



Automatiser votre installation Linux avec Kickstart



Benjamin.Barras@epfl.ch, EPFL - Domaine IT - responsable de la plate-forme Unix à l'EPFL

Kickstart is an automated and unattended method, for the installation and configuration of operating systems GNU/Linux RedHat and derivatives.

Kickstart est une méthode automatisée et sans surveillance, d'installation et de configuration pour les systèmes d'exploitation GNU/Linux RedHat et dérivés.

Fiche descriptive

Anaconda (Kickstart)		
Domaine		
◆ Installation de systèmes d'exploitation Linux		
Licence	langue	version
◆ GPLv2	◆ anglais	◆ Fedora (version)
Autres alternatives libres		
◆ boot.fedoraproject.org (BFO)		
◆ Ubiquity (Debian)		
◆ AI (OpenSolaris)		
◆ FAI (fai-project.org)		
Alternatives non libres		
◆ Unattended Windows		
◆ Jumpstart (Solaris)		
Sites Web		
◆ Projet et téléchargement: fedoraproject.org/wiki/Anaconda		
Plates-formes		

Introduction

Kickstart est une méthode d'installation automatisée pour les distributions GNU/Linux RedHat¹, et dérivées (Fedora, CentOS, ScientificLinux, ...), qui permet d'installer notre système d'exploitation de manière simple et rapide. Il suffit d'indiquer à l'installateur² à l'aide d'un fichier texte, la configuration souhaitée.

Un exemple commenté servira de fil conducteur dans cet article, et on pourra toujours compléter ses connaissances sur le sujet par la suite en se basant sur la documentation officielle³ qui est très bien fournie.

Situation actuelle

Il peut paraître absurde d'utiliser à ce jour un *vieil* outil qui a plus de dix ans. Bien au contraire, c'est un bel outil et les machines virtuelles lui ont donné une seconde vie. En effet, il existe des outils⁴ pour *fabriquer* des machines virtuelles, et ils permettent l'utilisation d'un fichier kickstart. On se retrouve donc avec une machine virtuelle toute prête en moins de dix minutes, ce qui reste remarquable. De plus, lors de déploiement massif, Kickstart a un gros avantage par rapport aux déploiements classiques par image, c'est qu'il est indépendant du matériel. Vous pouvez utiliser le même fichier Kickstart sur du matériel différent, et vous retrouverez votre système d'exploitation tel que vous l'avez souhaité.

Quelques règles

Le fichier kickstart est un simple fichier texte, composé de sections, qui doivent être écrites dans l'ordre, soit:

- La section des **commandes** ou **Kickstart Options**: permet de définir les paramètres de base de votre système et contient une liste d'éléments identifiés par un mot-clé suivi de ses paramètres. Une documentation très complète concernant ces mots-clés, ou options, est fournie de la part de RedHat⁵ et de Fedora³. Cette section requiert des mots-clés *obligatoires*, et seule la documentation officielle nous permet de les connaître.
- La section **%packages**: c'est la plus difficile à paramétrer, car peu documentée, ici l'outil graphique ou un fichier existant peut nous aider à démarrer rapidement. Cette section se termine obligatoirement par **%end**⁶.
- Les sections **%pre** et **%post**: permettent d'écrire un script de pré-installation ou de post-installation, c'est la seule manière de configurer notre système de manière précise (mais nous verrons cela en détail un peu plus loin). Elles doivent également se terminer par **%end**⁶ et l'ordre entre les deux n'a pas d'importance.

¹ On peut également installer Ubuntu avec kickstart, mais je ferai un article dédié sur ce sujet

² Anaconda: fedoraproject.org/wiki/Anaconda

³ Kickstart (Fedora): fedoraproject.org/wiki/Anaconda/Kickstart

⁴ *Appliance Creator* pour Fedora Et *Virt Install* pour RedHat par exemple

⁵ Kickstart Options (RHEL 6): docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/s1-kickstart2-options.html

⁶ Sauf pour RHEL 5.x

Automatiser votre installation avec kickstart

```
# kickstart file automatically generated by anaconda.
# Version: RHEL 6.2 (Desktop 32-bit)
# C'est uniquement pour une raison de présentation, si certains mots-clés
# et leurs paramètres associés sont écrits sur plusieurs lignes.

# Première section: Paramètres
# -----
install
url --url=http://linuxline.epfl.ch/RHEL/ws-6/update2/32-bit-x86/i386
lang en_US.UTF-8
keyboard fr_CH-latin1

# Rappel: Une seule ligne
network --onboot=yes --device=eth0 --bootproto=static --hostname=ditsbpcl3.epfl.ch
--ip=128.178.1.55 --netmask=255.255.255.0 --gateway=128.178.1.1 --noipv6
--nameserver=128.178.15.8

rootpw --iscrypted $1$kdpm93n$ByTxRuyrZsYcZvHqKYE4a0
firewall --service=ssh
authconfig --enableshadow --passalgo=sha512
selinux --disabled
timezone --utc Europe/Zurich

# On efface tout et on crée une partition de 8Gb pour /, 2 Gb pour le swap
bootloader --location=mbr
clearpart --all
part / --size=8000 --fstype=ext4
part swap --size=2000
firstboot --disabled

# Pour Fedora, on ajoute le repository update
# repo --name=f16-updates --baseurl=http://mirror.switch.ch/ftp/mirror/fedora/linux/updates/16/i386/
reboot

# Deuxième section: Paquetages
# -----
%packages
@base
@client-mgmt-tools
@core
@debugging
@basic-desktop
@desktop-debugging
@desktop-platform
@directory-client
@fonts
@general-desktop
@graphical-admin-tools
@input-methods
@internet-applications
@internet-browser
@java-platform
@legacy-x
@network-file-system-client
@office-suite
@print-client
@remote-desktop-clients
@server-platform
@workstation-policy
@x11
mtools
pax
python-dmidecode
oddjob
sgpio
genisoimage
wodim
abrt-gui
certmonger
pam_krb5
krb5-workstation
gnome-pilot
libXmu
%end

# Troisième section: Post-installation
# -----
%post
%end
```

Important

- Les éléments qui ne sont pas obligatoires peuvent être ignorés.
- Un élément obligatoire manquant obligera l'installateur à poser la question à l'écran comme si vous faisiez une installation manuelle. À éviter, bien sûr, si l'on souhaite avoir une installation totalement automatique.
- Des commentaires peuvent être ajoutés à l'aide du symbole # au début d'une ligne.
- Chaque mot-clé ainsi que ses paramètres associés doivent être écrits **sur une seule ligne**.
- La documentation peut changer d'une version à l'autre. Toujours bien regarder la documentation correspondante à la version du système d'exploitation installé.

Premier fichier kickstart

Vous pouvez créer votre fichier kickstart de trois manières différentes:

- à la main: ce que l'on fait toujours après avoir récupéré un modèle;
- à l'aide d'un outil graphique: **Kickstart Configurator**;
- lorsque l'on fait une nouvelle installation, quelle que soit la méthode, notre installateur va automatiquement générer un fichier `/root/anaconda-ks.cfg`, qui contient tous les paramètres que vous avez fournis lors de votre installation, et qui se trouve sous la forme d'un fichier texte avec la syntaxe kickstart. Il suffit dès lors de l'éditer et de le modifier afin de le faire correspondre à vos besoins.

Pour notre premier exemple, récupérons un fichier kickstart créé par anaconda et qui va nous servir à installer, par exemple, une version de RedHat (voir ci-dessus).

Automatiser votre installation avec kickstart

Commentaires sur la première section

Le mot-clé `install` et sa méthode d'installation (`cdrom`, `hard-drive`, `nfs` ou `url`) doivent être sur deux lignes différentes et consécutives. La méthode d'installation permet de définir où se trouve notre source et comment y accéder (en règle générale `nfs` ou `http`). Exemples:

```
install
harddrive --partition=<part> --dir=/<dir>
# Exemple:
# harddrive --partition=hda3 --dir=/repository
```

```
install
nfs --server=<server> --dir=/<dir>
# Exemple:
# nfs --server=xxx.epfl.ch --dir=/export/fedora
```

```
install
url --url http://<server>/<dir>
ou
url --url ftp://<server>/<dir>
# Exemple:
# url --url http://mirror.switch.ch/ftp/mirror/
# fedora/linux/releases/16/Everything/i386/os
```

lang permet de définir la langue utilisée par défaut pour notre système d'exploitation. On trouvera la liste complète des paramètres dans le fichier `/usr/share/system-config-language/locale-list`.

keyboard configure le type de clavier par défaut de notre système. La liste complète des paramètres figure dans la documentation.

network configure nos périphériques réseau. Les différents périphériques sont différenciés par le paramètre `--device=`. La seule difficulté est qu'on ne connaît pas à l'avance le nom des périphériques, seul le fichier `/root/anaconda-ks.cfg` peut nous aider. Exemples:

```
# Une ligne par périphérique
network --onboot=yes --device=eth0
--bootproto=dhcp
network --onboot=no --device=wlan0 --noipv4
--noipv6
```

rootpw permet de fixer le mot de passe **root**, il est recommandé de mettre la version encryptée du mot de passe et d'utiliser l'algorithme **SHA-512** ou **SHA-256** (*Secure Hash Algorithm*) à la place de **MD5**. Exemple:

```
# SHA-256
rootpw --iscrypted $5$abcde ... dkCQbYyk2
```

firewall permet de configurer le pare-feu, il faut utiliser la documentation si l'on veut faire une configuration avancée.

selinux est utile si l'on souhaite changer la configuration par défaut.

timezone est obligatoire. Fort heureusement pour nous, nous sommes en Europe et le paramètre est toujours le même.

authconfig est riche en paramètres puisqu'il sert à configurer aussi bien l'identification que l'authentification (`ldap`, `kerberos`, ...). Il permet aussi de définir l'algorithme de hachage des mots de passe (`MD5`, `SHA-256/SHA-512`). La seule manière de bien configurer notre système est de faire sa configuration sur une machine de test déjà installée à l'aide de la commande:

```
authconfig --kickstart ...paramètres...
# Exemple (les paramètres doivent être sur une
# seule ligne):
authconfig --kickstart --enablesshadow
--passalgo=sha512 --enableldap
--ldapserver=ldap.epfl.ch --ldapbasedn=o=epfl,c=ch
--enableldapauth --enableforcelegacy
--enableldaptls --disablefingerprint
--enablemkhomedir --updateall
```

Sur les versions récentes, on peut aussi faire:

```
authconfig --savebackup=myConfig
```

avant toute manipulation, et

```
authconfig --restorebackup=myConfig
```

avant de faire d'autres essais.

Les paramètres sont nombreux, mais pas tous présents, c'est pourquoi il est parfois utile de prendre la commande ci-dessus telle quelle et de la mettre dans la partie post-installation.

Pour le partitionnement du disque, les mots-clés **bootloader** (*démarrage du système*), **clearpart** (*effacement des partitions*) et **part** (*partitionnement*) sont tous intimement liés. C'est la seule partie délicate, car vous pouvez détruire tout ce que contient votre unité de stockage. Il est recommandé de faire des essais sur une machine de test afin de bien comprendre comment se comporte l'installateur avec ces instructions. Ici aussi, la dénomination des disques n'est pas connue à l'avance et en cas de problèmes, il faut récupérer le fichier `/root/anaconda-ks.cfg` pour comprendre ce qui se passe.

Voici quelques exemples de partitionnement dont on peut avoir besoin:

```
# On réinstalle sur des partitions (sda1 & sda2)
# déjà existantes
# Elles seront reformatées, mais on ne touche pas
# les autres partitions
#
bootloader --location=mbr
part / --onpart=sda1
part swap --onpart=sda2
```

```
# On efface tout et on partitionne de manière
# automatique
#
bootloader --location=mbr
clearpart --all
part --recommended
```

```
# Si l'on ne veut pas toucher au Master Boot
# Record (mbr)
# On pourra utiliser l'option suivante
#
bootloader --location=partition
```

La documentation nous fournit ici une multitude de possibilités. Seuls l'expérience et des tests permettent de bien comprendre et de configurer correctement cette partie essentielle pour notre système d'exploitation.

firstboot permet de désactiver une série de questions lors du premier démarrage.

reboot redémarre automatiquement la machine une fois l'installation terminée.

Automatiser votre installation avec kickstart

Test

Comme nous l'avons vu dans le texte précédent, il est souvent utile de faire quelques essais avant de bien construire son fichier kickstart. Si nous avons un déploiement massif, on peut se permettre de faire une installation manuelle sur une machine afin d'en récupérer le fichier `/root/anaconda-ks.cfg`. Mais si nous avons juste une machine à installer, nous pouvons utiliser un fichier kickstart qui ne contient que le strict minimum afin de faire une installation automatique et d'en récupérer ce même fichier. On procédera de manière itérative, mais automatique, afin d'atteindre notre but. Cela nous sera toujours utile, le jour où nous devrons réinstaller la même machine pour une quelconque raison. Voici un exemple d'un fichier kickstart minimal:

```
# Version: RHEL 6.2 (Desktop 32-bit)

install
url --url=http://linuxline.epfl.ch/RHEL/ws-6/
update2/32-bit-x86/i386
lang en_US.UTF-8
keyboard fr_CH-latin1
network --bootproto=dhcp
rootpw --iscrypted $1$abcde...qwrPNHQJqJy/
authconfig --enablesshadow --passalgo=sha512
selinux --disabled
firewall --disabled
timezone --utc Europe/Zurich
bootloader --location=mbr
clearpart --all
part / --size=8000 --fstype=ext4
part swap --size=2000
firstboot --disabled
reboot

%packages
@Desktop
mtools
%end
```

Commentaires sur la deuxième section

Dans la section `%packages`, une entrée par ligne, on spécifie soit un groupe (`@`) de paquetages ou juste un nom de paquetage. La difficile lecture du fichier `repodata/comps.xml`, qui se trouve dans votre distribution, permet de connaître la liste complète des groupes et des paquetages (balises `XML`). Spécifiez les groupes avec le nom ou l'ID de groupe complet décrite dans ce fichier et les paquetages individuels par leur nom.

Pour un groupe dont le nom figure explicitement sous la rubrique `%packages`, les paquetages marqués *mandatory* (dans le fichier xml) sont toujours installés, ainsi que ceux marqués comme *default*, ceux marqués comme *optional* ne le sont pas. Si l'on souhaite modifier ces options par défaut, il suffit d'inclure ou d'exclure les paquetages. On utilise un tiret (moins) pour spécifier un paquetage ou un groupe à exclure de l'installation.

Par exemple:

```
-@base
-autofs
```

Commentaires sur la troisième section

La partie `%post` est essentiellement un script, généralement écrit en *GNU bash*⁷, mais qui peut être changé de la manière suivante:

```
%post --interpreter /usr/bin/python
# Mon script python
%end
```

L'expérience montre que la partie post-installation devient de plus en plus difficile à écrire. Cela est dû au fait que l'installeur tourne dans un environnement de plus en plus restreint et qui de plus **n'est pas votre environnement final**. Les longs scripts écrits il y a de nombreuses années pour cette partie sont donc déconseillés aujourd'hui. La solution à ce problème est simple et elle est prévue par votre système, on finalise l'installation au premier démarrage de votre machine. Cela a un avantage, c'est que l'on tourne vraiment dans l'environnement réel et définitif, mais l'inconvénient c'est que cela peut prendre du temps (par exemple pour les mises à jour). La méthode recommandée est d'utiliser un service au démarrage de la machine qui exécute un script. Ce dernier vérifie qu'il n'a pas déjà été exécuté, et si tel est bien le cas, il utilise le (long) script que vous avez écrit et qui se trouve simplement encapsulé dans ce service. Ce dernier est déjà prévu par votre système et s'appelle `rc.local`, voici le squelette d'une post-installation:

```
%post
cat <<EOF > /etc/rc.d/rc.local
#!/bin/sh
#
# On vérifie que le script n'a pas déjà été
# exécuté
if [ ! -e /root/myFile ]; then
#
# Mon script shell (à cause du #!/bin/sh ci-
# dessus)
#
# On marque que le script a été exécuté
touch /root/myFile
fi
exit 0
EOF
# On active le service
/bin/chmod +x /etc/rc.d/rc.local
%end
```

Et un exemple complet d'un script de post-installation:

```
%post
#
# rc.local
cat <<EOF > /etc/rc.d/rc.local
#!/bin/sh
#
# Si le fichier /root/post-install n'existe pas,
# on exécute le script
if [ ! -e /root/post-install ]; then
#
# On met ici les instructions post-install à
# exécuter au premier boot
#
# Log
echo `date` > /root/post-install.log
echo 'Begin post-installation script at first
boot' >> /root/post-install.log
#
# Update
yum -y -q update
#
# Clefs ssh
```

⁷ Redhat & Fedora ont utilisés busybox dans des versions précédentes (RHEL < 6 & Fedora < 14)

Automatiser votre installation avec kickstart

```
cd /root
mkdir .ssh
chmod 755 .ssh
cat <<EOF> .ssh/authorized_keys
ssh-dss AAKTW... too long ...bGK2 barras@epfl.ch
EOF
#
# Some configuration
cp /etc/profile /etc/profile.orig
cat <<EOF>> /etc/profile
export HISTTIMEFORMAT=" %d.%m.%Y-%T "
export HISTFILESIZE=1000
EOF
#
# Ajouter un utilisateur LOCAL
useradd -c "Toto Local" -d /home/toto -u 500 toto
usermod -p '$5$abcd$nfrq132Mc ... HfiiTyk2' toto
#
# Log
echo `date` >> /root/post-install.log
echo 'End post-installation script at first boot'
>> /root/post-install.log
#
# On crée le fichier /root/post-install puisque
# le script a été exécuté
touch /root/post-install
fi
exit 0
EOF
#
# On active le service
chmod +x /etc/rc.d/rc.local
%end
```

Installation avec kickstart

Le plus important est de dire à notre installeur que l'on souhaite faire une installation via kickstart. Pour cela nous devons indiquer au boot, quelle que soit la méthode, où se trouve notre fichier texte.

En règle générale, l'installation se fait depuis un CD/DVD (ISO-LINUX), un disque externe (EXTLINUX), PXE (PXELINUX), clef USB (SYSLINUX), mais dans tous les cas vous devez passer le paramètre `ks=` dans le fichier de configuration de votre amorce.

Exemple complet avec un cdrom:

```
# Répertoire de montage & monter l'image pour
# récupérer les fichiers
mkdir iso
sudo mount -o loop boot.iso ./iso
# On récupère les fichiers qui se trouvent dans
# le dossier isolinux
mkdir myIso
cp iso/isolinux/* ./myIso
# On se donne les droits d'écriture
chmod 644 myIso/isolinux.*
```

On édite le fichier `myIso/isolinux.cfg`, et on lui rajoute ces quelques lignes:

```
LABEL Install from cdrom
KERNEL vmlinuz
APPEND initrd=initrd.img ks=cdrom:/ks.cfg
LABEL Install from http
KERNEL vmlinuz
APPEND initrd=initrd.img http://xxx.epfl.ch/dossier/ks.cfg
LABEL Install from nfs
KERNEL vmlinuz
APPEND initrd=initrd.img ks=nfs:xxx.epfl.ch:/export/kickstart/ks.cfg
```

Ne pas oublier de poser le fichier `ks.cfg` dans le dossier `myIso` si l'on n'utilise pas les autres méthodes (http ou nfs). Il faut ensuite refaire l'image ISO:

```
mkisofs -o myBoot.iso -b isolinux.bin -c boot.cat \
-no-emul-boot -boot-load-size 4 \
-boot-info-table myIso
```

Quelle que soit la méthode d'amorçage choisie, vous pouvez passer ce paramètre `ks=...` au noyau (par exemple: à l'invite de démarrage), le noyau passera l'information au programme d'installation qui cherchera alors votre fichier kickstart.

Par exemple:

```
# Démarrez l'installation du système et saisissez
# la commande suivante à l'invite:
```

```
linux ks=cdrom:/ks.cfg
ou
linux ks=http://xxx.epfl.ch/dossier/ks.cfg
ou
linux ks=nfs:xxx.epfl.ch:/export/kickstart/ks.cfg
```

Vous trouverez toutes les possibilités disponibles pour le chargement du fichier kickstart dans la documentation officielle *Anaconda Boot Options*⁸ sous le mot-clé **ks**. Pour plus de détails concernant le passage de paramètres au démarrage, je vous conseille de lire mon article sur ce sujet⁹.



Article du FI-EPFL 2012 sous licence CC BY-SA 3.0 / B. Barras

⁸ Anaconda Boot Options: fedoraproject.org/wiki/Anaconda/Options

⁹ Personnaliser vos images ISO Linux: flashinformatique.epfl.ch/spip.php?article2203