

Installation d'applications par déploiement de paquets binaires

Un paquet contient :

- les fichiers (binaires, bibliothèques, fichiers de configuration, ressources, etc.) nécessaires pour exécuter un programme ;
- un ensemble de scripts qui configurent le programme automatiquement après son installation ;
- les informations sur le propriétaire et permissions d'accès de chaque fichier ;
- des informations optionnelles sur les paquets dépendants et les services fournis ;
- une description du paquet.

L'ensemble des informations de tous les paquets installés sont stockés dans une base de données (*/var/lib/rpm*) gérée par l'utilitaire *rpm* (Redhat Package Management) pour les distributions basées sur RedHat et les outils *dpkg* et *apt* pour la distribution Debian.

INSTALLATION PAR PAQUETS

Manuellement	Automatiquement
Commande rpm .	Commande yum .
Assure de déploiement.	Assure de déploiement.
Permet la création de paquets. (commande rpmbuild du paquet rpm-build)	Ne permet pas la création de paquets.
Ne gère pas les dépendances.	Gère les dépendances.

- **Nomenclature**

Le nom d'un fichier de paquet est constitué du nom du paquet, du numéro de version du logiciel et du numéro de version du paquet séparés par des '-'.
Par exemple : openssl-1.0.2k-8.el7.x86_64.rpm

- **Exemples d'utilisation**

- installation d'un paquet `rpm -i fichier.rpm`
!! le .rpm peut être local ou distant (accessible via une URL)
exemple: `rpm -ivh fichier.rpm`
 - v: verbose
 - h: barre de progression
- mise-à-jour d'un paquet `rpm -U fichier.rpm`
- suppression d'un paquet `rpm -e paquet`
- vérification d'un paquet `rpm -V paquet`
- interrogation sur les paquets installés:
 - . liste des fichiers d'un paquet `rpm -ql paquet`
 - . paquet contenant un fichier `rpm -qf /chemin/vers/fichier`
 - . informations sur un paquet `rpm -qi paquet`
 - . liste triée de tous les paquets installés `rpm -qa | sort`
 - . savoir si un paquet précis est installé `rpm -qa | grep -i package`
 - . date d'installation des paquets `rpm -qa --last`
- interrogation sur les paquets non-installés:
 - . liste des fichiers `rpm -qp1 fichier.rpm`
 - . informations `rpm -qpi fichier.rpm`
 - . liste des dépendances `rpm -qpR fichier.rpm`

- **Gestion des signatures GPG**

- *rpm* supporte la signature des paquets.
- Cette technique permet de garantir qu'un paquet n'a pas subi de modification (intégrité du paquet) et que la personne qui l'a construit est bien celle qu'elle prétend être.
- Il faut pour cela importer les clés publiques GPG des personnes ou organisations qui réalisent les paquets devant être installés avec l'option `--import` de *rpm*.
- L'option `--checksig` permet de vérifier la signature d'un paquet avant de l'installer.
- La clé publique de la distribution CentOS est disponible sur les CD-ROM, dans le fichier RPM-GPG-KEY.

Ateliers rpm

1^{er} cas: Manipulations de base

1. Listez les applications installées.
2. Une application telle que 'openssh' est-elle installée ?
(Il faut connaître le nom du paquet)
3. Affichez les caractéristiques d'un paquet.
4. Listez les fichiers appartenant à un paquet.
5. A quel paquet déjà installé appartient le fichier /etc/ssh/sshd_config ?
6. Installez le paquet lynx (à partir des paquets du dépôt CentOS)

Téléchargement

Installation

Vérification
7. Désinstallez le paquet précédent.
8. On s'informe sur le paquet 'lynx' téléchargé qui n'est pas installé.
 - a) On liste ses caractéristiques.
 - b) On liste les fichiers qui le composent.
 - c) On liste les dépendances de 'lynx'.

2^{ème} cas: Signatures

Prenons , par exemple, le cas du paquet webmin :

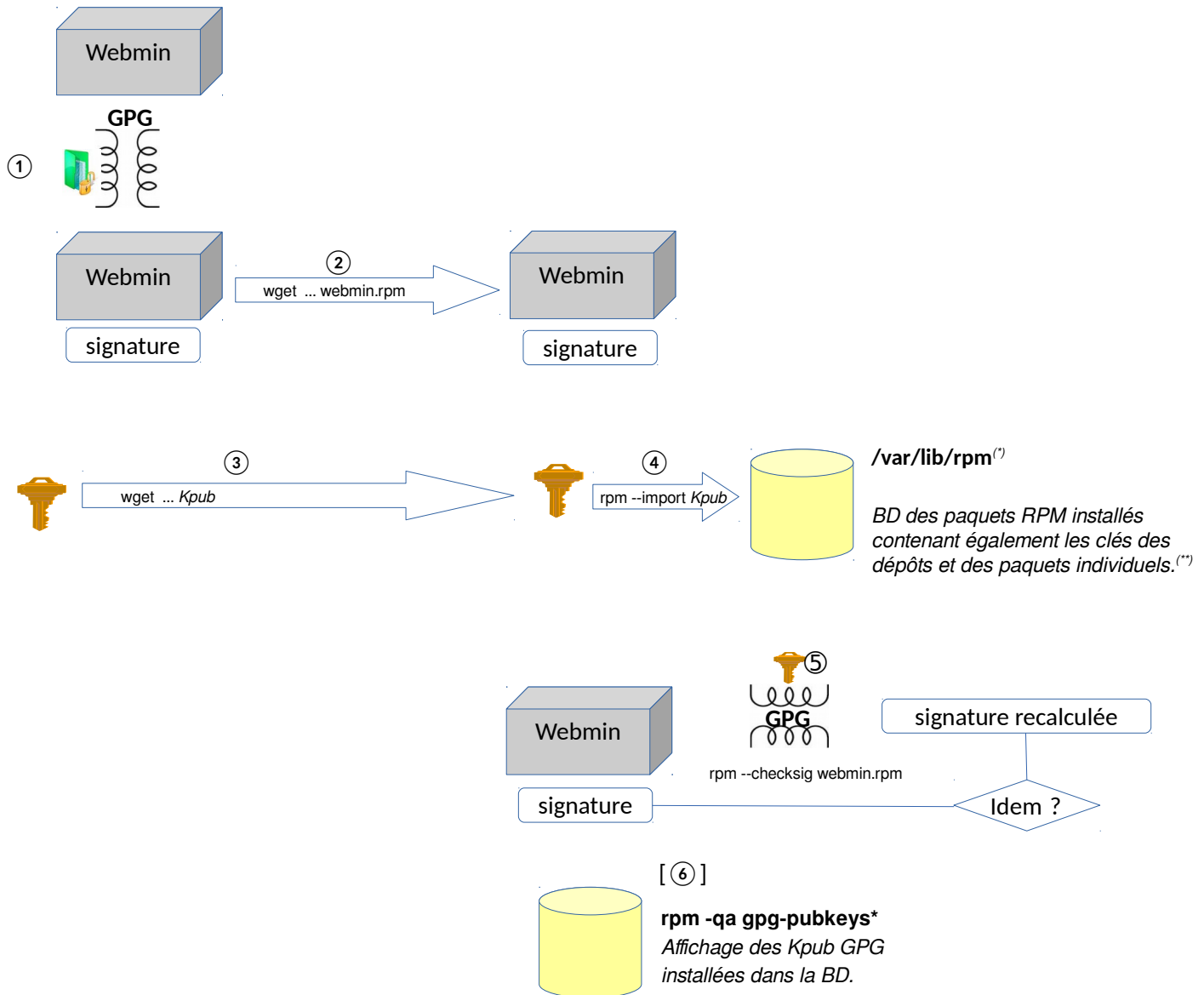


L'administrateur a généré un trousseau de clés
(publique/privée) afin de pouvoir signer son paquet.

Kpr Kpu

En cryptographie :

- on SIGNE avec la Kpr et on vérifie la signature avec la Kpub.
- on CHIFFRE avec la Kpu et on déchiffre avec la Kpr.



(*) **/var/lib/rpm**: Ensemble de bases de données (au format Berkeley pour certaines) qui contiennent notamment les Kpub *effectivement utilisées* par 'rpm' ou 'yum' afin de vérifier l'intégrité du paquet avant de l'installer.

(# `file /var/lib/rpm/*` → on constate que les fichiers sont de type Berkeley...)

/etc/pki/rpm-gpg: Dossier d'archivage des clés publiques.

Il contient déjà toutes des Kpub de tous les paquets de la distribution CentOS.

Après téléchargement et importation de la Kpub de Webmin, on pourrait la transférer dans ce dossier...

(**) Lors de l'import d'une clé publique GPG, celle-ci sera ajoutée à la base de données `_db.003`.

Pour preuve, à l'aide d'un éditeur, une recherche sur le mot clef 'GPG' nous envoie sur une clé publique GPG...

Remarque

GPG : Norme plus récente que PGP et libre !

Plus d'info

<http://www.commentcamarche.net/contents/206-pgp-pretty-good-privacy>

Téléchargement du paquet webmin.

Tentative de vérification de l'intégrité du paquet.

Téléchargement de sa clé gpg.

Enregistrement de cette clé dans le système

Vérification de l'intégrité du paquet.

Affichage des clés installées :

Informations sur une clé particulière :

Suppression d'une clé :

- **Liens**

<http://www.rpm.org>

http://fr.wikipedia.org/wiki/RPM_Package_Manager

<http://rpmfind.net>

Un dépôt CentOS (repository) est un ensemble organisé et indexé de paquets.

Il existe plusieurs types de dépôts CentOS.

- **Dépôts de base**

Les dépôts suivants sont installés:

- **base :** Dépôt où sont stockés tous les paquets de base de la distribution CentOS.
- **updates :** Mises à jour des paquets contenus dans le dépôt de base.
- **extras :** Paquets maintenus et testés par les développeurs de CentOS. Ces paquets ne sont pas considérés comme pouvant intégrer le dépôt de base. Il est donc déconseillé de les installer sur un serveur.
- **centosplus :** Pas activé par défaut. Les paquets de ce dépôt sont maintenus par les développeurs de CentOS et des utilisateurs. Certains de ces paquets peuvent écraser des paquets de la distribution originale.
- **CR, DebugInfo, Media, Sources, Vault, fasttrack :** D'autres dépôts CentOS qui ne sont pas activés par défaut. (hors cadre du cours)

- **Dépôts tiers**

- **EPEL** (Extra Packages for Enterprise Linux) (Liste des paquets via Google : [index EPEL](#))
Paquets conçus pour Fedora (ex. le serveur de sauvegarde 'Backup PC', le serveur ftp 'proftpd')
Installation: <http://fedoraproject.org/wiki/EPEL/FAQ#howtouse>
- **Elrepo** (Enterprise Linux Repository)
Paquets qui étendent la compatibilité matérielle des distributions RHEL et CentOS. On y trouve notamment des pilotes propriétaires pour les cartes graphiques ou les cartes ethernet.
Installation: <http://elrepo.org/tiki/tiki-index.php>
- **nux-desktop** (Liste des paquets via Google : [index nux-desktop](#))
Ce dépôt est prévu pour fonctionner de pair avec le dépôt EPEL. Il propose de nombreux logiciels très communs chez les particuliers, notamment Steam, VLC ou Skype.
(Le dépôt EPEL doit être installé et activé...)
Installation: <https://www.certdepot.net/rhel7-install-nux-repository/>
- **Adobe**
Contient le plugin Flash pour le navigateur Mozilla Firefox.
Installation: <https://www.tecmint.com/install-adobe-flash-on-centos-rhel-fedora-linux/>
- **RPM Fusion**
Fournit de nombreux logiciels très utiles qui ne peuvent être inclus au sein du Projet Fedora pour une raison ou une autre. Il existe une distinction entre libre (free) et non-libre (nonfree).

La section libre contenant des programmes dont la licence est compatible avec le Projet Fedora mais ayant une partie de code soumise à des brevets logiciels (interdit dans le logiciel libre au États-Unis).

La section non-libre contenant les programmes redistribuables mais ayant une licence propriétaire
Installation: <http://rpmfusion.org/Configuration>
(Le dépôt EPEL doit être installé et activé...)
- **Remi Collet**
Contient des versions plus récentes de MySQL et de PHP que celles fournies avec CentOS.
Installation: <http://rpms.famillecollet.com>

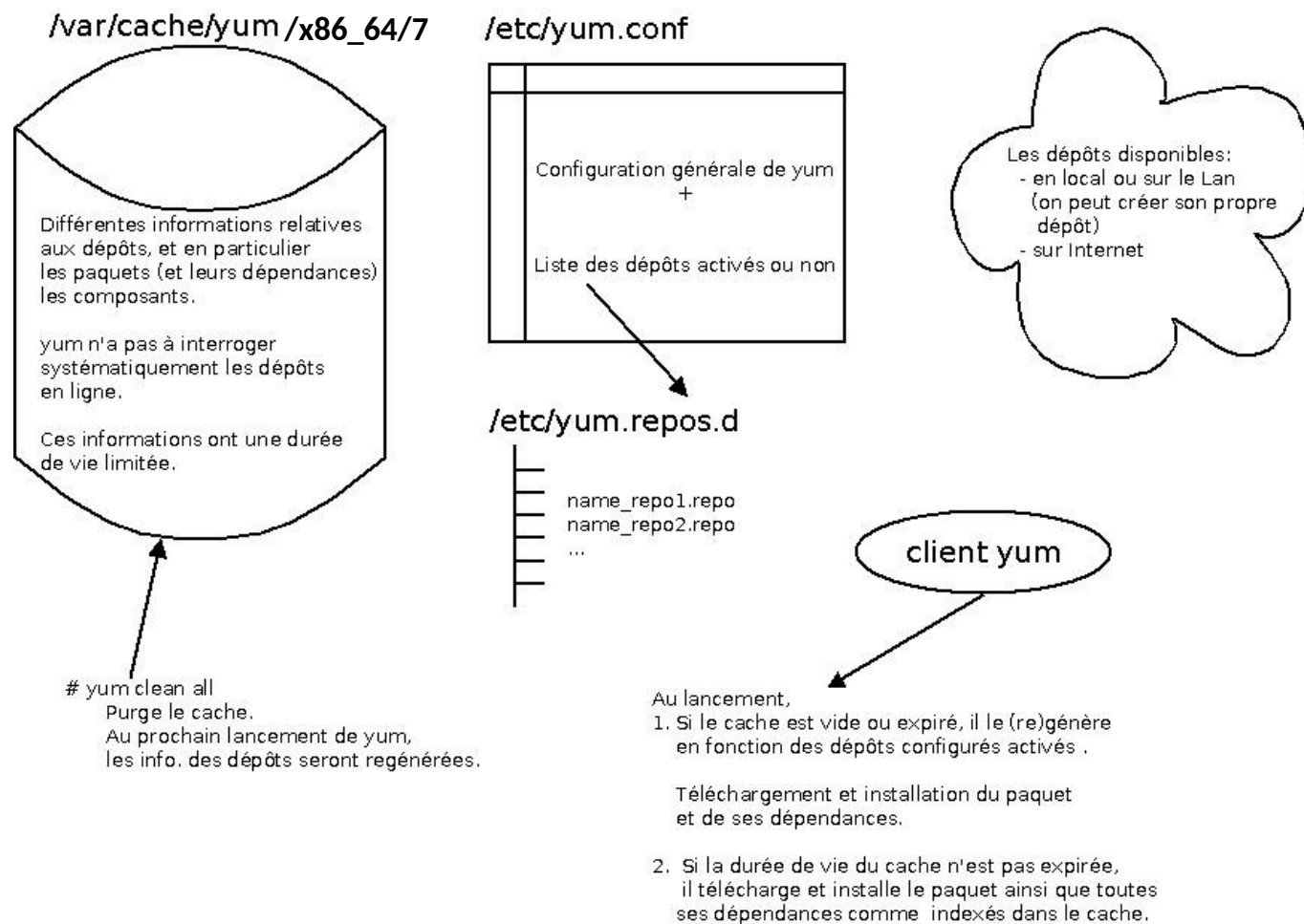
- **VirtualBox**

Installation: <http://download.virtualbox.org/virtualbox/rpm/fedora/virtualbox.repo>

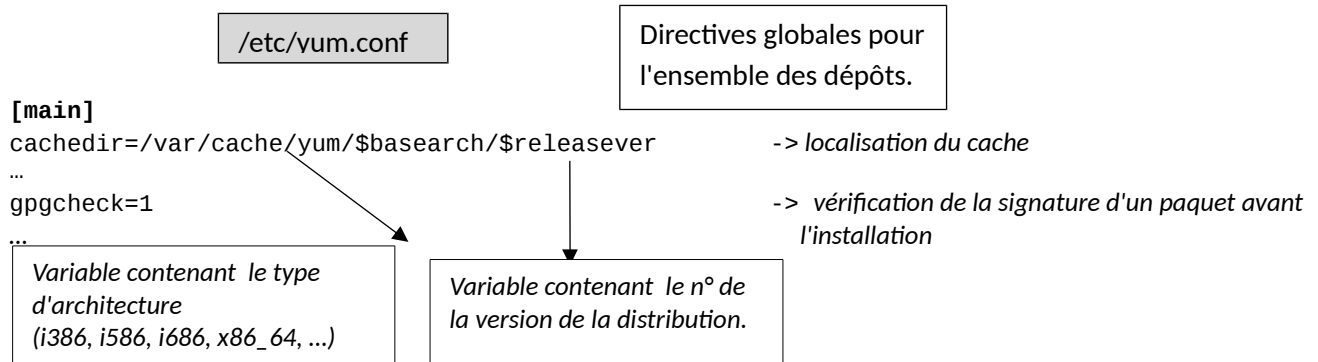
- ...

• **Architecture de yum**

Le gestionnaire de paquets 'yum' permet notamment d'installer automatiquement des packages mais aussi toutes leurs dépendances.



- Exemple de configuration de yum



/etc/yum.repos.d/CentOS-Base.repo

[base]
 name=CentOS-\$releasever - Base -> nom du dépôt
 mirrorlist=http://mirrorlist.centos.org/?release=\$releasever&arch=\$basearch&repo=os&infra=\$infra
 -> url vers un fichier contenant une liste de 'baseurl'.
 #baseurl=http://mirror.centos.org/centos/\$releasever/os/\$basearch/
 -> url vers un dossier qui contient toute l'arborescence du dépôt

gpgcheck=1 -> Faut-il réaliser une vérification de la signature gpg de tout paquet provenant de ce dépôt ? (1=oui 0=non)
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
 -> Localisation de la clé publique du dépôt. Clé de référence utilisée si la clé codée dans la bd rpm /var/lib/rpm/Pubkeys a été supprimée.

#released updates
 [updates]
 name=CentOS-\$releasever - Updates
 mirrorlist=http://mirrorlist.centos.org/?release=\$releasever&arch=\$basearch&repo=updates&infra=\$infra
 #baseurl=http://mirror.centos.org/centos/\$releasever/updates/\$basearch/
 gpgcheck=1
 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that may be useful
 [extras]
 name=CentOS-\$releasever - Extras
 mirrorlist=http://mirrorlist.centos.org/?release=\$releasever&arch=\$basearch&repo=extras&infra=\$infra
 #baseurl=http://mirror.centos.org/centos/\$releasever/extras/\$basearch/
 gpgcheck=1
 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that extend functionality of existing packages
 [centosplus]
 name=CentOS-\$releasever - Plus
 mirrorlist=http://mirrorlist.centos.org/?release=\$releasever&arch=\$basearch&repo=centosplus&infra=\$infra
 #baseurl=http://mirror.centos.org/centos/\$releasever/centosplus/\$basearch/
 gpgcheck=1
 enabled=0
 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

- **Exemples d'utilisation**

- **Installation**

- . Pour installer un paquet: `# yum install <pkg>`
 - . Idem sans confirmation: `# yum install <pkg> -y`
 - . Pour installer un rpm particulier
avec ses éventuelles dépendances `# yum localinstall <rpm>`

- **Mise-à-jour**

- . Maj de yum lui-même (avant de faire un update ou upgrade): `# yum update yum`
 - . Maj de tous les paquets (non obsolètes) à la dernière version disponible (correction d'erreur, amélioration) `# yum update`
 - . Idem 'update' sauf que les paquets obsolètes (pas de mise-à-jour possible) sont supprimés et remplacés comme il convient. Cela permet de passer de la version x.y à x.(y+1) `# yum upgrade`
 - . Maj d'un paquet `# yum update <pkg>`
 - . Maj sélective `# yum --exclude=<pkg> update`
 - . Maj de tous les paquets (sauf ceux du kernel) `# yum --exclude=package kernel* update`
 - . Connaître les paquets qui ont besoin d'être maj `# yum check-update`

- **Suppression:**

- `# yum remove <pkg>`

- **Interrogation**

- . Pour rechercher un paquet particulier: `# yum list <pkg>`
 - . Si l'on ne connaît pas le nom du paquet, on peut utiliser le caractère joker * (étoile). YUM listera les paquets correspondants à votre recherche déjà installés, et ceux que vous pouvez installer: `# yum list kerne*`
 - . Connaître la liste de tous les paquets installés: `# yum list installed`
 - . Connaître le nom du paquet qui possède un fichier: `# yum provides <fichier>`
ex. `yum provides /sbin/ifconfig` ou `# yum whatprovides /sbin/ifconfig`
`yum provides */ifconfig` ou `# yum whatprovides */ifconfig`
 - . Connaître les paquets contenant un mot clé dans leur descriptif: `# yum search <mot_cle>`
 - . Connaître les paquets contenant un mot clé dans leur descriptif: `# yum -C search <mot_cle>`
(la recherche se fait dans le cache)
 - . Connaître les infos d'un paquet: `# yum info <pkg>`
 - . Connaître les dépendances d'un paquet: `# yum deplist <pkg>`
 - . Connaître la liste des dépôts actifs: `# yum repolist`

- **Nettoyage du cache** et des paquets déjà téléchargés `# yum clean all`
et installés (ils restent dans le cache!!)

- **Gestion de groupes de paquets**

- Un groupe est une collection de paquets spécifiques à un sujet tel que "Development Tools".

- . Connaître les groupes disponibles: `# yum grouplist | more`
 - . Installation d'un groupe : `# yum groupinstall 'Development Tools'`
 - . Suppression d'un groupe: `# yum groupremove 'Development Tools'`
 - . Mise à jour d'un groupe: `# yum groupupdate 'Development Tools'`
 - . Liste des paquets d'un groupe: `# yum groupinfo 'Development Tools'`

- **Création de son propre dépôt**

- <https://blog.microlinux.fr/yum-local/>
 - <http://www.supinfo.com/articles/single/1512-repository-local-centos-redhat>
 - <https://www.it-connect.fr/centos-creer-son-propre-repository-local/>

Ateliers yum

1er cas : Installation du dépôt EPEL

1. Vérifiez si le dépôt epel est activé.
2. Vérifiez si le dépôt epel est installé.
3. Si c'est le cas, désinstallez-le.
4. Vérifiez.
5. Réinstallez-le.

Vérifiez l'architecture de votre processeur (ici x86_64)

Téléchargez le paquet d'installation du dépôt epel pour el6_64bits (*)

Importez la clé du repository (*)

Vérifiez l'intégrité du dépôt (*)

Installez le paquet (dépôt) (*)

(*) ou

6. Empêchez yum de naviguer par la suite dans ce dépôt, mettre l'option enabled à 0 dans le epel.repo

Voir si le dépôt est bien désactivé

7. Testez l'installation d'un paquet se trouvant dans le dépôt désactivé

8. Réautorisez à yum à naviguer par la suite dans ce dépôt

Désinstallez 2ping

Videz le cache de yum

Reconstruisez le cache

Vérifiez si le dépôt est bien activé

9. Désinstallez 2ping

10. Installez 2ping (plus besoin de forcer car le dépôt est actif)

2^{ème} cas : Gestion des signatures

1. Visualisez les clés recopiées automatiquement dans `/etc/pki/rpm-gpg` lors de l'installation du dépôt epel.
2. Visualisez toutes les clés gpg enregistrées dans la bd rpm (`/var/lib/rpm`) lors de l'installation des dépôts. Ce sont ces clés qui serviront à vérifier la signature.
3. Désinstallez 2ping.
4. Repérez quelle est la clé système utilisée pour vérifier l'intégrité des paquets provenant du dépôt epel.
5. Supprimez la clé système du dépôt epel.
6. Vérifiez si cette clé est bien absente.
7. Réinstallez 2ping.

Yum ne sait plus vérifier l'intégrité du paquet et demande s'il faut réenregistrer la clé de référence se trouvant dans `/etc/pki/rpm-gpg` afin de pouvoir poursuivre.

Remarques

	<code>/etc/pki/rpm-gpg</code>	<code>/var/lib/rpm/Pubkeys</code>
<code>rpm --checksigrpm</code> ou <code>rpm -Krpm</code>	×	✓
<code>rpm --import</code>	×	✓
<code>rpm -qa gpg-pubkey*</code>	×	✓
<code>rpm -e gpg-pubkey...</code>	×	✓
<code>yum install ...</code> (<code>gpgcheck=1</code>)	✓ en cas de nécessité de restauration de la clé du dépôt.	✓