

Chapitre 4

Installation de logiciels

1. Introduction et historique

L'installation de logiciels est sans aucun doute l'un des points les plus sensibles de l'administration système. En effet, certaines tâches auparavant peu évidentes sous Linux, ont été grandement facilitées comme l'installation du système par exemple. D'autres tâches font appel à des compétences techniques importantes et laissent peu de place au hasard, citons la gestion des utilisateurs.

Cependant, pour certaines raisons, la gestion des installations de logiciels est un sujet qu'il faut comprendre et maîtriser techniquement mais aussi théoriquement.

L'une des principales raisons vient du fait qu'il n'y a pas de consensus dans le monde Linux, et dans le monde du libre en général. Là où d'autres systèmes d'exploitation propriétaires ont imposé leur schéma de gestion des logiciels, il y a dans le monde Linux plusieurs options proposées, variables dans leur efficacité et adaptables aux besoins.

Cette variété de systèmes de gestion de logiciels s'explique par la multiplicité des distributions Linux d'un côté, et le grand nombre de plates-formes matérielles supportées de l'autre.

Pour un système d'exploitation propriétaire qui se réduit à quelques moutures sur une ou deux plates-formes matérielles, on trouve des centaines de distributions Linux sur des dizaines de plates-formes, sur lesquelles doit travailler l'auteur d'un logiciel.

Ainsi, alors qu'au début chaque administrateur système devait lui-même compiler le logiciel à installer, avec le temps sont apparues des solutions de plus en plus pratiques.

Gestion des logiciels - Historique

Retour aux sources

Dans ce chapitre, nous allons aborder les différentes façons d'installer un logiciel, en terminant par la plus ancienne. À l'origine, bien souvent l'auteur d'un logiciel fournissait le code source, que les administrateurs devaient compiler eux-mêmes pour le faire fonctionner sur leur machine.

Aujourd'hui encore, il arrive que des logiciels ne soient fournis que sous la forme du code source, ou bien qu'ils ne soient pas disponibles sous forme de binaire pour votre distribution. Dans ce cas, il faudra le compiler vous-même, opération gourmande en temps et qui demande un minimum de connaissances techniques.

Un joli cadeau

C'est pourquoi les éditeurs de distributions Linux ont vite compris la nécessité de produire des logiciels déjà compilés pour leur distribution. Red Hat précisément, est même allé plus loin en sortant le format de paquet RPM (*Red Hat Packet Manager*, gestionnaire de paquets Red Hat) et les outils associés, permettant de faciliter la distribution d'un logiciel. Avec RPM, le programme sous sa forme binaire est « empaqueté » avec ses fichiers de configuration et d'aide. Lorsque le programme est désempaqueté, il est installé automatiquement sur le système. L'utilisateur n'a plus qu'à taper la commande dans le shell.

Remarque

*Un paquet est donc un logiciel « empaqueté » avec les différents fichiers qui lui sont nécessaires : fichiers de configuration et fichiers d'aide entre autres. Autre précision : alors qu'en anglais, on utilise le mot « packet », en français on utilise indifféremment « **paquet** » ou « **paquetage** ».*

Notons que d'autres éditeurs de distributions ont plus ou moins au même moment sorti des systèmes de paquets. Un autre bien connu est le format de paquet .deb, utilisé principalement dans les distributions Debian et Ubuntu.

Le temps de l'indépendance

Cependant, cela ne résolvait pas un autre problème. En effet, pour fonctionner, un programme dépend bien souvent de bibliothèques tierces, sans lesquelles il ne peut pas fonctionner. Ces dépendances doivent également être installées sous la forme de paquetages. Ainsi, pour installer un logiciel, l'utilisateur devait télécharger le paquet principal du programme et l'ensemble de ses dépendances et les installer les uns après les autres. L'opération étant contraignante, les gestionnaires de paquets sont alors apparus.

Un gestionnaire de paquets est un programme qui indexe l'ensemble des applicatifs disponibles sous forme de paquets et l'ensemble des dépendances dont ils ont besoin (qui peuvent être très nombreuses). Tout cela est stocké dans des bases dont la taille est à l'évidence assez importante.

Le gestionnaire de paquets fait ainsi à votre place la « résolution des dépendances ». Lorsque vous installez un logiciel, il cherche dans sa base quels sont les autres paquets nécessaires et installe le tout, ce qui rend la tâche simple pour l'utilisateur.

Mieux encore, tous ces paquets sont stockés dans des dépôts logiciels (*repositories* en anglais), dans lesquels le gestionnaire de paquet va se servir. Il existe des dépôts avec des milliers de logiciels disponibles. Red Hat alimente ses serveurs avec ses dépôts officiels mais il en existe d'autres pour ce système d'exploitation.

Il est possible de gérer l'installation de paquets sur un serveur Red Hat Enterprise Linux avec YUM (maintenant DNF).

2. Installation avec le gestionnaire de paquets YUM

2.1 Introduction à YUM

YUM (*Yellow Dog Updater Modified*) est le gestionnaire de paquets officiel de Red Hat Enterprise Linux. Basé sur le format de paquet RPM de Red Hat, il est l'œuvre des auteurs de la distribution Yellow Dog Linux mais est utilisé sur beaucoup d'autres distributions que Red Hat.

À partir de RHEL 8, l'outil DNF remplace YUM. C'est pourquoi la commande `dnf` remplace peu à peu la commande `yum` dans l'ensemble du monde Red Hat, ainsi que dans la suite de cet ouvrage.

■ Remarque

Il faut être super-utilisateur `root` pour pouvoir installer, supprimer ou mettre à jour des paquets. De plus, il faut que le système soit enregistré auprès du Portail Client pour pouvoir télécharger des paquets à partir des dépôts de Red Hat. Pour savoir comment enregistrer le système, rendez-vous au chapitre *Déploiement d'un serveur Red Hat*. Dernier détail important : il faut que l'heure sur le système soit correcte.

■ Remarque

Les paragraphes suivants développent l'installation de paquets en ligne de commande. Il existe néanmoins un gestionnaire de paquet en interface graphique, dans le menu **Système - Administration - Ajouter/Enlever des logiciels**.

Mise à jour de YUM avec la version 4, ou DNF

La version 4 de YUM est en fait l'outil DNF (pour *Dandified Yum*, de *dandy*, élégant) :

```
[root@scott ~]# ls -lh /usr/bin/yum
lrwxrwxrwx. 1 root root 5 14 févr. 07:02 /usr/bin/yum -> dnf-3
[root@scott ~]# yum -version
4.0.9
  Installés : dnf-0:4.0.9.2-5.e18.noarch à lun. 17 juin 2019 21:35:49 GMT
(...)
  Installés : rpm-0:4.14.2-9.e18.x86_64 à lun. 17 juin 2019 21:24:16 GMT
```

Dans la pratique, cela ne change pas grand-chose, à part que les commandes `dnf` et `yum` sont absolument **interchangeables**.

```
[root@scott ~]# dnf repolist
Updating Subscription Management repositories.
(...)
id du dépôt                                nom du dépôt
rhel-8-for-x86_64-appstream-rpms           Red Hat Enterprise Linux 8
for x86_64 - AppStream (RPMs)
rhel-8-for-x86_64-baseos-rpms             Red Hat Enterprise Linux 8
for x86_64 - BaseOS (RPMs)
[root@scott ~]# yum repolist
(...)
id du dépôt                                nom du dépôt
rhel-8-for-x86_64-appstream-rpms           Red Hat Enterprise Linux 8
for x86_64 - AppStream (RPMs)
rhel-8-for-x86_64-baseos-rpms             Red Hat Enterprise Linux 8
for x86_64 - BaseOS (RPMs)
[root@scott ~]#
```

DNF est très simple d'utilisation. Voici un résumé des commandes principales.

Lister les dépôts logiciels

Pour afficher la liste des dépôts actifs disponibles, utilisez la commande :

```
■ dnf repolist
```

Cette commande affiche seulement les dépôts actifs. Si vous désirez également voir les dépôts existants mais inactifs, ajoutez `all` :

```
■ dnf repolist all
```

Par exemple, nous constatons ici qu'il existe beaucoup de dépôts, mais un seul est actif :

```
[root@rhel9 ~]# dnf repolist all
Mise à jour des référentiels de gestion des abonnements.
id du dépôt                                nom du dépôt                                état
rhel-9-for-x86_64-appstream-rpms           RHEL 9 for x86_64 - AppStream              activé
rhel-9-for-x86_64-baseos-rpms              RHEL 9 for x86_64 - BaseOS                 activé
rhceph-6-tools-for-rhel-9-x86_64-rpms     RH Ceph Storage Tools for RHEL 9          désactivé
openstack-17-tools-for-rhel-9-x86_64-rpms  RH OpenStack Tools for RHEL 9             désactivé
```

2.1.1 Chercher et installer des paquets

Lister les paquets

Pour afficher la liste des paquets, tapez :

```
■ dnf list
```

Cela vous affiche une très longue liste. Il peut être utile de rediriger la sortie dans un fichier :

```
■ [root@cobb ~]# dnf list > liste  
■ [root@cobb ~]# vi liste
```

Cette commande n'est cependant pas très claire. S'agit-il de la liste des paquets installés, ou la liste totale de paquets disponibles ? Pour lever tout doute, utilisez les commandes suivantes :

- `dnf list installed` : donne la liste des paquets installés sur le système.
- `dnf list available` : donne la liste des paquets disponibles pour être installés.
- `dnf list all` : donne la liste totale des paquets, installés + disponibles.

Groupes de paquets

Red Hat a réparti les paquets en groupes. En effet, au-delà des dépendances, certains logiciels sont liés entre eux pour former un ensemble ayant un but commun.

Par exemple, plutôt que d'installer soi-même chaque paquet afin de bénéficier de l'interface graphique KDE, installez simplement le groupe `Bureau KDE`.

Vous obtenez la liste des groupes installés et disponibles en tapant :

```
■ dnf grouplist
```

Remarque

L'option `-v` (pour *verbose*, *verbeux*) vous donne plus d'informations. Entre parenthèses, vous trouvez l'identifiant du groupe.

Rechercher un fichier, un paquet, afficher des informations sur un paquet

Vous pouvez chercher un mot parmi la liste des paquets disponibles et leurs descriptions, en tapant :

```
■ dnf search <mot>
```

Cela vous permet de retrouver un paquet même si vous ne disposez pas de son nom précis, ou en vous basant sur sa description.

Par exemple :

```
[root@rhel9 ~]# dnf search vpn
Mise à jour des référentiels de gestion des abonnements.
Dernière vérification de l'expiration des métadonnées effectuée
il y a 0:02:11 le lun. 05 déc. 2022 19:33:48.
===== Résumé correspond à : vpn =====
NetworkManager-libreswan.x86_64 : NetworkManager VPN plug-in for
IPsec VPN
NetworkManager-libreswan-gnome.x86_64 : NetworkManager VPN plugin
for libreswan - GNOME files
podman-gvproxy.x86_64 : Go replacement for libslirp and VPNKit
wireguard-tools.x86_64 : Fast, modern, secure VPN tunnel
```

Notez qu'il est possible de faire des recherches sur plusieurs mots en même temps.

Une fois qu'un paquet est trouvé, vous pouvez avoir plus d'informations avec la commande :

```
■ dnf info <paquet>
```

Par exemple :

```
[root@rhel9 ~]# dnf info wireguard-tools
Mise à jour des référentiels de gestion des abonnements.
Dernière vérification de l'expiration des métadonnées effectuée
il y a 0:04:42 le lun. 05 déc. 2022 19:33:48.
Paquets disponibles
Nom          : wireguard-tools
Version      : 1.0.20210914
Publication  : 2.el9
Architecture : x86_64
Taille       : 137 k
```