

Méthode preseed pour installer Debian

Introduction

L'installateur Debian (d-i pour les intimes) est le programme qui sert à installer le système d'exploitation GNU/Linux Debian sur une machine (physique ou virtuelle) à partir d'un support d'installation qui peut être un CD, une clef USB ou une image accessible sur un réseau IP.

L'installateur Debian offre une interface utilisateur pour régler les paramètres tels que la langue du système, le partitionnement des périphériques de masse, la configuration du réseau ainsi que de choisir quels sont les groupes de paquets qui seront installés en plus du système de base. Il suffit de répondre aux questions des différentes boîtes de dialogue pour obtenir un système bien paramétré.

Mais pour déployer Debian sur une centaine de machines avec pour chacune d'entre elles une configuration adaptée on peut:

- créer des images pré-installées qui seront recopiées bits à bits sur le support de stockage (processus dit de "Ghost" de machine) mais cette tendance tend à diminuer.
- installer un système d'exploitation GNU/Linux Debian avec une configuration personnalisée et ce de manière totalement automatisée avec le mécanisme "preseed" qui est disponible dans l'installateur Debian depuis de très nombreuses années.

Utilisation de Preseed

Le principe du fichier preseed est de contenir les réponses posées en temps normal par l'installateur Debian. Celui-ci ouvre normalement tout un tas de boîtes de dialogues selon un scénario bien précis. L'intérêt du fichier preseed est de répondre à ces questions en amont. Si une boîte de dialogue dispose déjà d'une réponse, elle ne sera pas présentée à l'utilisateur. Si on travaille bien, on peut donc complètement automatiser l'installation du système sur la machine.

Le fichier preseed est un simple fichier texte qui contient des chaînes de configuration qui prennent la forme suivante:

```
d-i module/paramètre type_de_données contenu_de_la_réponse
```

- **d-i** indique qu'on s'adresse à l'installateur Debian (d-i) ou un autre programme si ce dernier utilise debconf.
- **module** indique quel est le module de l'installateur Debian concerné. Car l'installateur Debian fonctionne avec des modules. Par exemple un module s'occupe de la configuration réseau (netcfg), un autre du partitionnement (partman), etc.
- **paramètre** indique quelle est la variable qu'on souhaite pré-renseigner.
- **type_de_données** indique quel est le type de la variable concernée (ex: string pour une chaîne de caractère, toggle pour cocher une case, etc.).
- **contenu_de_la_réponse** contient la valeur affectée à la variable.

Ces chaînes reprennent la syntaxe Debconf pour ceux qui l'auraient remarqué.

Pour utiliser un fichier preseed, il existe plusieurs méthodes. La plus simple consiste à mettre dans l'initrd de l'image d'installation un fichier nommé preseed.cfg à la racine. Une autre méthode qui est du même niveau consiste à faire indiquer ce fichier par un serveur DHCP (en cas d'installation par le réseau).

Plus directement, on peut utiliser l'astuce présentée ici pour inclure le fichier preseed dans l'initrd.

Voyons maintenant quelques astuces qui permettront de voir quels modules et quelles paramètres sont utilisés dans l'installateur Debian.

Astuce n°1: Bien configurer le clavier

Cette astuce est assez simple mais on a parfois du mal à trouver les bonnes commandes pour garantir d'avoir un clavier français (azerty) configuré au redémarrage de la machine suite à l'installation. Voici donc les chaînes de configuration:

```
### Localization
# Configurer la locale permet aussi de configurer
# la langue et le pays de l'OS
d-i debian-installer/locale string fr_FR.UTF-8

# Choix du clavier
# keymap est un alias de keyboard-configuration/xkb-keymap
d-i keymap select fr(latin9)
# On désactive la sélection fine de la configuration du clavier
d-i keyboard-configuration/toggle select No toggling
```

Astuce n°2: Désactiver la détection de réseau

Parfois, on ne souhaite pas que l'installateur fasse de la détection de réseau. C'est par exemple le cas lors d'un désastre majeur qui affecte une infrastructure et notamment son réseau. Il faut parfois rebâtir un serveur de base de l'infrastructure et ce, assez rapidement, sans que ce dernier ne puisse se connecter au réseau local. Cela peut également être intéressant pour remonter une machine de zéro sans le réseau.

La configuration de ce dernier peut d'ailleurs être ajoutée plus tard lors de l'installation d'un paquet de configuration ou par simple copie de fichier à l'issue du processus d'installation.

Pour y parvenir, voici un exemple de recette :

```
### Configuration Réseau
# On désactive la configuration réseau
d-i netcfg/enable boolean false

# Mais on doit quand même configurer le nom de machine
d-i netcfg/get_hostname string goofy
```

```
# Ainsi que le domaine
d-i netcfg/get_domain string example.com

# Ici, on définit le nom local de la machine
d-i netcfg/hostname string goofy
...
# On désactive l'utilisation de NTP pour configurer l'heure (le réseau n'est
pas disponible)
d-i clock-setup/ntp boolean false
```

Astuce n°3: Partitionnement automatique

Un des points complexes de d-i est la gestion du partitionnement. En effet, on peut paramétrer totalement le programme partman de l'installateur Debian (le programme qui gère la configuration des partitions) via le fichier preseed.

Ici, on va aborder un point simple: on veut utiliser une table de partition au format GPT et tout mettre dans une seule partition.

```
### Partitionnement
# Utilisation d'une table de partition au format GPT
d-i partman-basicfilesystems/choose_label string gpt
d-i partman-basicfilesystems/default_label string gpt
d-i partman-partitioning/choose_label string gpt
d-i partman-partitioning/default_label string gpt
d-i partman/choose_label string gpt
d-i partman/default_label string gpt
partman-partitioning partman-partitioning/choose_label select gpt

# Seul le premier disque est partitionné
d-i partman-auto/disk string /dev/sda
# On partitionne en "normal": pas de RAID ni de LVM
d-i partman-auto/method string regular
# Pour être sûr, on supprime une éventuelle configuration LVM
d-i partman-lvm/device_remove_lvm boolean true
# Même chose pour le RAID
d-i partman-md/device_remove_md boolean true
# Chaînes pour ne pas toucher la configuration LVM (donc pas de
configuration)
d-i partman-lvm/confirm boolean true
d-i partman-lvm/confirm_nooverwrite boolean true

# L'installation sera simple: on fout tout dans une seule partition
# C'est ce que fait la recette atomic
d-i partman-auto/choose_recipe select atomic

# On valide sans confirmation utilisateur la configuration de partman
d-i partman-md/confirm boolean true
```

```
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true

# fstab utilisera des UUID plutôt que des noms de périphériques
d-i partman/mount_style select uuid
```

Pour un autre exemple, on va configurer un partitionnement un peu plus complexe:

- On utilise un disque dur complet (le premier)
- Sur ce volume, nous allons réaliser 4 partitions (principalement en ext4):
 - 20 Go pour l'espace système.
 - 20 Go pour les répertoires home.
 - 60 Go pour le répertoire /opt.
 - 200% de la taille de la RAM et au maximum 16Go pour le swap.

```
d-i partman-auto/expert_recipe string \
    boot-root :: \
        20480 20480 20480 ext4 \
            method{ format } format{ } \
            use_filesystem{ } filesystem{ ext4 } \
            mountpoint{ / } \
        . \
        20480 20480 20480 ext4 \
            method{ format } format{ } \
            use_filesystem{ } filesystem{ ext4 } \
            mountpoint{ /home } \
        . \
        60000 60000 60000 ext4 \
            method{ format } format{ } \
            use_filesystem{ } filesystem{ ext4 } \
            mountpoint{ /opt } \
        . \
        200% 16000 16000 linux-swaps \
            method{ swap } format{ } \
        .
```

Pour appliquer cette recette, il faut bien entendu supprimer la recette atomic du point précédent.

Ce qu'il faut retenir de cette recette:

- Tout doit tenir sur une seule ligne. C'est pourquoi on échappe tout.
- Chaque description de partition se termine par un caractère ..
- Les trois chiffres en entrée de partition correspondent à la taille minimale, la taille prioritaire et la taille maximum.
- Les tailles peuvent s'exprimer également en pourcentage de la RAM.
- **method { format }** indique qu'on souhaite formater la partition.
- **format { }** effectue le formatage.
- **use_filesystem { }** indique qu'on souhaite utiliser cette partition dans le futur système d'exploitation.

- **mountpoint { xxx }** permet de préciser le point de montage de la partition.
- **method { swap }** permet d'indiquer qu'on utilise un swap.

On peut bien sûr aller plus loin et créer des volumes RAID ou LVM et spécifier comment faire avec.

Vous pouvez lire la documentation de l'installateur Debian sur ce sujet: `partman-auto-recipe.txt` et `partman-auto-raid-recipe.txt`.

Astuce n°4: Désinstaller des paquets non indispensables

Parfois, l'installation de base de Debian peut ne pas convenir. Par exemple, en ce qui concerne l'éditeur de texte par défaut. Ce dernier est `vim.tiny` ce qui ne correspond pas du tout à ce que j'installe de facto sur mes machines (je suis un sale emacs fanboy). On pourrait également avoir le même raisonnement avec le MTA (`Exim4`) qui pour certaines configuration peut paraître "overkill" (on peut le remplacer par un MTA plus léger comme `msmtp` par exemple).

Voici un exemple de suppression de paquets dans un fichier `preseed`:

```
## Suppression de certains paquets
d-i preseed/late_command string \
    in-target apt-get -y purge nano; \
    in-target apt-get -y purge vim-tiny; \
    in-target apt-get -y purge vim-common
```

Le principe est assez simple à comprendre: on lance une commande terminale qui va se charger de supprimer ces paquets via le classique `apt-get`. Cette commande est unique et c'est pour ça qu'on est obligé de mettre une seule ligne qu'on peut bien entendu "échapper". Dans le cas présent, le `in-target` indique qu'on joue la commande dans le système installé (et non dans l'environnement de l'installateur Debian).

Les paquets de la catégorie "essential" sont tous installés par défaut. Cela signifie que dans notre cas, l'installateur va commencer par installer tous ces paquets et qu'il les désinstallera lorsqu'il traitera la chaîne `preseed/late_command`. Ce n'est pas très élégant mais c'est comme ça que ça marche. Dans la pratique, on enlève quand même assez peu de paquets "essential".

Astuce n°5: Configurer mes dépôts de paquets APT

Depuis l'avènement d'`httpredir`, ce point n'est pas forcément très intéressant mais si on a des dépôts Debian personnalisés on peut les incorporer directement via l'installateur Debian. Voici comment y parvenir...

Le premier point est de définir un miroir Debian. Il semble que ce soit nécessaire pour que l'installateur Debian ne pose pas de question sur ce sujet. Il existe un certain nombre de chaînes de configuration pour cela.

Ensuite, on indique qu'on ne souhaite pas utiliser de miroir. La configuration pré-existante n'est donc pas injectée dans `/etc/apt/sources.list`. Dans un troisième temps, on indique qu'on souhaite activer les

services security et updates. Les adresses de ces services sont hardcodées (security.debian.org). Enfin, on peut procéder à l'ajout du dépôt local avec une chaîne de configuration. On peut en ajouter autant qu'on veut et même y glisser des commentaires.

```
### Paramètres de miroir
d-i mirror/country string manual
d-i mirror/http/hostname string httpredir.debian.org
d-i mirror/http/directory string /debian
d-i mirror/http/proxy string

### Configuration d'apt
# On indique qu'on ne souhaite pas utiliser de miroir
d-i apt-setup/use_mirror boolean false
d-i apt-setup/no_mirror boolean true
# On active les services security et updates
d-i apt-setup/services-select multiselect security, updates
d-i apt-setup/security_host string security.debian.org
# Notez qu'on ne peut pas modifier la chaîne pour updates.
# Ensuite, on indique quel est notre dépôt local (le premier qui démarre à 0)
d-i apt-setup/local0/repository string \
    http://debian.example.com/debian stable main contrib
# On peut même y mettre un commentaire qui sera injecté dans le fichier
sources.list
d-i apt-setup/local0/comment string Serveur de paquets locaux de
l'organisation
```

On peut tester plusieurs valeurs et ajouter d'autres dépôts (les backports par exemple).

Automatisation du processus d'installation

Exemple de fichier de preseed qui automatise complètement le processus d'installation

L'installation est complètement automatique, sans aucune interaction de l'administrateur (hors manipulations pour booter sur l'image d'installation). Ce serveur est destiné à être remonté "from scratch" à partir de la simple clef USB qui contient également les paquets à installer et la configuration des principaux logiciels utilisés ainsi que la configuration système.

```
#### Fichier de preseed pour Debian jessie
# Configurer la locale permet aussi de configurer
# la langue et le pays de l'OS
d-i debian-installer/locale string fr_FR.UTF-8

# Choix du clavier
# keymap est un alias de keyboard-configuration/xkb-keymap
d-i keymap select fr(latin9)
# On désactive la sélection fine de la configuration du clavier
```

```
d-i keyboard-configuration/toggle select No toggling

### Configuration Réseau
# On désactive la configuration réseau
d-i netcfg/enable boolean false

# Mais on doit quand même configurer le nom de machine
d-i netcfg/get_hostname string testjessie
# Ainsi que le domaine
d-i netcfg/get_domain string example.com

# Ici, on définit le nom local de la machine
d-i netcfg/hostname string goofy

### Mirror settings
# Do not configure mirror repository
#d-i apt-setup/no_mirror boolean true
d-i mirror/country string manual
d-i mirror/http/hostname string debian.proxad.net
d-i mirror/http/directory string /debian
d-i mirror/http/proxy string

### Configuration des comptes
# On indique le mot de passe root, sous forme d'un hash MD5 (echo
"mypassword" | mkpasswd -s -H MD5)
d-i passwd/root-password-crypted password xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

# On ne souhaite pas créer d'autres utilisateurs
d-i passwd/make-user boolean false

### Gestion de l'horloge
# L'horloge matérielle fonctionne sur la zone UTC
d-i clock-setup/utc boolean true

# La zone utilisée par l'OS sera celle de Paris
d-i time/zone string Europe/Paris

# On désactive l'utilisation de NTP pour configurer l'heure
d-i clock-setup/ntp boolean false

### Partitioning
# Nous voulons une table de partition au format GPT
d-i partman-basicfilesystems/choose_label string gpt
d-i partman-basicfilesystems/default_label string gpt
d-i partman-partitioning/choose_label string gpt
d-i partman-partitioning/default_label string gpt
d-i partman/choose_label string gpt
d-i partman/default_label string gpt
partman-partitioning partman-partitioning/choose_label select gpt

# Seul le premier disque est partitionné
```

```
d-i partman-auto/disk string /dev/sda
# On partitionne en "normal": pas de RAID ni de LVM
d-i partman-auto/method string regular
# Pour être sûr, on supprime une éventuelle configuration LVM
d-i partman-lvm/device_remove_lvm boolean true
# Même chose pour le RAID
d-i partman-md/device_remove_md boolean true
# Chaînes pour ne pas toucher la configuration LVM (donc pas de
configuration)
d-i partman-lvm/confirm boolean true
d-i partman-lvm/confirm_nooverwrite boolean true

# L'installation sera simple: on fout tout dans une seule partition
d-i partman-auto/choose_recipe select atomic

# On valide sans confirmation la configuration de partman
d-i partman-md/confirm boolean true
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true

# fstab utilisera des UUID plutôt que des noms de périphériques
d-i partman/mount_style select uuid

### Installation du système
# On ne souhaite pas installer les paquets recommandés
# L'installation sera limitée aux paquets "essentials"
d-i base-installer/install-recommends boolean false

### Configuration d'apt
# On indique qu'on ne souhaite pas utiliser de miroir
d-i apt-setup/use_mirror boolean false
d-i apt-setup/no_mirror boolean true
# On active les services security et updates
d-i apt-setup/services-select multiselect security, updates
d-i apt-setup/security_host string security.debian.org
# Notez qu'on ne peut pas modifier la chaîne pour updates.
# Ensuite, on indique quel est notre dépôt local (le premier qui démarre à
0)
d-i apt-setup/local0/repository string \
    http://debian.proxad.net/debian stable main contrib
# On peut même y mettre un commentaire qui sera injecté dans le fichier
sources.list
d-i apt-setup/local0/comment string Serveur principal

# Pas d'envoi de rapport vers popcon
popularity-contest popularity-contest/participate boolean false

### Configuration de GRUB
# Seul Debian sera géré par GRUB
```

```
d-i grub-installer/only_debian boolean true

# Si on détecte un autre OS, on installera GRUB sur le MBR
d-i grub-installer/with_other_os boolean true

# On installe GRUB sur /dev/sda
d-i grub-installer/bootdev string /dev/sda

## Suppression de paquets non désirés
d-i preseed/late_command string \
    in-target apt-get -y purge nano; \
    in-target apt-get -y purge vim-tiny; \
    in-target apt-get -y purge vim-common

### Fin de l'installation
# Désactivation du message indiquant que l'installation est terminée.
d-i finish-install/reboot_in_progress note

# Pas d'éjection du média d'installation (bien pour faire des tests sur une VM)
d-i cdrom-detect/eject boolean true

# Une fois l'installation terminée, on éteint la machine
d-i debian-installer/exit/poweroff boolean true
```

Les .templates de debconf

Le guide d'installation de Debian donne un exemple assez complet de fichier preseed avec des explications dans les commentaires pour avoir une bonne idée de la majorité des chaînes de configuration. Néanmoins, il est parfois obligatoire d'utiliser une option peu courante ou non documentée.

La recherche de ces informations peut parfois être complexe. Le plus simple consiste à rechercher les fichiers `.templates` de `debconf` pour le module `d-i` que l'on souhaite explorer. Dans ces fichiers, on trouvera le nom des variables et des paramètres ainsi que le texte des questions posées.

Pour cela on peut utiliser `Codesearch`. Par exemple, pour avoir plus d'informations sur les possibles questions sur le module `apt-setup`. Faire une recherche avec les bons paramètres sur `Codesearch` : `path:/debian/* apt-setup`. On trouvera tous les paquets qui contiennent la chaîne `apt-setup` dans le répertoire `debian`. Dans la liste des paquets, on trouve le paquet `apt-setup`. Ce dernier semble être le bon candidat car il est écrit dans sa description qu'il s'agit d'un module de `d-i`.

Le fichier `apt-mirror-templates` du répertoire `debian`, fournit l'ensemble des chaînes de configuration disponibles pour cette partie de module.

Industrialisation du processus

On peut utiliser des programmes qui permettront d'industrialiser vos déploiements avec les fichiers preseed et même plus.

simple-cdd

Permet de construire des images ISO hybrides (bootables à partir de clef USB) à partir de simples fichiers de configuration. On peut ainsi gérer autant d'images ISO que l'on veut selon la configuration du système et les paquets que vous souhaitez utiliser. L'intérêt de simple-cdd est que l'image construite contient tous les paquets indispensables à l'installation de la machine. On peut même y ajouter des paquets que l'on a fabriqué. En une simple image ISO (ou une clef USB), On peut donc reconstruire une machine "from scratch" et même sans accès au réseau local ou à Internet. Pour information, simple-cdd est un simple script shell qui est une surcouche à debian-cd. Ce dernier est le programme qui est utilisé pour fabriquer les images officielles de Debian et il est assez complexe à utiliser. Avec simple-cdd, quelques fichiers de configuration suffisent.

Les outils de config-package-dev

Permettent de fabriquer un ou plusieurs paquets Debian dits "de configuration". Le principe est de mettre l'ensemble des fichiers de configuration spécifiques dans un répertoire de source et de créer un paquet Debian avec les outils config-package-dev.

Par exemple, on peut faire un paquet qui installe le fichier /etc/network/interface spécifique à une machine. L'installation de ce dernier sera gérée sous Debian. De plus, lorsqu'on désinstalle ce paquet de configuration, cette dernière reviendra à l'état initial (c'est-à-dire selon la configuration par défaut du mainteneur Debian ou la configuration customisée avant).

Le processus de déploiement est le suivant :

1. Production d'un paquet qui contient toute la configuration spécifique de la machine (réseau/configuration de grub/configuration du MTA système/netfilter/anonymisation IPv6/ajout de cacet dans les autorités de confiance/etc.).
2. Embarquement de ce paquet dans la configuration simple-cdd de la machine
3. A l'issue de l'installation, on obtiens une machine prête à fonctionner et complètement configurée.

Bien entendu, rien n'empêche de mettre ce paquet sur un dépôt local et d'orchestrer la configuration des machines avec. On peut même gérer ce paquet avec un gestionnaire de version pour pouvoir facilement revenir en arrière.

Conclusions

Avec la méthode preseed, on peut donc installer Debian de manière complètement automatisée tout en bénéficiant de toutes les options présentes dans les boîtes de dialogues de l'installateur Debian. La plus grande difficulté sera de trouver les chaînes qui permettront de bien configurer la machine comme on le souhaite.

Attention, certains éléments ne sont pas gérés par l'installateur Debian et il faudra trouver une autre solution. Par exemple, lorsqu'on utilise la configuration réseau via le mécanisme `/etc/network`, on ne peut pas indiquer des scripts à lancer lors du lancement de l'interface (`pre-up.d`) avec l'installateur Debian. Il faudra un paquet de configuration ou utiliser un programme d'orchestration de configuration (Chef, Puppet, Ansible, Propellor, etc.)

Cette méthode d'installation automatisée présente l'avantage d'utiliser uniquement des outils Debian et reste sans doute moins complexe que des logiciels d'orchestration de configuration classiques qui imposent de mettre en place toute une infrastructure pour ça et qui ne règlent pas le problème de l'installation initiale. C'est assez intéressant lorsqu'on gère des serveurs physiques hétérogènes notamment car ces derniers ne sont pas clonables facilement.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=debian:debian-preseed-overview>

Last update: **2025/02/19 10:59**

