

config.sinerj.org

Installation de FAI

Fred m'a installé une Debian Etch toute neuve. Je m'y connecte en tant que root. J'y ajoute le dépôt officiel de FullyAutomaticInstallation? dans /etc/apt/sources.list

```
deb http://www.informatik.uni-koeln.de/fai/download etch koeln
```

Je fais un petit

```
apt-get update
```

Et je vois que les paquets fai sont bien présents

```
:/#apt-cache search fai-
```

```
fai-client - Fully Automatic Installation client package
fai-doc - Documentation for FAI
fai-nfsroot - Fully Automatic Installation nfsroot package
fai-quickstart - Fully Automatic Installation quickstart package
fai-server - Fully Automatic Installation server package
fai-kernels - special kernels for FAI (Fully Automatic Installation)
```

Allez, on installe un serveur fai, en version 3.2.4 à ce jour

```
apt-get install fai-server
```

Configuration de FAI

Après, on va modifier les paramètres de fai

```
vi /etc/fai/NFSROOT
```

On indique que certains paquets doivent être installés dans la NFSROOT en les ajoutant au premier paragraphe de /etc/fai/NFSROOT

```
PACKAGES aptitude
nfs-common fai-nfsroot module-init-tools dhcp3-client ssh rdate lshw portmap
bootpc rsync lftp less dump reiserfsprogs ext2resize usbutils
hwinformisc pciutils hdparm smartmontools parted mdadm lvm2
dnsutils ntpdate dosfstools cvs jove xfsprogs xfsdump
sysutils dialog discover mdetect
console-tools console-common
expect iproute udev subversion
live-initramfs
linux-image-generic
python
libnet-perl grub libmail-sender-perl
```

Ensuite, on indique à FAI les dépôts à utiliser

```
vi /etc/fai/apt/sources.list
```

Dans notre cas, on choisit d'installer des Ubuntu Gutsy et on inclut également le depot pour l'installation de [BCFG2]? et celui de FAI car on va y retrouver live-initramfs. Voici la tete que doit avoir sources.list apres ca

```
deb ftp://ftp.mcs.anl.gov/pub/bcfg/ubuntu gutsy/  
deb http://fr.archive.ubuntu.com/ubuntu/ gutsy main restricted universe multiverse  
deb http://fr.archive.ubuntu.com/ubuntu/ gutsy-security main restricted universe multiverse  
deb-src http://fr.archive.ubuntu.com/ubuntu/ gutsy-security main restricted universe multiverse  
deb http://www.informatik.uni-koeln.de/fai/download etch koeln
```

Il faut aussi indiquer que la procedure de demarrage de l'installation automatique doit etre celle qui correspond a Ubuntu Gutsy et non a Debian Etch

```
vi /etc/fai/make-fai-nfsroot.conf
```

On y remplace une ligne par une autre

```
#FAI_DEBOOTSTRAP="etch http://ftp.debian.org/debian"  
FAI_DEBOOTSTRAP="gutsy http://fr.archive.ubuntu.com/ubuntu"
```

Et il manque le script de demarrage correspondant a gutsy. Il faut aller le prendre sur une gutsy bien a jour et le copier dans le meme chemin sur le serveur d'installation que l'on est en train de configurer. Le chemin du script en question est

```
/usr/lib/debootstrap/scripts/gutsy
```

Le probleme, c'est que ce script fait appel a des fonctions de debootstrap (notamment default_mirror) qui ne sont pas disponibles sur notre hote etch. Donc, il faut virer les 8 premieres lignes de ce script gutsy (ces 8 premieres lignes sont un bloc "case") de maniere a ce qu'il commence maintenant par les lignes suivantes:

```
mirror_style release  
download_style apt  
findebs_style from-indices
```

Ensuite, on creee un utilisateur par lequel les machines en cours d'installation (clients FAI) se connecteront au serveur pour uploader ses fichiers de log en fin d'operation

```
adduser fai
```

Et il faut indiquer a FAI que c'est cet utilisateur qui doit etre utilise pour les uploads de fichiers de logs

```
vi /etc/fai/fai.conf
```

Voici les deux paragraphes qu'il faut mettre a jour. L'un pour preciser que fai est l'utilisateur en question et l'autre pour preciser que le protocole de communication a utiliser sera ssh

```
# LOGUSER: an account on the install server which saves all log-files  
# and which can change the kernel that is booted via network.  
# Configure .rhosts for this account and PAM, so that root can log in  
# from all install clients without password. This account should have  
# write permissions for /srv/tftp/fai. For example, you can use write  
# permissions for the group linuxadm. chgrp linuxadm /srv/tftp/fai;chmod  
# g+w /srv/tftp/fai. If the variable is undefined, this feature is disabled.  
# Define it, to enable it, eg. LOGUSER=fai  
LOGUSER=  
LOGSERVER=$FAISERVER
```

```
# set protocol type for saving logs. Values: ssh, rsh, ftp
FAI_LOGPROTO=ssh
```

Et, toujours dans le meme fichier, il faut indiquer l'URL pour se connecter en NFS au serveur de configuration FAI

```
# how to access the fai config space
# default if undefined here: nfs://`hostname`/$FAI_CONFIGDIR
# supported URL-types: nfs, file, cvs, cvs+ssh, svn+file, svn+http,...
#FAI_CONFIG_SRC=nfs://yourservername$FAI_CONFIGDIR
FAI_CONFIG_SRC=nfs://config.sinerj.org$FAI_CONFIGDIR
```

Le parametage du serveur FAI est pret. Il ne reste plus, en theorie, qu'a generer les fichiers qui permettront l'installation automatique avec un petit

```
fai-setup -v
```

MAIS... comme vous avez le malheur d'installer FAI dans un vserver, fai-setup va planter au moment ou il utilise la commande debootstrap. Voici comment je contourne le probleme. Sur le serveur physique qui heberge votre vserver, votre gentil administrateur systeme va lancer un

```
debootstrap gutsy /srv/fai/nfsroot/live/filesystem.dir http://fr.archive.ubuntu.com/ubuntu
```

Ensuite, il faut editer /usr/sbin/make-fai-nfsroot et mettre en commentaire la derniere ligne de la fonction call_debootstrap(), celle qui commence par 'yes' de maniere a ce que debootstrap ne soit plus lance depuis le vserver

```
call_debootstrap() {
    local dversion=$(dpkg -l debootstrap | grep debootstrap | cut -f7 -d' ')
    echo "Creating base system using debootstrap version $dversion"
    echo "Calling debootstrap $1 $NFSROOT $2"
    # yes '' | LC_ALL=C call_verbos debootstrap $FAI_DEBOOTSTRAP_OPTS $1 $NFSROOT $2
}
```

Toujours dans le meme fichier, il faut desactiver tout le bloc qui supprime les jolies choses que la commande debootstrap a cree dans votre vserver. Pour cela, mettez la ligne if en commentaire et remplacez-la par un test `1 = 2`

```
# if [ -d $NFSROOT/proc -a ! "$recover" ]
if [ false ]
then
    echo $NFSROOT already exists. Removing $NFSROOT
    umount $NFSROOT/dev/pts 1>/dev/null 2>&1 || true
    [ -L $NFSROOT/proc/self ] && umount $NFSROOT/proc || true
    [ -L $NFSROOT/proc/self ] && die "/proc is still mounted inside the nfsroot."
    rm -rf $oldnfsroot/.??* $oldnfsroot/*
    # also remove files $NFSROOT/.? but not . and ..
    find $oldnfsroot -xdev -maxdepth 1 ! -type d | xargs -r rm -f
fi
```

Ce n'est qu'ensuite que vous pouvez obtenir de bons resultats en lançant la commande

```
fai-setup -v
```

Au passage, comme je note que perl se plaint a plusieurs reprises de l'absence de configuration de "locale", on va installer et configurer locale

```
apt-get install locales
```

puis

```
echo "fr_FR.UTF-8 UTF-8" > /etc/locale.gen
```

et

```
locale-gen
```

enfin

```
echo LANG="fr_FR.UTF-8" > /etc/environment
```

Avec ca, perl ne devrait plus avoir a se plaindre ?

Il manque encore un truc: il faut installer NFS pour que les machines a installer puissent se connecter au serveur d'installation et y telecharger leurs configurations. Donc

```
apt-get install nfs-user-server
```

Alors, quand on en arrive la, on se dit que tout va bien. Mais non. Comme on est dans un vserver, les choses ne sont jamais aussi simples que dans la vraie vie. Alors on fait appel a Super-Fred, qui regle la situation d'un petit coup de baguette magique pour que fai-setup tourne comme il faut.

Ensuite, on reprend comme si de rien n'etait. Fred a indique qu'il manque un morceau alors on fait un scp depuis une debian ou une ubuntu qui a un noyau qui manque au vserver

```
scp /boot/initrd.img-2.6.22-14-generic root@config.sinerj.org:/srv/fai/nfsroot/live/filesystem.di
```

Maintenant, on va creer l'image ISO depuis lesquelles les machines pourront booter. Pour cela, une premiere modif pour faire un bon menu de demarrage

```
vi /etc/fai/menu.lst
```

On y ajoute, en premier menu, le menu suivant (qui contient l'adresse IP du serveur d'installation)

```
title Fully Automatic Installation - sinerj (pw: installme)
password installme
kernel /boot/vmlinuz boot=live FAI_FLAGS="verbose,createvt" FAI_ACTION=install nfsroot=80.67.172
initrd /boot/initrd.img
```

Ensuite, il manque encore quelques paquets (le seul vraiment utile est peut-etre mkisofs mais je suis pas sur...)

```
apt-get install live-initramfs mknbi mkisofs
```

Maintenant, on est prêts a fabriquer l'image ISO du CD de boot

```
mkdir /srv/fai/mirror
fai-cd -m /srv/fai/mirror /srv/fai/faicd.iso
```

On pourra graver cette image ou bien la monter pour faire booter une machine virtuelle.

Configuration des machines a installer

Maintenant, il s'agit d'indiquer a FAI la configuration que les machines a installer devront adopter ainsi que diverses etapes specifiques de cette installation.