupe des mises à jour les URL qu'on considère comme étant sûres car EPEL se base sur plusieurs serveurs accessibles par plusieurs noms de domaines. Pour les mises à jour officielles, c'est beaucoup plus simple car il n'y a qu'une seule URL.

Le proxy doit correctement être configuré dans `/etc/yum.conf` pour que le nouveau dépôt fonctionne. La configuration présente dans le fichier `/etc/sysconfig/rhn/up2date` ne s'applique qu'au dépôt RHN.

Ensuite, il est possible lorsqu'on fait un `yum install nompaket` d'installer beaucoup plus de

paquets.

Par contre, il est très important d'ajouter ce dépôt dès l'installation du système afin d'éviter de casser complètement le gestionnaire de paquets avec des paquets identiques installés à la main. Il faut tout de même être prudent, un système de paquet cassé peut devenir dramatique.

Connaître si un paquet installé provient du dépôt EPEL.

```
rpm -qp foo-0.1-5.el5.i386.rpm --qf '%{NAME}-%{VERSION}-%{RELEASE}
%{VENDOR}\n'
foo-0.1-5.el5 Fedora Project
```

```
rpm -qp foo-0.1-5.el5.i386.rpm --qf '%{NAME}-%{VERSION}-%{RELEASE}
%{DISTRIBUTION}\n'
foo-0.1-5.el5 Extras Packages for Enterprise Linux
```

```
rpm -qp foo-0.1-5.el5.i386.rpm --qf '%{NAME}-%{VERSION}-%{RELEASE}
%{SIGGPG}\n'
foo-0.1-5.el5
883f030500468e3e4e119cc036217521f611025863009f5fe424c6fe4bc81a57f45722e465e7
1381dda2f6009f7c08e1743794b5b9a5a4cd149081092801a5d935
```

Server HA Clustering

Voici un depot créé par OpenSuse qui fournit les derniers paquets d'heartbeat. L'URL d'accès au dépôt est la suivante : http://download.opensuse.org/repositories/server:/ha-clustering/RHEL_5/

Pour installer le dépôt, il faut positionner le fichier

http://download.opensuse.org/repositories/server:/ha-clustering/RHEL_5/server:ha-clustering.repo

dans le dossier `/etc/yum.repos.d/` Ensuite, on peut installer classiquement avec un `yum install`.

Utilisation du gestionnaire de paquet Debian

En commande

`apt-get` est l'outil de base mais `aptitude` vient améliorer la gestion des dépendances. Les deux fonctionnent très bien par contre il est recommandé de choisir l'un ou l'autre mais de ne pas mélanger les deux utilisations. Aujourd'hui, il est plus intéressant d'utiliser `aptitude`.

```
aptitude clean apt-get clean      nettoyer complètement le dépôt local des
fichiers de paquets récupérés
aptitude autoclean apt-get autoclean  nettoyer le dépôt local des
fichiers des paquets périmés
aptitude show toto apt-cache show <paquet>  afficher des informations
```

```
détaillées concernant le paquet « toto »
aptitude search <expression rationnelle> apt-cache search <expression
rationnelle>      rechercher les paquets qui correspondent à l'<expression
rationnelle>
aptitude why <expression rationnelle>      N/A      expliquer les raisons qui
font que les paquets correspondant à l'<expression rationnelle> devront être
installés
aptitude why-not <expression rationnelle>      N/A      expliquer les raisons
pour lesquels les paquets qui correspondent à l'<expression rationnelle> ne
peuvent pas être installés
```

Mettre à jour les méta-données.

```
aptitude update
```

Installer un paquet.

```
aptitude install numpaquet
```

Supprimer un paquet en gardant les fichiers de configuration.

```
aptitude remove numpaquet
```

Supprimer un paquet complètement en ne gardant aucun fichier de configuration.

```
aptitude purge numpaquet
```

Lister si le paquet numpaquet est disponible à l'installation dans le gestionnaire de paquet et dire si il est déjà installé.

```
aptitude search numpaquet
```

Afficher des informations détaillées concernant le paquet numpaquet.

```
aptitude show numpaquet
```

Mettre à jour le système. Installe les versions candidates des paquets installés sans supprimer aucun autre paquet.

```
aptitude safe-upgrade
```

Mettre le système complètement à jour avec les dernières versions disponibles y compris si cela nécessite des désinstallations d'autres paquets (dépendances). full-upgrade est le synonyme de dist-upgrade. Permet aussi de passer d'une version de distribution à une autre.

```
aptitude full-upgrade
```

Option `-V --show-versions` pour lister les versions actuelles et futures qui vont être mises à jour.

```
aptitude -V full-upgrade
```

Tester une mise à jour du système pour étudier l'impact avant de la faire réellement (option -s : simulate).

```
aptitude safe-upgrade -s
```

Empêcher un paquet de se mettre à jour par un safe-upgrade ou un full-upgrade. Utile pour les kernels par exemple.

```
aptitude hold nomaquet  
aptitude unhold nomaquet
```

Nettoyer complètement le dépôt local des fichiers de paquets récupérés.

```
aptitude clean
```

Nettoyer le dépôt local des fichiers des paquets périmés

```
aptitude autoclean
```

Installation à partir de paquets téléchargés

Paquets rpm

Sites fournisseurs d'rpm.

- <http://rpm.pbone.net/>
- <http://rpmfind.net/>
- <http://dag.wieers.com/rpm/packages.php>
- <http://freshmeat.net/>

Pas mal de paquets sont récupérés sur le site de dag wieers, rpm.pbone et rpmfind qui fonctionnent depuis des années et font de très bons paquets.

A partir d'un de ces sites on choisit la distribution, l'architecture et on peut mettre à jour via un rpm -Uvh <nom du package> sur le serveur cible.

Commandes rpm

Installer un rpm téléchargé.

```
rpm -ivh nomaquet.rpm
```

Il faut savoir que lorsqu'un fichier rpm veut être téléchargé, il est souvent intéressant de l'installer

avec yum parce que si des dépendances sont demandés, il va essayer de les résoudre lui-même.

```
yum install nempaquet.rpm
```

Mettre à jour un paquet existant sinon installer le nouveau paquet.

```
rpm -Uvh nempaquet.rpm
```

Pour installer un paquet de force sans vérification aucune (à utiliser lorsqu'on sait ce qu'on fait).

```
rpm -ivh --nodeps --force paquet.rpm
```

Supprimer un paquet.

```
rpm -e nempaquet.rpm
```

De la même manière que pour l'installation, il est intéressant de supprimer le paquet avec yum car il va supprimer les paquets dont dépend le paquet spécifié qui ne servent plus à rien et qui ne sont plus utilisés par d'autres.

```
yum remove nempaquet
```

Connaître la liste totale des paquets triés.

```
rpm -qa | sort
```

Chercher un paquet d'un certain nom.

```
rpm -qa | grep nom_a_rechercher
```

Vous êtes en face d'un fichier de configuration par exemple et vous voulez savoir à quel paquet il appartient. Exemple avec /etc/named.conf.

```
rpm -qf /etc/named.conf
```

Liste des paquets installés avec description sous forme de colonnes.

```
rpm -qa --qf '%-30{NAME} : %{SUMMARY}\n' | sort
```

Lister les noms de paquets sous la même forme que les rpm habituels.

```
rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}-%{ARCH}\n"
```

Idem avec un descriptif des paquets.

```
rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}-%{ARCH} : %{SUMMARY}\n"
```

Afficher toutes les query possibles en queryformat (ex : %{NAME}).

```
rpm --querytags
```

Renvoie quel paquet rpm a installé un certain fichier : permet ainsi de déterminer le nom du paquet ayant installé tel programme ou telle librairie.

```
rpm -q --whatprovides /usr/bin/libgd.so
```

Liste des paquets qui nécessitent logrotate.

```
rpm -q --whatrequires logrotate
```

Liste des dépendances du paquet samba déjà installé.

```
rpm -q --requires samba
```

Lister les fichiers d'un paquet rpm.

```
rpm -ql nom_paquet
```

Lister uniquement les fichiers de configurations d'un paquet rpm.

```
rpm -qlc nom_paquet
```

Connaître le paquet d'un fichier.

```
rpm -qf /etc/httpd/conf/httpd.conf
```

Connaître le contenu d'un rpm.

```
rpm -pql package.rpm
```

Connaître les derniers paquets installés.

```
rpm -qa --last
```

Vérifier les fichiers d'un paquet installés. La vérification d'un paquetage compare les informations sur les fichiers installés dans le paquetage avec les informations sur les fichiers obtenues à partir des méta-données du paquetage original conservées dans la base de données rpm. Entre autres choses, la vérification compare la taille, la somme MD5, les permissions, le propriétaire et le groupe de chaque fichier. Toutes les discordances sont affichées.

```
rpm -V nom_paquet
```

Sur cette commande, un résultat n'est affiché que lorsqu'une différence est constatée. Pour la signification des retours, se reporter au man rpm.

Paquets deb

Site fournisseur de deb : <http://www.getdeb.net/>

Commandes dpkg

Installer un paquet.

```
dpkg -i nempaquet.deb
```

Lister les paquets du système.

```
dpkg -l
```

Lister les fichiers d'un paquet déjà installé.

```
dpkg -L nom_paquet
```

Lister les fichiers que fournit un paquet .deb.

```
dpkg-deb -c nom_paquet.deb
```

Retrouver le nom d'un paquet à partir d'un fichier.

```
dpkg -S nom_fichier  
dpkg -S <chemin_absolu>/nom_fichier
```

Lister/auditer tous les paquets du systèmes, lister ceux qui ont été installés partiellement et suggestion de ce qu'il faut faire pour résoudre cette situation.

```
dpkg -C
```

Supprimer un paquet.

```
dpkg -e nempaquet.deb
```

Supprimer un paquet complètement avec tout les fichiers de configurations (P=purge).

```
dpkg -P nempaquet.deb
```

Rechercher la présence d'un fichier dans les paquets installés.

```
dpkg-query -S ssh2.ini
```

Liste des derniers paquets installés.

```
grep " install " /var/log/dpkg.log
```

```
ls -rtl /var/lib/dpkg/info/*.list  
vi /var/log/apt/history.log
```

Liste des derniers paquets mis à jour.

```
grep " upgrade " /var/log/dpkg.log
```

Création d'un repository yum

Un repository est un dossier local ou distant qui contient une liste de rpm avec toutes les dépendances requises organisées de manière cohérente. L'idée est de pouvoir utiliser le gestionnaire de paquets yum présent sous Red Hat qui permet notamment d'installer des paquets avec les dépendances de manière automatique. Ce paragraphe explique comment utiliser ou mettre en place le repository de manière local sur la machine à installer.

Pour recréer ce repository, suivre les informations qui suivent.

Installer createrepo

Installer le paquet createrepo-0.4.11-3.el5 fournit dans le DVD Red Hat qui a pour fonction de créer des repository yum.

```
rpm -ivh createrepo-0.4.11-3.el5.noarch.rpm
```

Récupérer les paquets supplémentaires du repository

Récupérer tous les rpm du DVD officiel et les ajouter dans le dépôt. Le dossier choisi pour être le repository yum est /data/rhel-repos. Nous devons monter la version de PHP, MySQL et PHP-pear. Supprimer les anciennes versions de paquets php-*, et mysql-* présent dans le DVD officiel pour ne pas entrer en conflit avec les nouveaux que nous allons ajouter.

Sur une machine qui dispose d'un accès internet, il faut récupérer sur un dépôt d'Internet les paquets PHP récents (version 5.3.3-1) et MySQL de manière cohérente. Le dépôt utilisé pour les paquets PHP est celui de Remi Collet (<http://blog.famillecollet.com/>) et d'Extra Package Enterprise Linux (EPEL <http://fedoraproject.org/wiki/EPEL>).

Configurer les dépôts remi et EPEL sur la machine qui dispose d'Internet.

```
wget  
http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-3.noarch.rp  
m  
wget http://rpms.famillecollet.com/enterprise/remi-release-5.rpm  
rpm -Uvh remi-release-5*.rpm epel-release-5*.rpm  
yum clean all //vider toutes les informations des dépôts.
```



```
yum update //mettre à jour les dépôts de la machine.
```

Nos dépôts sont installés pour récupérer les paquets que nous souhaitons. L'outil qui permet de télécharger les paquets d'un dépôt avec les dépendances associées est yumdownloader. Yumdownloader sert à télécharger des paquets et leurs dépendances mais sans rien installer et qu'on veut pouvoir copier facilement sur un autre ordinateur avec exactement la même liste de paquets que sur notre ordinateur. Il a été utilisé pour récupérer les paquets PHP et PHP-pear. Yumdownloader est un outil présent dans le paquet yum-utils qui est déjà présent dans le DVD Red Hat officiel.

```
rpm -ivh yum-utils-1.1.16-13.el5.noarch.rpm
```

Récupérer les paquets PHP et PHP-pear en même temps.

```
yumdownloader --resolv php-*
```

Cette commande récupère à la racine d'où est lancée la commande tous les paquets et dépendances requises pour tous les paquets PHP.

Pour MySQL, les paquets ont été récupérés de l'excellent site <http://rpm.pbone.net> en version community.

La totalité des rpms qu'on vient de récupérer avec yumdownloader comme décrit ci-dessus sont à regrouper dans un dossier par exemple php_mysql_php-pear. Ces rpms constituent, MySQL Community, la dernière version de PHP et de PHP-pear avec toutes les dépendances nécessaires.

Copier les paquets récupérés dans le dépôt qui ne contient que les paquets officiels pour le moment.

```
cp /chemin/php_mysql_php-pear/* /data/rhel-repos
```

Il est aussi possible de présenter directement le contenu de l'iso officielle.

```
mount -o loop /path/to/rhel.iso /data
```

Créer le repository

Positionner/vérifier les droits sur tous les rpm du dépôt.

```
cd /data/rhel-repos  
chmod 444 *
```

Créer le repository yum.

```
createrepo /data/rhel-repos
```

Cette commande crée un dossier /data/rhel-repos/repodata avec les metadata nécessaires pour le dépôt.

Si un paquet est de nouveau ajouté ou supprimé du dépôt, il faut supprimer le dossier repodata et relancer la commande de création du dépôt.

Configurer yum

Il ne reste qu'à configurer yum pour prendre en compte notre dépôt local.

```
vi /etc/yum.repos.d/rhel-repos.repo (extension .repo obligatoire).
```

```
[localrepo]
name=OS $release – MyLocalRepo
baseurl=file:///data/rhel-repos/
enabled=1
gpgcheck=0
```

Tester le fonctionnement du repository

```
yum clean all //vider les informations stockées par yum.
yum list "php*"
yum list httpd
```

Ces deux commandes doivent renvoyer une liste de paquets.

Le repository yum local est maintenant créé. Il va nous servir pour automatiser l'installation des rpm pré-requis pour l'atelier installation et la gestion de configuration.

Configurer un repository yum pour utiliser le DVD officiel Red Hat ou CentOS

Editer un fichier de dépôt yum.

```
vi /etc/yum.repos.d/dvd.repo
```

Ajouter les instructions suivantes. Le mediaid est un chiffre aléatoire ici et ça ne pose pas de problème.

```
[dvd]
mediaid=1170972069.396645*
name=DVD for RHEL5
baseurl=file:///media/cdrom/Server
enabled=1
gpgcheck=0
```

On peut utiliser les paquets du DVD officiel pour installer des paquets.

```
yum update
```

```
yum list, yum install...
```

Appliquer mise à jour d'un repo local

```
reposync -n -p /tools/
```

Cette commande permet sur un serveur accédant à Internet de mettre à jour un repo local avec les dernières versions des paquets. Permet de pouvoir créer un repo local à la base de la toute dernière version des paquets et non de celle en date du DVD officiel.

Subscription Red Hat Developer

Il est possible d'avoir une RHEL officielle gratuitement via ce site : <https://developers.redhat.com/>

Après enregistrement, configurer subscription-manager pour ne proposer que les mises à jour de la 8.2 et pas passer automatiquement à la 8.3 (par défaut)

```
subscription-manager release --set=8.2
```

Passer à la 8.4, demandera de seter la valeur à 8.4 et de faire un classique dnf update ensuite.

Troobleshoots

rpmdb: Lock table is out of available locker entries

Si up2date ou yum ou une commande rpm envoi une erreur Python qui dit rpmdb: Lock table is out of available locker entries, on peut restaurer la base rpm avec les étapes suivantes.

Les erreurs.

```
rpmdb: Lock table is out of available locker entries
error: db4 error(22) from db->close: Invalid argument
error: cannot open Packages index using db3 - Cannot allocate memory (12)
error: cannot open Packages database in /var/lib/rpm
```

Ces erreurs empêchent de faire une quelconque interrogation sur la liste des paquets, aucun ajout, modification ou suppression n'est possible. Les commandes de mise à jour yum ne fonctionnent plus non plus.

Faire une sauvegarde de /var/lib/rpm au cas où.

```
tar cvzf /root/rpmdb_backup.tar.gz /var/lib/rpm
```

Supprimer la base de données Berkley databases utilise par rpm.

```
rm /var/lib/rpm/___db.00*
```

Reconstruire la base de données rpm depuis 0 (prend un peu de temps).

```
rpm --rebuilddb
```

Vérifier que la liste des rpm est accessible pour s'assurer que la base fonctionne bien.

```
rpm -qa | sort
```

Vérifier que les fichiers ___db00* ont été recréés.

```
ls -l /var/lib/rpm
```

Pourquoi ceci arrive ?

Quand rpm accède à sa base de données fichier Berkley database, il crée un fichier de verrou temporaire sur les tables dans lequel il cherche. Si on interrompt l'opération avec Ctrl+C souvent, ce message peut arriver car les verrous ne sont jamais supprimés.

From:

<https://wiki.ouieuhtoutca.eu/> - **kilsufi de noter**

Permanent link:

https://wiki.ouieuhtoutca.eu/gestionnaire_paquets_gnu_linux

Last update: **2021/01/21 21:42**

