

FAI Guide (Fully Automatic Installation)

Contents

1	Abstrait	1
2	Introduction	2
2.1	Disponibilité	2
2.2	Motivation	2
2.3	Comment fonctionne FAI	3
2.4	Caractéristiques	3
2.5	Le temps de l'installation	4
3	Quickstart - Pour l'utilisateur impatient	4
3.1	Ma première installation	4
3.2	Mon premier serveur d'installation	4
4	Vue d'ensemble et concepts	5
4.1	Conditions Générales	5
4.2	Le concept de classe	7
5	Configurer votre faiserver	7
5.1	Installer les paquetages FAI	8
5.2	Créer le nfsroot	8
5.3	Création de l'espace de configuration	9
5.4	Configurer les démons réseau	9
5.4.1	Configuration du démon DHCP	9
5.4.2	Ajout d'une entrée d'hôte au DHCP	10
5.4.3	TFTP	10
5.4.4	NFS	10
5.5	Création de la configuration PXELINUX	10
5.6	Serveur personnalisé	11
6	Planifiez votre installation	11
6.1	L'espace de configuration et ses sous-répertoires	12
6.2	Définition des classes	13
6.3	Définition des Variables	13
6.4	Configuration du disque dur	14
6.5	Extraction du fichier de base	14
6.6	Debconf préconfiguration	14
6.7	L'Accès au dépôt de paquetages	14
6.8	configuration du progiciel	14

6.9	Scripts de personnalisation	16
6.9.1	Scripts shell	16
6.9.2	Scripts cfengine	16
6.10	Hooks	16
6.11	FAI flags	17
7	FAI installe votre planification	18
7.1	La première partie d'une installation	18
7.2	Messages de boot	18
7.3	Redémarrage de l'ordinateur dans le nouveau système	19
7.4	Démarrage de FAI (tâche confdir)	20
7.5	Définition de classes et de variables (tâches defclass et defvar)	20
7.6	Partitionnement de disques locaux, création de systèmes de fichiers (tâches de partitionnement)	20
7.7	Préréglage Debconf (tâche debconf)	20
7.8	Installation de progiciels (tâche instsoft)	20
7.9	Personnalisation spécifique au site (task configure)	20
7.10	Enregistrement des fichiers journaux (tâche savelog)	21
7.11	Redémarrez le nouveau système installé	21
8	Chapitre avancés de FAI	21
8.1	Vérification des paramètres reçus des serveurs DHCP	21
8.2	Surveillance de plusieurs installations clientes	21
8.3	Collecte d'adresses Ethernet pour plusieurs hôtes	22
8.3.1	Débogage du trafic réseau	22
8.4	Détails du démarrage PXE	22
8.5	Personnalisation de la configuration de votre serveur d'installation	23
8.6	Création d'un CD FAI ou d'une clé USB	23
8.7	Création d'images de disque VM à l'aide de FAI	24
8.8	Système de sauvetage FAI	24
8.9	FAI sans NFS	25
8.10	Installation d'autres distributions à l'aide d'un nfsroot Debian	25
8.11	Création d'environnements chrooter et virtualiser	25
8.12	Utilisation de FAI pour les mises à jour	26
8.13	Comment installer un système d'exploitation 32 bits à partir d'un système d'exploitation 64 bits	27

9	Divers conseils et détails	27
9.1	La liste des tâches	27
9.2	Tests automatisés	29
9.3	Découvrir automatiquement	29
9.4	Modification du périphérique d'amorçage	29
9.5	Comment créer un miroir Debian local	30
9.6	Petits conseils	30
9.7	flag_reboot (FAI_FLAGS)	31
9.8	CentOS reboot	31
9.9	Fichiers journaux	31
9.10	Comment utiliser HTTP pour le démarrage PXE	32
10	Dépannage	32
10.1	Erreurs d'amorçage	32

1 Abstrait

FAI est un système non interactif permettant d'installer, de personnaliser et de gérer les configurations de systèmes et de logiciels Linux sur les ordinateurs ainsi que sur les machines virtuelles et les environnements chroot, des petits réseaux aux grandes infrastructures et clusters.

Ce manuel décrit le logiciel d'installation entièrement automatique. Cela inclut l'installation des paquets, la configuration du serveur, la création de la configuration et la gestion des erreurs.

```
+-----+
| This manual describes FAI 5.3 but most things are also valid for 4.x. |
+-----+
```

(c) 2000-2017 Thomas Lange

Copyright Ce manuel est un logiciel libre; Vous pouvez le redistribuer et / ou le modifier selon les termes de la Licence Publique Générale GNU publiée par la Free Software Foundation; Soit la version 2, soit (à votre choix) toute version ultérieure.

Ceci est distribué dans l'espoir qu'il sera utile, mais sans aucune garantie ; Sans même la garantie implicite de qualité marchande ou d'adaptation à un usage particulier. Pour plus de détails, consultez la GNU General Public License.<

Une copie de la GNU General Public License est disponible sous la forme /usr/share/common-licenses/GPL dans la distribution Debian GNU/Linux ou sur le World Wide Web sur le site GNU Vous pouvez également l'obtenir en écrivant à la Free Software Foundation , Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, États-Unis.

2 Introduction

2.1 Disponibilité

Page d'accueil

<http://fai-project.org>

FAI wiki

<http://wiki.fai-project.org>

Téléchargement

<http://fai-project.org/download>

Entrée pour *sources.list*

```
deb http://fai-project.org/download jessie koeln
```

Pages du manuel

<http://fai-project.org/doc/man/>

Liste de diffusion

<https://lists.uni-koeln.de/mailman/listinfo/linux-fai>

Retour d'information

Envoyez vos commentaires et vos commentaires à fai@fai-project.org à la liste de diffusion .

Bogues

Utiliser le système de suivi des bogues Debian (BTS) <http://bugs.debian.org>

Changements visibles par l'utilisateur

<http://fai-project.org/NEWS>

Arbre source via git

```
git clone git://github.com/faiproject/fai.git
```

Voir l'arbre source avec http

<https://github.com/faiproject/fai>

Les pages man incluent toujours des informations à jour et beaucoup de détails sur toutes les commandes FAI. Alors, n'oubliez pas de les lire attentivement. Lisez maintenant ce manuel, puis profitez de l'installation entièrement automatique et de votre temps économisé.

2.2 Motivation

Avez-vous déjà effectué des installations identiques d'un système d'exploitation à plusieurs reprises? Souhaitez-vous être en mesure d'installer un cluster Linux avec des dizaines de nœuds à lui seul?

Répéter la même tâche encore et encore est ennuyeux - et conduira certainement à des erreurs. Aussi beaucoup de temps pourrait être sauvé si les installations ont été faites automatiquement. Un processus d'installation avec interaction manuelle n'est pas à l'échelle. Mais les grappes ont l'habitude de croître au fil des ans. Pensez à long terme plutôt que de planifier quelques mois dans l'avenir.

En 1999, j'ai dû effectuer une installation d'un cluster Linux avec un serveur et 16 clients. Puisque j'ai eu beaucoup d'expérience en faisant des installations automatiques des systèmes d'exploitation de Solaris sur le matériel de SUN SPARC, l'idée de construire une installation automatique pour Debian est née. Solaris dispose d'une fonctionnalité d'installation automatique appelée JumpStart ¹. En conjonction avec les scripts d'auto-installation de Casper Dik ², Je pourrais sauver beaucoup de temps non seulement pour chaque nouvel ordinateur de SUN, mais aussi pour la réinstallation des postes de travail existants. Par exemple, j'ai dû construire un LAN temporaire avec quatre stations de travail SUN pour une conférence, qui a duré seulement quelques

¹ Solaris 8 Advanced Installation Guide at <https://docs.oracle.com/cd/E19455-01/806-0957/806-0957.pdf>

² <http://www.science.uva.nl/pub/solaris/auto-install>

jours. J'ai retiré ces postes de travail de notre réseau de recherche habituel et mis en place une nouvelle installation pour la conférence. Quand il était terminé, j'ai simplement intégré les postes de travail dans le réseau de recherche, redémarré une seule fois, et après une demi-heure, tout était opérationnel comme avant. La configuration de tous les postes de travail était exactement la même qu'avant la conférence, car tout était effectué par le même processus d'installation. J'ai également utilisé l'installation automatique pour réinstaller un poste de travail après un disque dur endommagé avait été remplacé. Il m'a fallu deux semaines pour recevoir le nouveau disque dur, mais seulement quelques minutes après l'installation du nouveau disque, le poste de travail fonctionnait comme avant. Et c'est pourquoi j'ai choisi d'adapter cette technique à un cluster de PC sous Linux.

2.3 Comment fonctionne FAI

Le client d'installation qui sera installé à l'aide de FAI, est démarré via une carte réseau ou à partir d'un CD ou d'une clé USB. Il obtient une adresse IP et démarre un noyau Linux qui monte son système de fichiers racine via NFS (nfsroot) du serveur d'installation. Une fois le noyau démarré, le script de démarrage FAI exécute l'installation automatique qui n'a pas besoin d'interaction. Tout d'abord, les disques durs seront partitionnés, les systèmes de fichiers seront créés et des progiciels seront ensuite installés. Après cela, le nouveau système d'exploitation installé est configuré selon vos besoins locaux en utilisant quelques scripts. Enfin, le nouveau système d'exploitation sera démarré à partir du disque local.

Les détails sur la façon d'installer l'ordinateur (la configuration) sont stockés dans l'espace de configuration du serveur d'installation. Les fichiers de configuration sont partagés entre des groupes d'ordinateurs s'ils sont similaires en utilisant le concept de classe. Vous n'avez donc pas besoin de créer une configuration pour chaque nouvel hôte. Par conséquent, FAI est une méthode évolutive pour installer un gros cluster avec un grand nombre de nœuds même si leur configuration n'est pas identique.

FAI peut également être utilisé comme un système de sauvetage ou pour l'inventaire matériel. Vous pouvez démarrer votre ordinateur, mais il n'effectuera pas une installation. Au lieu de cela, il exécutera un Debian GNU / Linux entièrement fonctionnel sans utiliser les disques durs locaux. Ensuite, vous pouvez effectuer une connexion à distance et sauvegarder ou restaurer une partition de disque, vérifier un système de fichiers, inspecter le matériel ou effectuer toute autre tâche.

2.4 Caractéristiques

- Une installation entièrement automatisée peut être effectuée.
- Très rapide installation sans surveillance.
- Système flexible grâce à un concept de classe simple.
- Mise à jour des systèmes en cours d'exécution sans réinstallation.
- Création facile d'un environnement de virtualisation ou d'un chroot
- Les hôtes peuvent démarrer à partir d'une carte réseau, d'un CD, d'une clé USB.
- Création simple d'un CD d'installation ou d'une clé USB.
- PXE avec la méthode de démarrage DHCP est pris en charge.
- ReiserFS, ext3 / ext4, btrfs et support de système de fichiers XFS.
- Support logiciel RAID et LVM.
- Détection automatique du matériel.
- Vous pouvez déployer Debian, Ubuntu, CentOS, SuSE, Scientific Linux
- Connexion à distance via ssh lors du processus d'installation possible.
- Toutes les configurations similaires sont partagées entre tous les clients d'installation.
- Les fichiers journaux de toutes les installations sont enregistrés sur le serveur d'installation.
- Les scripts Shell, Perl, Python, Ruby, expect et CFEngine sont pris en charge lors de l'étape de personnalisation.
- Prise en charge de nombreux protocoles comme NFS, FTP, HTTP, git

- Peut être utilisé comme un système de sauvetage et pour l'inventaire matériel.
- Prise en charge du client sans disque.
- Ajoutez facilement vos propres fonctions via des hooks ou modifiez le comportement par défaut.
- Clonage de machines utilisant des images de disque est pris en charge

2.5 Le temps de l'installation

Le temps d'installation est déterminé par la quantité de logiciel et la vitesse du disque dur. Voici quelques exemples de temps. Tous les clients d'installation avaient une carte réseau 1Gbit installée.

CPU	RAM	Disk	Software installed	time
i7-3770T 2.50GHz	8GB	SSD	6 GB software	8.5 min
Core-i7 3.2GHz	6GB	SATA disk	4.3GB software	7 min
Core-i7 3.2GHz	6GB	SATA disk	471 MB software	77sec
Intel Core2 Duo	2GB	SATA disk	3 GB software	14 min

3 Quickstart - Pour l'utilisateur impatient

3.1 Ma première installation

Sans plus tarder, cette section fournira une démonstration rapide et facile d'une installation entièrement automatique à l'aide du CD FAI et d'une machine virtuelle.

Il suffit de télécharger l' image ISO du CD à partir de <http://fai-project.org/fai-cd> et de démarrer votre VM à l'aide de ce CD. Vous verrez un menu grub où vous pouvez choisir parmi différents types d'installation.

Cette installation s'exécutera sans serveur d'installation. L'installation du CD est identique à celle exécutée dans un environnement réseau à l'aide du serveur d'installation FAI.

3.2 Mon premier serveur d'installation

S'il vous plaît noter, si vous avez l'intention d'utiliser QEMU/KVM, vous devez avoir qemu-kvm qemu-utils bridge-utils installés sur la machine à utiliser fai-mk-network et fai-kvm ³.

Vous pouvez le faire via

```
# apt-get install qemu-kvm qemu-utils bridge-utils
```

Si vous avez l'intention d'utiliser VMware ou VirtualBox, assurez-vous que votre client utilise une connexion réseau pontée. En outre, il n'est pas possible d'utiliser des interfaces réseau pontées via le réseau sans fil, car la plupart des cartes réseau WiFi ne prennent pas en charge cette fonctionnalité.

Pour configurer votre propre serveur FAI, nous vous recommandons de créer un réseau de test sur votre ordinateur et d'utiliser KVM. Pour créer ce réseau privé, il ya le script fai-mk-network (dans le paquet fai-server). Il configure un pont logiciel avec plusieurs dispositifs de dérivation qui appartiennent à l'utilisateur<username>.

```
fai-mk-network <username>
```

Après cela, vous pouvez utiliser fai-kvm (-h vous aidera) pour démarrer des machines virtuelles en utilisant KVM qui sont connectés à ce réseau privé. Fais attention. Par défaut, fai-kvm créera les images de disque pour les machines /tmp, qui est un disque RAM sur la plupart des systèmes. Il n'y a aucun problème à créer une image de disque vide de 20G dans /tmp (même si cette partition est de 4 Go de taille), mais alors que la VM écrit des données sur son disque, cela commencera à consommer de l'espace dans /tmp.

³ fai-kvm a besoin de beaucoup de ram pour la vm, à cause de la mise en cache de /var, 2GB sont OK

Démarrez le premier hôte virtuel, qui deviendra le serveur FAI ⁴:

```
fai-kvm -Vn -s20 -u 1 cd fai-cd.iso
```

Dans le menu grub `faiserver,faiserver, fixed IP`. Cela va installer un hôte appelé `faiserver` avec IP 192.168.33.250 qui contient tous les logiciels nécessaires pour un serveur FAI. Il configurera également un cache de paquets local (en utilisant `apt-cacher-ng`). Une fois l'installation terminée, redémarrez la machine. Lors du premier démarrage du nouveau système, il configurera automatiquement le `nfsroot`. Cela peut prendre quelques minutes.

Après cela, vous pouvez démarrer des hôtes supplémentaires en utilisant le démarrage réseau. Pour chaque nouvel hôte, vous devez utiliser une valeur différente pour `-u`, qui sera utilisée pour générer des adresses MAC différentes et utiliser des noms de fichier d'image de disque différents.

```
fai-kvm -Vn -u 2 pxe
```

Ces clients d'installation vous montreront un menu, où vous pouvez sélectionner le type d'installation que vous souhaitez effectuer. Si le client d'installation ne trouve pas le serveur, c'est généralement parce que `fai-monitor` ne fonctionne plus. Cela peut se produire si vous redémarrez le `faiserver` après l'installation. Pour remédier à cela, exécutez simplement `fai-monitor` sur le `faiserver` et relancez le démarrage du client.

Un autre client pourrait être lancé avec:

```
fai-kvm -Vn -u 3 pxe
```

Vous pouvez démarrer autant de machines dans le réseau que les périphériques de prise sont disponibles. Toutes ces machines peuvent se connecter à l'Internet extérieur, mais sont seulement accessibles à partir de votre machine hôte.

4 Vue d'ensemble et concepts

FAI est un système non interactif permettant d'installer, de personnaliser et de gérer les configurations de systèmes et de logiciels Linux sur les ordinateurs ainsi que sur les machines virtuelles et les environnements `chroot`, des petits réseaux aux grandes infrastructures et clusters. Vous pouvez prendre un ou plusieurs PC vierges, mettre sous tension et après quelques minutes, Linux est installé, configuré et exécuté sur l'ensemble du cluster, sans aucune interaction nécessaire. Ainsi, il s'agit d'une méthode évolutive pour installer et mettre à jour un cluster sans surveillance avec peu d'efforts impliqués. FAI utilise le système d'exploitation Linux et une collection de scripts shell et Perl pour le processus d'installation. Les modifications apportées aux fichiers de configuration du système d'exploitation peuvent être effectuées par `CFEngine`, shell (`bash` et `zsh`), Perl, Python, Ruby et attendent des scripts.

Le groupe cible de FAI sont des administrateurs système qui doivent installer Linux sur une ou même des centaines d'ordinateurs. Parce qu'il s'agit d'un outil d'installation à usage général, il peut être utilisé pour l'installation d'un cluster `Beowulf`, d'une batterie de rendu ou d'un laboratoire Linux ou d'une salle de classe. De plus, des réseaux Linux de grande envergure avec différents matériels ou différentes exigences d'installation sont faciles à établir à l'aide de FAI. Mais n'oubliez pas de planifier votre installation. Le chapitre [\[plan\]](#) contient quelques conseils utiles pour ce sujet.

4.1 Conditions Générales

Premièrement, certains termes utilisés dans ce manuel sont décrits.

Installer le serveur

Il fournit les services DHCP, TFTP et NFS ainsi que les données de configuration pour tous les clients d'installation. Dans les exemples de ce manuel, cet hôte s'appelle *faiserver*. L'hôte où le package *fai-server* est installé.

Installer le client

Un hôte qui sera installé à l'aide de FAI et une configuration fournie par le serveur d'installation. Aussi appelé client pour courte. Dans ce manuel, les hôtes d'exemple sont appelés `demohost`, `xfcehost`, `gnomehost` . . . Cet ordinateur doit démarrer à partir de son interface réseau à l'aide de PXE.

⁴ Cette installation consommera environ 2 Go d'espace dans `/tmp`.

Espace de configuration

Une structure de sous-répertoire contenant plusieurs fichiers. Ces fichiers décrivent les détails de la manière dont l'installation des clients sera effectuée. Toutes les données de configuration sont stockées ici. Il est également appelé config space pour le court. Il comprend des informations sur:

- Disposition du disque dur dans un format similaire à fstab
- Systèmes de fichiers locaux, leurs types, points de montage et options de montage
- Logiciels
- Disposition du clavier, fuseau horaire, configuration Xorg, systèmes de fichiers distants, comptes utilisateurs, imprimantes ...

Le package *fai-doc* inclut un exemple d'espace de configuration incluant des exemples pour les hôtes utilisant l'environnement XFCE et GNOME parmi d'autres exemples.

nfsroot, NFS-Root

Un système de fichiers situé sur le serveur d'installation. Pendant le processus d'installation, c'est le système de fichiers complet pour les clients d'installation. Tous les clients partagent le même nfsroot, qu'ils montent en lecture seule. Le nfsroot a besoin d'environ 690 Mo d'espace disque libre.

TFTP

Sert aux clients l'initrd et le noyau utilisés pour le processus d'installation. Avec le système de fichiers servi par NFS, ces deux composent un OS temporaire dans lequel les installations sont exécutées.

Classes FAI

Les classes sont des noms qui déterminent quel fichier de configuration est sélectionné. Si un client appartient à la classe WEBSERVER, il sera configuré en tant que serveur Web, la classe DESKTOP pour, par exemple, détermine les logiciels qui seront installés.

profil

Un profil FAI est juste une liste de classes FAI assigné à un nom de profil, qui est étendu par une description de ce profil. C'est-à-dire que l'on peut avoir deux profils "Webserver", l'un incluant la classe APACHE, y compris la classe NGINX, pour ensuite installer la solution webserver respective.

les tâches

L'installation d'un client se compose de plusieurs parties, appelées tâches. Les tâches sont des sous-programmes prédéfinis qui effectuent une certaine partie de la FAI. Les tâches FAI suivantes sont exécutées au cours d'une installation sur les clients d'installation.

```
confdir # get the config space setup # some initialization, start sshd on demand
defclass # define FAI classes
defvar # define variables
action # evaluate FAI_ACTION
install # Start the installation
partition # partition the harddisks,
create file systems
mountdisks # mount the file systems
extrbase # extract the base.tar.xz
debconf # do the Debian debconf
preseed repository # prepare access to the package repository
updatebase # Set up package tools and update packages
instsoft # install software packages
configure # call customization scripts
finish # do some cleanup, show installation statistics
tests # call tests if defined
chboot # call fai-chboot on the install server
savelog # save log files to local and remote location
faiend # reboot host, eject CD if needed
```

Il s'agit de tâches qui ne sont exécutées que lorsqu'une action différente est exécutée

```
dirinstall # install a chroot environment
softupdate # only do the system configuration
sysinfo # print detailed system information
inventory # print short hardware inventory list
```

Pour une description plus détaillée des *tâches*, voir [\[tasks\]](#).

Notez que vous n'êtes pas limité aux tâches FAI. Vous pouvez également définir des programmes ou des scripts supplémentaires qui seront exécutés à certaines occasions. On les appelle des *hooks*.

hooks

Les Hooks sont des plugins, ils peuvent ajouter des fonctionnalités supplémentaires au processus d'installation ou même remplacer des tâches entières de FAI. Les Hooks sont expliqués en détail dans [\[hooks\]](#).

4.2 Le concept de classe

Les classes sont utilisées dans presque toutes les tâches de l'installation. Les classes déterminent quels fichiers de configuration choisir parmi une liste d'alternatives disponibles. Pour déterminer les fichiers de configuration à utiliser, FAI recherche la liste des classes définies et utilise tous les fichiers de configuration correspondant à un nom de classe⁵. La boucle suivante implémente cette fonction en pseudo code shell:

```
for class in $all_classes; do
  if [ -r $config_dir/$class ]; then      # if a file with name $class exists
    your_command $config_dir/$class      # call a command with this file name
    # exit if only the first matching file is needed
  fi
done
```

La caractéristique très intéressante de ceci est que vous pouvez ajouter une nouvelle alternative de configuration et elle sera automatiquement utilisée par FAI sans changer le code, si le fichier de configuration utilise un nom de classe.

C'est parce que la boucle détecte automatiquement les nouveaux fichiers de configuration qui doivent être utilisés. L'idée d'utiliser des classes en général et d'utiliser certains fichiers correspondant à un nom de classe pour une configuration est adoptée à partir des scripts d'installation par Casper Dik pour Solaris. Cette technique s'est avérée très utile et facile.

Vous pouvez regrouper plusieurs hôtes partageant les mêmes fichiers de configuration en utilisant la même classe. Vous pouvez également diviser l'ensemble des données de configuration pour tous les clients en plusieurs classes et les utiliser comme des briques de lego et construire la configuration entière pour un seul client en assemblant les briques ensemble.

Si un client appartient à la classe *A*, nous disons que la classe *A* est définie pour ce client. Une classe n'a pas de valeur, elle est juste définie ou non définie.

Les classes déterminent comment l'installation est effectuée. Par exemple, un client d'installation peut être configuré pour obtenir le bureau XFCE en y ajoutant simplement la classe *XFCE*. Naturellement, des configurations plus granulaires sont également possibles. Par exemple, les classes peuvent décrire comment le disque dur doit être partitionné, ils peuvent définir quels paquets logiciels seront installés ou quelles étapes de personnalisation seront exécutées.

Souvent, une configuration client est créée en modifiant ou en ajoutant uniquement les classes auxquelles ce client appartient, ce qui rend l'installation d'un nouveau client très facile. Ainsi, aucune information supplémentaire ne doit être ajoutée à l'espace de configuration si les classes existantes suffisent à vos besoins.

Comme vous pouvez le voir, les classes sont un pilier central de la personnalisation de votre espace de configuration et de l'installation de votre client. Pour définir vos propres classes, reportez-vous à [\[defining classes\]](#).

5 Configurer votre faiserver

Voici comment configurer le serveur d'installation en quelques minutes. Les étapes suivantes sont nécessaires:

1. Configurer le serveur d'installation
 - a. Installer des packages FAI
 - b. Créez le nfsroot
 - c. Copiez les exemples dans l'espace de configuration
 - d. Configurer les démons réseau
 - e. Créer les configurations PXELINUX
2. Démarrage et installation des clients

⁵ Il est également possible d'utiliser uniquement le fichier de configuration avec la plus haute priorité puisque l'ordre des classes définit une priorité de bas à haut dans la liste des classes.

5.1 Installer les paquetages FAI

- Installez la clé du référentiel de package de projet FAI:
- Ajoutez l'URL du référentiel de packages du projet FAI.
- Installez le paquet `fai-quickstart` sur votre serveur d'installation .

```
# wget -O - http://fai-project.org/download/074BCDE4.asc | apt-key add -
# echo "deb http://fai-project.org/download jessie koeln" > /etc/apt/sources.list.d/fai. ←
  list
# apt-get update
# aptitude install fai-quickstart
```

Cela installera également les paquets pour les démons de serveur DHCP, TFTP et NFS.

5.2 Créez le nfsroot

- Activez également le référentiel de package du projet FAI dans un autre fichier `sources.list` qui est utilisé lors de la construction du `nfsroot`. Ensuite, activez l'utilisateur de journal pour FAI.

```
# sed -i -e 's/^#deb/deb/' /etc/fai/apt/sources.list
# sed -i -e 's/#LOGUSER/LOGUSER/' /etc/fai/fai.conf
```

- Par défaut, FAI utilise <http://httpredir.debian.org> comme mirror de paquets, qui devrait tenter de trouver un référentiel de paquets rapide pour vous. ⁶ Maintenant, nous pouvons exécuter `fai-setup(8)` ⁷ Et vérifier si tout s'est bien passé. Le fichier journal est écrit dans `/var/log/fai/fai-setup.log`.

```
# fai-setup -v
```

- Ce sont quelques-unes des lignes que vous verrez à la fin de `fai-setup` . Un exemple complet de `fai-setup.log` est disponible sur la page Web FAI à l'adresse <http://fai-project.org/logs/fai-setup.log>.

```
FAI packages and related packages inside the nfsroot:
dracut                044+189-2
dracut-network        044+189-2
fai-client            5.3.3~bpo8+2
fai-nfsroot           5.3.3~bpo8+2
fai-setup-storage    5.3.3~bpo8+2
Waiting for background jobs to finish
fai-make-nfsroot finished properly.
Log file written to /var/log/fai/fai-make-nfsroot.log
Adding line to /etc/exports: /srv/fai/config 192.168.33.250/25(async,ro,no_subtree_check)
Adding line to /etc/exports: /srv/fai/nfsroot 192.168.33.250/25(async,ro,no_subtree_check, ←
  no_root_squash)
Reloading nfs-kernel-server configuration (via systemctl): nfs-kernel-server.service.

You have no FAI configuration space yet. Copy the simple examples with:
cp -a /usr/share/doc/fai-doc/examples/simple/* /srv/fai/config
Then change the configuration files to meet your local needs.
Please don't forget to fill out the FAI questionnaire after you've finished your project ←
  with FAI.

FAI setup finished.
Log file written to /var/log/fai/fai-setup.log
```

⁶ Si vous souhaitez utiliser un miroir plus rapide, ajustez l'URL dans `/etc/fai/apt/sources.list` et `FAI_DEBOOTSTRAP` in `/etc/fai/nfsroot.conf` avant d'appeler `fai-setup`.

⁷ Ceci appellera `fai-make-nfsroot(8)` interne.

- Fai-setup a créé le LOGUSER, le nfsroot et a ajouté des lignes supplémentaires à */etc/exports*. Les sous-répertoires ajoutés à */etc/exports* sont exportés via NFS v3, de sorte que tous les clients d'installation dans le même sous-réseau peuvent les monter via NFS.

5.3 Création de l'espace de configuration

Installez les exemples simples dans l'espace de configuration ⁸.

```
$ cp -a /usr/share/doc/fai-doc/examples/simple/* /srv/fai/config/
```

Ces exemples contiennent la configuration pour certains hôtes d'exemple. Selon le nom d'hôte utilisé, votre ordinateur sera configuré comme suit:

demohost

Une machine qui n'a besoin que d'un petit disque dur. Cette machine est configurée avec le réseau en tant que client DHCP, et une démo de compte est créée.

xfcehost

Un bureau XFCE est installé, utilisant LVM, et la démo du compte est créée.

gnomelhost

Un bureau GNOME est installé et la démo du compte est créée.

other host names

Les hôtes disposant d'un autre nom d'hôte utiliseront notamment les classes FAIBASE, DHCPC et GRUB.

Tous les hôtes auront un compte appelé demo avec mot de passe *fai*. Le compte root a également le mot de passe *fai*.

Si l'indicateur FAI menu est ajouté, au lieu d'utiliser le nom d'hôte pour déterminer le type d'installation, un menu est présenté et l'utilisateur peut choisir un profil pour l'installation.

5.4 Configurer les démons réseau

Pour démarrer le client d'installation via PXE, le serveur d'installation a besoin d'un DHCP et d'un démon TFTP en cours d'exécution. Le paquet *fai-quickstart* a déjà installé les progiciels pour ces daemons. En outre, le paquetage du serveur NFS pour l'exportation du nfsroot et de l'espace de configuration a été installé.

5.4.1 Configuration du démon DHCP

également, votre faiserver doit également être votre serveur DHCP. Si ce n'est pas le cas, demandez à l'administrateur responsable du serveur DHCP de le configurer conformément à cette section. En option, il est possible d'éviter cela en utilisant la fonctionnalité [\[autodiscover\]](#) diffusée dans FAI 5.0.

Un exemple pour *dhcpd.conf* (5) est fourni avec le paquet *fai-doc*. Commencez à utiliser cet exemple et regardez toutes les options qui y sont utilisées.

```
# cp /usr/share/doc/fai-doc/examples/etc/dhcpd.conf /etc/dhcp/
```

Les seules informations spécifiques FAI contenues dans ce fichier de configuration sont de définir le filename de *fai/pxelinux.0* et de définir *next-server* et *server-name* sur le nom de votre serveur d'install. Toutes les autres informations sont uniquement des données liées au réseau, qui est utilisé dans presque toutes les configurations DHCP. Ajustez ces paramètres de réseau à vos besoins locaux.

⁸ Ces fichiers ne doivent pas appartenir au compte racine.

```
deny unknown-clients;
option dhcp-max-message-size 2048;
use-host-decl-names on;

subnet 192.168.33.0 netmask 255.255.255.0 {
    option routers 192.168.33.250;
    option domain-name "my.example";
    option domain-name-servers 192.168.33.250;
    option time-servers faiserver;
    option ntp-servers faiserver;
    server-name faiserver;
    next-server faiserver;
    filename "fai/pxelinux.0";
}
```

Si vous apportez des modifications à la configuration DHCP, vous devez redémarrer le démon.

```
# /etc/init.d/isc-dhcp-server restart
```

Si vous disposez de plusieurs interfaces réseau, vous pouvez définir l'interface que le serveur écoutera dans */etc/default/isc-dhcp-server*. Par défaut, le démon DHCP écrit ses messages de journalisation dans */var/log/daemon.log*.

5.4.2 Ajout d'une entrée d'hôte au DHCP

L'adresse MAC est donnée par le matériel de la carte réseau. Pour chaque client d'installation, vous collectez son adresse MAC et la mappez à une adresse IP et à un nom d'hôte. Tout d'abord, nous ajoutons l'adresse IP et le nom d'hôte à */etc/hosts*⁹.

```
192.168.33.100    demohost
```

Le mappage de l'adresse MAC à l'adresse IP est effectué dans le fichier *dhcpd.conf*. Ici, nous ajoutons une entrée d'hôte en utilisant la commande `dhcp-edit (8)`. Ici, vous devez remplacer `01:02:03:AB:CD:EF` avec le MAC que vous avez trouvé.

```
# dhcp-edit demohost 01:02:03:AB:CD:EF
```

Après avoir appelé cette commande, c'est ce que l'entrée hôte dans *dhcpd.conf* ressemblera à:

```
host demohost {hardware ethernet 01:02:03:AB:CD:EF;fixed-address demohost;}
```

5.4.3 TFTP

Normalement, vous n'avez pas besoin d'apporter de modifications à la configuration du démon TFTP. Les fichiers fournis par TFTP sont situés dans */srv/tftp/fai*.

5.4.4 NFS

La commande `fai-setup` a déjà configuré le démon NFS et ajouté quelques lignes au fichier de configuration */etc/exports*. Il exporte les répertoires en utilisant NFS v3.

5.5 Création de la configuration PXELINUX

La dernière étape avant de démarrer votre client pour la première fois est de spécifier quelle configuration le client doit démarrer lors de l'amorçage PXE. Nous `fai-chboot(8)` la commande `fai-chboot (8)` pour créer une configuration pxelinux pour chaque client d'installation. Cela comprend des informations sur le noyau, l'initrd, l'espace de configuration et certains paramètres d'amorçage. Vous devriez lire la page de manuel, qui vous donne quelques bons exemples. Voici la commande pour démarrer l'installation de l'hôte demohost.

⁹ Vous pouvez également ajouter ceci dans votre système de noms de domaine (DNS)

```
$ fai-chboot -IFv -u nfs://faiserver/srv/fai/config demohost
Booting kernel vmlinuz-3.16.0-4-amd64
append initrd=initrd.img-3.16.0-4-amd64 ip=dhcp
    FAI_FLAGS=verbose,sshd,createvt
    FAI_CONFIG_SRC=nfs://faiserver/srv/fai/config

demohost has 192.168.33.100 in hex C0A82164
Writing file /srv/tftp/fai/pxelinux.cfg/C0A82164 for demohost
```

À ce stade, vous devriez avoir une configuration faiserver de travail et vos clients devraient démarrer dans FAI et être en mesure d'installer l'un des exemples.

Dans la section suivante, vous pouvez lire la planification de votre installation, adapter votre espace de configuration à vos besoins particuliers et étendre FAI à l'aide de hooks.

5.6 Serveur personnalisé

Le faiseur et sa configuration n'est nullement statique. Il est possible de personnaliser et d'étendre votre serveur. Pour cela, reportez-vous à la section [Customizing your install server setup] dans [advanced].

6 Planifiez votre installation

Avant de commencer votre installation, vous devriez investir beaucoup de temps dans la planification de votre installation. Une fois que vous êtes satisfait de votre concept d'installation, FAI peut faire toutes les tâches ennuyeuses et répétitives pour transformer vos plans en réalité. FAI ne peut pas faire de bonnes installations si votre concept est imparfait ou manque de quelques détails importants. Commencez à planifier l'installation en répondant aux questions suivantes:

- Est-ce que je vais créer un cluster Beowulf ou dois-je installer des machines de bureau?
- À quoi ressemble ma topologie LAN?
- Ai-je un matériel uniforme? Le matériel sera-t-il uniforme à l'avenir?
- Le matériel a-t-il besoin d'un noyau spécial?
- Comment nommer les hôtes?
- Comment les disques durs locaux doivent-ils être partitionnés?
- Quelles applications seront exécuté par les utilisateurs?
- Les utilisateurs ont-ils besoin d'un système de mise en file d'attente?
- Quel logiciel doit être installé?
- Quels démons devraient être lancés, et à quoi devrait ressembler la configuration?
- Quels systèmes de fichiers distants doivent être montés?
- Comment effectuer les sauvegardes? How should backups be performed?

Vous devez également penser à des comptes d'utilisateur, des imprimantes, un système de courrier, des travaux de cron, des cartes graphiques, l'initialisation double, le NIS, le NTP, le fuseau horaire, la disposition de clavier, l'exportation et le montage des annuaires via NFS et beaucoup d'autres choses. Donc, il ya beaucoup à faire avant de commencer une installation. Et rappelez-vous que la connaissance est le pouvoir, et c'est à vous de l'utiliser. L'installation et l'administration sont un processus et non un produit. FAI ne peut pas faire les choses que vous ne lui dites pas de faire.

Mais vous ne devez pas commencer à partir de zéro. Examinez les fichiers et les scripts dans l'espace de configuration. Il ya beaucoup de choses que vous pouvez utiliser pour votre propre installation. Un bon article intitulé «Bootstrapping an Infrastructure» avec d'autres aspects de la construction d'une infrastructure est disponible sur <http://www.infrastructures.org/papers/bootstrap/-bootstrap.html>

6.1 L'espace de configuration et ses sous-répertoires

L'espace de configuration est la collection d'informations sur la façon exacte d'installer un client. L'espace de configuration central pour tous les clients d'installation se trouve sur le serveur d'installation dans `/srv/fai/config` et ses sous-répertoires. Cela sera monté par les clients d'installation dans `/var/lib/fai/config`. La commande d'installation principale `fai(8)` utilise tous ces sous-répertoires dans l'ordre indiqué sauf pour les hooks.

class/

Scripts et fichiers pour définir des classes et des variables.

disk_config/

Fichiers de configuration pour le partitionnement de disque, RAID logiciel, LVM et création de système de fichiers.

basefiles/

Normalement, le fichier *base.tar.xz* (situé à l'intérieur du `nfsroot`) est extrait sur le client d'installation après la création des nouveaux systèmes de fichiers et avant l'installation du package. Il s'agit d'une image de base minimale, créée juste après avoir appelé `debootstrap` lors de la création du `nfsroot` sur le serveur d'installation. Si vous voulez installer une autre distribution que la `nfsroot`, vous pouvez mettre un fichier tar dans le sous-répertoire *basefiles/* et le nommer après une classe. Ensuite, la commande `'ftar(8)'` est utilisée pour extraire le fichier tar en fonction des classes définies. Ainsi, le fichier doit être nommé *CLASS.tar.xz* et non *CLASS.base.tar.xz*. Cela se fait dans la tâche *extrbase*. Utilisez cette option si vous souhaitez installer une autre distribution ou une version différente de celle exécutée pendant l'installation.

Ce fichier de base peut également être reçu en fonction des classes FAI via HTTP ou FTP en définissant la variable `FAI_BASEFILEURL`. FAI téléchargera un fichier `CLASSNAME.tar.xz` (ou `tgz`, ou `tar.gz`, ...) à partir de cette URL, si `CLASSNAME` correspond à une classe FAI.

Exemple:

```
FAI_BASEFILEURL=http://fai-project.org/download/basefiles
```

Le dossier doit prendre en charge la liste des répertoires. FAI ne recherchera pas de fichiers potentiellement correspondants.

Voir le chapitre [\[otherdists\]](#) pour savoir comment installer différentes distributions.

debconf/

Ce répertoire contient toutes les données `debconf(7)`. Le format est le même que celui utilisé par `debconf-set-selections(8)`.

package_config/

Les fichiers contenant des noms de classe contiennent des listes de logiciels à installer ou à désinstaller par `'install_packages(8)'`. Les fichiers nommés `<CLASS>.asc` sont ajoutés à la liste des clés utilisées par `apt` (à l'aide de `'`apt-key(8)'`) pour les dépôts de paquets approuvés.

scripts/

Scripts pour la personnalisation de votre site local. Utilisé par `fai-do-scripts(1)`.

files/

Les Fichiers utilisés par les scripts de personnalisation. La plupart des fichiers se trouvent dans une structure de sous-arborescence qui reflète l'arborescence de répertoires ordinaire. Par exemple, les modèles de *nsswitch.conf* se trouvent dans `$FAI/files/etc/nsswitch.conf` et sont nommés en fonction des classes auxquelles ils doivent correspondre: `$FAI/files/etc/nsswitch.conf/NIS` est la version de `/etc/nsswitch.conf` à utiliser pour la classe NIS. Notez que le contenu du répertoire n'est pas automatiquement copié sur la machine cible, mais qu'il doit être explicitement copié par des scripts de personnalisation à l'aide de la commande `fcopy(8)`.

hooks/

Les hooks sont des programmes ou des scripts définis par l'utilisateur, qui sont appelés pendant le processus d'installation. cela peut étendre ou remplacer les tâches par défaut. Le nom du fichier doit être de format `taskname.CLASSNAME[.sh]`. Un hook appelé `updatebase.DEBIAN` est exécuté avant la mise à jour de la tâche `updatebase` et seulement si l'installation du client appartient à la classe DEBIAN.

6.2 Définition des classes

Il existe différentes possibilités pour définir des classes: . Certaines classes par défaut sont définies pour chaque hôte: DEFAULT, LAST et son nom d'hôte. . Les classes peuvent être répertoriées dans un fichier. . Les classes peuvent être dynamiquement définies par des scripts.

La dernière option est une fonctionnalité très intéressante, puisque ces scripts définiront des classes est un moyen très flexible. Par exemple, plusieurs classes peuvent être définies uniquement si certains matériels sont identifiés ou si une classe est définie en fonction des informations de sous-réseau du réseau.

Tous les noms de classes, sauf le nom d'hôte, sont écrits en majuscules. Ils ne doivent pas contenir un trait d'union, un dièse, un Point-Virgule OÙ un point, mais PEUVENT contenir des caractères de soulignement et des Chiffres.

La tâche *defclass* appelle la commande `fai-class(1)` pour définir les classes. Tous les scripts correspondant `^[0-9][0-9]*` (qui commencent avec deux chiffres) dans le sous-repertoire `$FAI/class` sont exécutés afin de définir les classes. Tout ce qui est affiché sur `STDOUT` est automatiquement définie comme une classe. Pour plus d'informations sur les définitions de classe, lire les pages de manuel versent `fai-class(1)`. Le script `50-host-classes` (voir ci-dessous la version allégée) est utilisé pour les définir des classes en fonction du nom d'hôte.

```
# use a list of classes for our demo machines
case $HOSTNAME in
  demohost)
    echo "FAIBASE GRUB DHCPC DEMO" ;;
  xfcehost)
    echo "FAIBASE GRUB DHCPC DEMO XORG XFCE" ;;
  faiserver)
    echo "FAIBASE DEBIAN DHCPC DEMO FAISERVER" ;;
  *)
    echo "FAIBASE GRUB DHCPC" ;;
esac
```

Les noms d'hôtes doivent rarement être utilisés pour les fichiers de configuration dans l'espace de configuration. À la place, une classe doit être définie et ensuite ajoutée pour un hôte donné. En effet, la plupart du temps les données de configuration ne sont pas spécifiques au nom d'hôte, mais peuvent être partagées entre différents hôtes.

L'ordre des classes est important car elle définit la priorité des classes de faible à élevé.

6.3 Définition des Variables

La tâche *defvar* définit les variables pour l'installation du client. Les variables sont définies par les scripts dans la `class/*.var`. Toutes les variables globales PEUVENT être définies dans `DEFAULT.var`. Pour certains groupes d'hôtes utiliser un fichier de classe ou pour un seul hôte utiliser le fichier `$HOSTNAME.var`. Ici aussi, il est utile d'étudier tous les exemples.

Les variables suivantes sont utilisées dans les exemples et peuvent être aussi utiles pour votre installation:

FAI_ACTION

Réglez les actions que doit effectuer FAI. Normalement, ceci se fait par `fai-chboot(8)`. Si vous ne pouvez pas utiliser cette commande, définir la variable dans le script `LAST.var`.

FAI_ALLOW_UNSIGNED

Si défini à 1, FAI permet l'installation de paquets à partir de référentiels non signés.

CONSOLEFONT

La police de qui est chargée lors de l'installation par `setfont(8)`.

KEYMAP

Définit les fichiers de mappage du clavier dans `/usr/share/keymaps` et `$FAI/files`. Vous ne devez pas spécifier le chemin complet, puisque ce fichier sera localisé automatiquement.

ROOTPW

Le mot de passe root chiffré pour le nouveau système. Vous pouvez utiliser `'crypt(3)`, `md5` et d'autres types de hachage pour le mot de passe. Utilisez ``mkpasswd(1)` pour créer le hachage d'un certain mot de passe. Par exemple, pour générer le hachage MD5 pour l'utilisation du mot de passe.

```
$ echo "yoursecretpassword" | mkpasswd -Hmd5 -s
```

UTC

Réglez l'horloge du matériel à UTC si *UTC=yes*. Sinon, régler l'horloge à l'heure locale. Voir `clock(8)` pour en plus d'informations.

TIMEZONE

Est-ce que le fichier d'initialisation par rapport à */usr/share/zoneinfo/* indique votre fuseau horaire. Par exemple: *TIMEZONE=Europe/Berlin*.

MODULESLIST

Une liste des modules du Noyau qui sont chargés pendant Le démarrage du nouveau système (Écrit dans */etc/modules*).

6.4 Configuration du disque dur

L'outil `setup-storage(8)` lit le fichier dans *\$FAI/disk_config* pour la configuration du disque. Ce fichier décrit comment tous les disques Locaux devrons etre partitionné, Quels types de Systèmes de Fichiers doivent etre écrits (Comme ext3/4, xfs, btrfs), et où ils seront Montés. Vous pouvez aussi créer des configurations RAID logiciel et LVM en Utilisant le Fichier de configuration. Il Est aussi possible de la mise en Conservation de le partitionnage du disque ou de conserver Les Données sur CERTAINES partitions.

Pendant le Processus d'installation de tous les Systèmes de Fichiers Locaux Sont Montés par rapport à */target*. Par exemple, si vous Specifiez le Point de montage */home* Dans un Fichier de configuration de disque, ce sera le répertoire */target/home* pendant le Processus d'installation et deviendra */home* pour le nouveau système Installé.

6.5 Extraction du fichier de base

6.6 Debconf préconfiguration

6.7 L'Accès au dépôt de paquetages

6.8 configuration du progiciel

Avant l'installation de de paquets, FAI va ajouter le contenu de Tous les Fichiers nommés *package_config/class.asc* à la liste des clés apt. Si votre depo locale est signé par votre keyid AB12CD34 vous pouvez Facilement ajouter cette clé, aussi FAI l'utilisera pendant l'installation. Utilisez cette commande pour Créer le fichier *CLASS.asc*:

```
faiserver$ gpg -a --export AB12CD34 > /srv/fai/config/package_config/MYCLASS.asc
```

Le script `install_packages(8)` installe les Logiciels Sélectionnés. Il lira tous les fichiers de configuration Dans *\$FAI/package_config* Dont le nom correspond aux classes définie. La syntaxe est tres simple.

```
# an example package class
```

```
PACKAGES taskinst
german
```

```
PACKAGES aptitude
adduser netstd ae
less passwd
```

```
PACKAGES remove
gpm xdm
```

```
PACKAGES aptitude GRUB
lilo- grub
```

Commentaires Commencent par un Dièse et se terminent à la fin de la ligne. Chaque commande de paquetage commence par Le mot *PACKAGES* Suivi par un nom de commande, Ce qui correspond à l'outil de package Comme *apt-get*, *aptitude* ou *yum* par exemple. la commande qui définit la commandent qui sera utilisé pour installer les paquets nommés après cette commande. La liste de toutes les commandes disponibles peuvent Être listé en utilisant *install_packages -H*. Les paquets d'outils pris en charges son *aptitude*, *apt-get*, *smart*, *yast*, *yum*, *rpm*, *zypper*

hold

Mettez un paquet en attente. Ce Paquet ne sera pas pris en charges par *dpkg*, pas exemple non mis à niveau.

install

Installez Tous les paquets (en utilisant *apt-get*) Qui sont précise dans les lignes Suivantes. Si un tiret est ajouté au nom du paquet (sans espace intermédiaire), le paquet sera supprimé, pas installé. Tous les noms de paquets sont vérifiées pour les fautes d'orthographe. Tout paquet qui n'existe pas, seront retiré de la liste des paquets à l'installation. Soyer donc prudentes de ne pas mal orthographier les noms de paquets.

install-norec

Comme *install*,mais sans installer les paquets recommandés.

remove

Supprimer tous les paquets qui sont préciser dans les lignes suivantes. Annexer un + au nom du paquet si le paquet doit Être installé.

taskinst

Installez tous les paquets appartenant aux tâches qui sont spécifiées dans les lignes suivantes à l'aide de *tasksel(1)*. Vous pouvez aussi utiliser *aptitude* pour installer les tâches.

aptitude

Installez Ttus les paquets avec la commande *aptitude*. Ce sera la Valeur par défaut à l'avenir et pourra remplacer *apt-get* et *taskinst*. Aptitudes peut aussi installer les paquets

aptitude-r

Idem *aptitude* avec l'option *--with-recommends*.

unpack

Télécharge les paquets et décompresse seulement. Ne configure pas le paquet.

dselect-upgrade

Defini la sélections des paquets en Utilisant les lignes suivantes et installe ou supprime les paquets précisés. Ces lignes sont le résultat de la commande *dpkg --get-selections*. Il est recommandé de ne pas utiliser ce format, puisque vous devez aussi specifiez tous les paquets qui ne sont pas installés en raison d'une dépendance ou recommandation. Il vaut mieux juste spécifier le paquet que vous voulez avoir, et de laisser FAI (et *apt-get*) résoudre les dépendances.

Plusieurs lignes avec des listes de noms de paquets séparés par des espaces suivent les directive *PACKAGES*. Toutes les dépendances sont résolues. Les paquetages avec suffixe - (par exemple, *lilo-*) seront supprimés au lieu d'être installés. L'ordre des paquet n'a pas d'importance. Si vous souhaitez installer des paquets d'une autre version que la valeur par défaut, vous pouvez ajouter le nom de la version au nom du paquet comme dans openoffice.org/etch-backports. Vous pouvez également spécifier une certaine version comme *apt=0.3.1*. Plus d'informations sur ces fonctionnalités sont décrites dans *aptitude(8)*.

Une ligne qui contient la commande *PRELOADRM*, télécharge un fichier à l'aide de *wget(1)* dans un répertoire avant d'installer les packages. À l'aide du *file: URL*, ce fichier est copié de *\$FAI_ROOT* vers le répertoire de téléchargement. Par exemple, le package *realplayer* a besoin d'une archive pour installer le logiciel, donc cette archive est téléchargée dans le répertoire */root*. Après l'installation des paquets, ce fichier sera supprimé. Si le fichier ne doit pas être supprimé, utilisez plutôt la commande *PRELOAD*.

Il est possible d'ajouter une liste de noms de classes après la commande pour *apt-get*. Ainsi, cette commande *PACKAGE* ne sera exécutée que si la classe correspondante est définie. Ainsi, vous pouvez combiner de nombreux petits fichiers dans le fichier *DEFAULT*. ATTENTION! Utilisez cette fonctionnalité uniquement dans le fichier *DEFAULT* pour garder tout simple. Voir ce fichier pour quelques exemples.

Si vous souhaitez supprimer un nom de paquet d'une certaine classe faisait partie avant de cette classe , vous ne devez pas supprimer le nom du paquet classe, mais plutôt de lui ajouter un tiret (-). Cela garantira que le paquet est enlevé pendant une mise a jour sur des hôtes qui étaient Installé en utilisant l'ancienne définition de classe qui comprenait ce nom de paquet.

Si vous spécifiez un paquet qui n'existe pas, ce paquet sera supprimé automatiquement de la liste d'installation uniquement si la commande *install* est utilisée.

6.9 Scripts de personnalisation

La commande `fai-do-scripts(1)` est appelée pour exécuter tous les scripts dans ce répertoire. Si un répertoire avec un nom de classe existe, tous les scripts correspondant à `^[0-9][0-9]*` sont exécutés par ordre alphabétique. Il est donc possible d'utiliser des scripts de différentes langues (shell, cfengine, Perl, Python, Ruby, expect,..) pour une classe.

Ces scripts écrivent leur sortie dans différents fichiers journaux, selon le type de script. Par exemple, Tous les scripts shell écrivent leur journal dans *shell.log*.

6.9.1 Scripts shell

La plupart des scripts sont des scripts Bourne shell. Les scripts shell sont utiles si la tâche de configuration ne doit seulement appeler certaines commandes shell ou créer un fichier à partir de zéro. Afin de ne pas écrire beaucoup de scripts courts, il est possible d'utiliser la commande `ifclass` pour tester si certaines classes sont définies.

```
ifclass -o A B C
```

Vérifie si l'une des classes A, B ou C est définie. L'utilisation de `-a` (AND logique) vérifie si toutes les classes d'une liste sont définies. La commande `ifclass C` vérifie si seule la classe C est définie.

Pour copier des fichiers avec des classes, utilisez la commande `fcopy(8)`. Si vous voulez extraire une archive à l'aide de classes, utilisez `ftar(8)`. Pour ajouter des lignes à un fichier de configuration, utilisez `ainsl(1)` au lieu de simplement `echo string >> filename`.

FAI prend également en charge les scripts `zsh(1)` pendant la tâche de personnalisation. Dans les scripts, la variable `$classes` contient une liste séparée par des espaces avec les noms de toutes les classes définies.

6.9.2 Scripts cfengine

CFEngine dispose d'un riche ensemble de fonctions pour modifier les fichiers de configuration existants, par exemple *Locate-LineMatching*, *ReplaceAll*, *InsertLine*, *AppendIfNoSuchLine*, *HashCommentLinesContaining*. Mais il ne peut pas traiter les variables qui sont indéfinies. Si une variable n'est pas définie, l'ensemble du script cfengine s'arrêtera.

Plus d'informations peuvent être trouvées dans la page de manuel `cfengine(8)` ou sur la page d'accueil cfengine <http://www.cfengine>

6.10 Hooks

Les Hooks vous permettent de spécifier des fonctions ou des programmes qui sont exécutés à certaines étapes du processus d'installation. Avant qu'une tâche soit appelée, FAI recherche les hooks existants pour cette tâche et les exécute. Comme on peut s'y attendre, les classes sont également utilisées lors de l'appel de hooks. Les hooks sont exécutés pour chaque classe définie. Vous n'avez qu'à créer le hook avec le nom de la classe désirée et il sera utilisé. Si plusieurs hooks pour une tâche existent, ils sont appelés dans l'ordre défini par les classes. Si *debug* est inclus dans `$FAI_FLAG` l'option `-d` est passée à tous les hooks, donc vous pouvez déboguer vos propres hooks. Si certaines tâches par défaut doivent être ignorées, utilisez la sous-routine *skiptask* et une liste de tâches par défaut comme paramètres. Dans les exemples fournis, les hooks de la classe CENTOS ignorent certaines tâches spécifiques de Debian.

Le répertoire `$FAI/hooks/` contient tous les hooks. Un hook est un fichier exécutable qui suit le nom de tâche `taskname.CLASSNAME[.sh]` (par exemple, `repository.CENTOS` ou `savelog.LAST.sh`), un nom de tâche et un nom de classe séparés par un point, éventuellement suivi de `.sh`. Le nom de la tâche spécifie la tâche devant précéder l'exécution de ce hook, si la classe spécifiée est définie pour le client d'installation. Voir la section [\[tasks\]](#) pour une liste complète des tâches par défaut pouvant être utilisées.

Un hook du formulaire `hookprefix.classname` ne peut pas définir de variables pour le script d'installation, car il s'agit d'un sous-processus. Mais vous pouvez utiliser n'importe quel exécutable binaire ou n'importe quel script que vous avez écrit. Les hooks qui ont le suffixe `.sh` (par exemple, `'partition.DEFAULT.sh`) doivent être des scripts Bourne shell et sont sourcés. Il est donc possible de redéfinir des variables pour les scripts d'installation.

Dans la première partie de FAI, tous les hooks avec le préfixe *confdir* sont appelés. Ces hooks ne peuvent pas être localisés dans l'espace de configuration, car il n'est pas encore disponible. Par conséquent, ces hooks sont les seuls hooks situés dans `$nfsroot/$FAI/hooks` sur le serveur d'installation. Tous les autres hooks se trouvent dans `$FAI_CONFIGDIR/hooks` sur le serveur d'installation.

Tous les hooks appelés avant la définition des classes ne peuvent utiliser que les classes suivantes: *DEFAULT \$HOSTNAME LAST*. Si un hook pour la classe *DEFAULT* doit être appelé uniquement si aucun hook pour la classe `$HOSTNAME` n'est disponible, insérez ces lignes sur le hook par défaut:

```
hookexample.DEFAULT:

#! /bin/sh

# skip DEFAULT hook if a hook for $HOSTNAME exists
scriptname=$(basename $0 .DEFAULT)
[-f $FAI/hooks/$scriptname.$HOSTNAME ] && exit
# here follows the actions for class DEFAULT
.
.
```

Quelques exemples de ce que les hooks pourraient être utilisés:

- Charger les modules du noyau avant que les classes soient définies dans `$FAI/class`.
- Envoyez un courriel à l'administrateur si l'installation est terminée.
- Installez un client sans disque et sautez le partitionnement de disque local.
- Jetez un oeil à `hooks/debconf.IMAGE` pour savoir comment cloner une machine en utilisant une image de système de fichiers.

6.11 FAI flags

La variable `$FAI_FLAGS` contient une liste de flags séparés par des espaces. Les flags suivants sont connus:

verbose

Créez une sortie verbeuse pendant l'installation. Cela doit toujours être le premier flag, de sorte que les définitions consécutives des flags seront affichées verbeusement.

debug

Créer une sortie de débogage. Aucune installation sans assistance n'est effectuée. Pendant l'installation du paquet, vous devez répondre à toutes les questions des scripts postinstall sur la console du client. Beaucoup d'informations de débogage seront imprimées. Ce flag n'est utile que pour les développeurs FAI.

sshd

Démarrez le démon ssh pour activer les connexions à distance. Vous pouvez ensuite vous connecter en tant que root à tous les clients d'installation pendant l'installation. Le mot de passe par défaut est `fai` et peut être modifié en définissant `FAI_ROOTPW` dans `nfsroot.conf(5)`. Pour vous connecter à partir de votre serveur vers le client d'installation (nommé `demohost` dans cet exemple), utilisez:

```
$ ssh root@demohost
Warning: Permanently added 'demohost,192.168.33.100' to the list of known hosts.
root@demohost's password:
```

Ce n'est que le mot de passe root pendant le processus d'installation, pas pour le nouveau système installé. Vous pouvez également vous connecter sans mot de passe lorsque vous utilisez `$SSH_IDENTITY`.

createvt

Créez deux terminaux virtuels et exécutez un bash si `ctrl-c` est tapé dans le terminal de console. Vous pouvez accéder aux terminaux supplémentaires en tapant `Alt-F2` ou `Alt-F3`. Sinon, aucun terminal n'est disponible et la saisie `ctrl-c` va redémarrer le client d'installation. La définition de ce flag est utile pour le débogage. Si vous voulez une installation qui ne devrait pas être interrompible, ne définissez pas ce flag.

menu

Cela permet à un menu utilisateur de sélectionner un profil. Tous les fichiers `class/*.profile` sont lus et un menu basé sur des curses sera créé.

reboot

Redémarrez le client d'installation une fois l'installation terminée sans taper RETURN sur la console. Si ce drapeau n'est pas défini, et que `error.log` contient quelque chose, le client d'installation s'arrêtera et attendra que vous appuyez sur RETURN. Si aucune erreur ne s'est produite, le client redémarre automatiquement.

halt

Arrêtez le client d'installation à la fin de l'installation, au lieu de redémarrer dans le nouveau système.

initial

Utilisé par `setup-storage(8)`. Les partitions marquées avec `preserve_reinstall` sont préservées à moins que ce flag ne soit défini. Souvent, ce drapeau est placé dans un `fichierclass/*.var` en utilisant le paramètre `flag_initial=1`.

7 FAI installe votre planification

7.1 La première partie d'une installation

Après le démarrage du noyau, il monte le système de fichiers racine via NFS à partir du serveur d'installation et démarre le script `/usr/sbin/fai` ¹⁰. Ce script contrôle la séquence de l'installation. Aucun autre script dans `/etc/init.d/` n'est utilisé.

L'espace de configuration est rendu disponible via la méthode configurée (un montage NFS par défaut) du serveur d'installation au chemin défini dans `$FAI` ¹¹.

7.2 Messages de boot

Lorsque vous démarrez le client d'installation à partir de la carte réseau avec PXE, vous obtiendrez des messages comme ceci:

```
Managed PC Boot Agent (MBA) v4.00
Pre-boot eXecution Environment (PXE) v2.00
DHCP MAC ADDR: 00 A2 A3 04 05 06
DHCP.../

CLIENT MAC ADDR: 00 A2 A3 04 05 06  GUID: 3D6C4552
CLIENT IP: 192.168.33.100 MASK: 255.255.255.0  DHCP IP: 192.168.33.250
GATEWAY IP: 192.168.33.1

!PXE entry point found (we hope) at 9854:0106 via plan A
UNDI code segment at: 9854 len 5260
UNDI data segment at: 921D len 63A2
Getting cached packet  01 02 03
My Ip address seems to be C0A82164 192.168.33.100
ip=192.168.33.100:192.168.33.250:192.168.33.1:255.255.255.0
BOOTIF=01-00-A2-A3-04-05-06
SYSUUID=
TFTP prefix: fai/
Trying to load pxelinux.cfg/C0A82164

Loading vmlinuz-3.16.0-4-amd64.....
Loading initrd.img-3.16.0-4-amd64.....ready.
```

À ce stade, le client d'installation a réussi à recevoir le réseau Config via DHCP et le noyau et `initrd` via TFTP. Il démarre maintenant Le noyau Linux et `initrd`. Si tout allait bien, l'`initrd` Monte `nfsroot` ¹² Et les scripts FAI sont lancés. La première chose que vous voyez est le message en rouge de copyright FAI.

¹⁰ Puisque le système de fichiers racine sur les clients est monté via NFS, `fai` est localisé in `/srv/fai/nfsroot/usr/sbin` sur le serveur d'installation.

¹¹ `$FAI` est une variable interne utilisée par les scripts FAI. Par défaut, le chemin est `/var/lib/fai/config`.

¹² `/srv/fai/nfsroot` depuis le serveur d'installation via NFS

```

-----
Fully Automatic Installation - FAI

5.3.3~bpo8+2 (c) 1999-2017
Thomas Lange <lange@informatik.uni-koeln.de>
-----

Calling task_confdir
Kernel currently running: Linux 3.16.0-4-amd64 x86_64 GNU/Linux
Kernel parameters: BOOT_IMAGE=vmlinuz-3.16.0-4-amd64 initrd=initrd.img-3.16.0-4-amd64 \
rw aufs ip=dhcp root=192.168.33.250:/srv/fai/nfsroot FAI_FLAGS=verbose,sshd,createvt \
FAI_CONFIG_SRC=nfs://faiserver/srv/fai/cskoeln FAI_ACTION=install quiet \
BOOTIF=01-00-a2-a3-04-05-06
Reading /tmp/fai/boot.log
FAI_FLAGS: verbose sshd createvt
Setting SERVER=faiserver. Value extracted from FAI_CONFIG_SRC.
FAI_CONFIG_SRC is set to nfs://faiserver/srv/fai/config
Configuration space faiserver:/srv/fai/config mounted to /var/lib/fai/config
Calling task_setup
FAI_FLAGS: verbose sshd createvt
15 Jan 13:22:37 ntpdate[1533]: step time server 192.168.33.250 offset -0.342793 sec
Press ctrl-c to interrupt FAI and to get a shell
Starting FAI execution - 20170115_132237
Calling task_defclass
fai-class: Defining classes.
Executing /var/lib/fai/config/class/10-base-classes.
10-base-classes OK.
Executing /var/lib/fai/config/class/20-hwdetect.source.
Loading kernel module md-mod
20-hwdetect.source OK.
Executing /var/lib/fai/config/class/50-host-classes.
50-host-classes OK.
List of all classes: DEFAULT LINUX AMD64 FAIBASE DHCPC DEMO GRUB client01 LAST

```

Vous pouvez également voir la liste des classes FAI, qui sont définies pour ce hôte. Cette liste est très importante pour le reste de l'installation.

La première tâche est appelée *confdir*, qui est chargée de l'accès à l'espace de configuration. Ici, nous utilisons un montage NFS à partir de l'installation. Comme vous pouvez le voir sur la console (et plus tard dans les journaux).

```

FAI_CONFIG_SRC is set to nfs://faiserver/srv/fai/config
Configuration space faiserver:/srv/fai/config mounted to /var/lib/fai/config

```

Avant de lancer l'installation ($\$FAI_ACTION=install$), l'ordinateur Bip trois fois. Donc, faites attention quand vous entendez trois bips mais vous ne voulez pas effectuer une installation et laisser FAI effacer toutes vos données sur le disque local!

7.3 Redémarrage de l'ordinateur dans le nouveau système

Pour redémarrer l'ordinateur pendant ou à la fin de l'installation, vous devez utiliser la commande *faireboot* en faveur de la commande de redémarrage normal. Utilisez aussi *faireboot* si vous êtes connecté depuis la télécommande. Si l'installation n'est pas terminée, utilisez *faireboot -s*, donc les fichiers journaux sont également copiés sur le serveur d'installation.

Si l'installation est terminée, l'ordinateur doit démarrer un petit système Debian. Vous pouvez vous connecter en tant que *demo* ou *root* avec le mot de passe *fai*.

7.4 Démarrage de FAI (tâche confdir)

Une fois le client d'installation démarré, seul le script `/usr/sbin/fai` est exécuté. Il effectuera une initialisation minimale. La variable `$FAI_CONFIG_SRC` ¹³ est utilisée pour accéder à l'espace de configuration FAI qui est alors disponible dans le répertoire `$FAI` ¹⁴. FAI ne se déroulera pas sans l'espace de configuration.

7.5 Définition de classes et de variables (tâches defclass et defvar)

La commande `fai-class(1)` exécute des scripts dans `$FAI/class` pour définir des classes. Si les scripts écrivent une chaîne sur `stdout`, cela sera défini comme une classe. Lisez tous les détails dans la page de manuel de `fai-class(1)`.

Après avoir défini les classes, chaque fichier correspondant à `.var` avec un préfixe qui correspond à une classe définie provient de variables définies. Il doit contenir le code shell `vaild`.

7.6 Partitionnement de disques locaux, création de systèmes de fichiers (tâches de partitionnement)

Pour le partitionnement du disque, un fichier de configuration de disque de `$FAI/disk_config` est sélectionné à l'aide de classes.

Le format de la configuration du disque est similaire à un fichier `fstab`.

L'outil de partitionnement `setup-storage(8)` exécute toutes les commandes nécessaires à la création de la disposition de la partition du disque, du RAID logiciel, du LVM et de la création des systèmes de fichiers. Lisez la page de manuel de `setup-storage(8)` pour une description détaillée et quelques exemples du format.

7.7 Préréglage Debconf (tâche debconf)

Les fichiers dans `$FAI/debconf` sont utilisés par `debconf(7)` habituel en présélectionnant si les noms de fichier correspondent à un nom de classe.

7.8 Installation de progiciels (tâche instsoft)

La commande `install_packages(8)` lit les fichiers de configuration à partir de `$FAI/package_config` en classe et installe des progiciels sur le nouveau système de fichiers.

Il installe les paquets en utilisant `apt-get(8)`, `aptitude(1)`, `yum` ou d'autres outils de paquetage sans aucune interaction manuelle nécessaire. Les paquets sont également résolus par les outils de paquets.

Le format des fichiers de configuration est décrit dans [\[packageconfig\]](#).

7.9 Personnalisation spécifique au site (task configure)

Souvent, les configurations par défaut des progiciels ne répondent pas à vos besoins spécifiques au site. Vous pouvez appeler des scripts arbitraires qui ajustent la configuration du système. Par conséquent, la commande `fai-do-scripts(1)` exécute des scripts dans `$FAI/scripts` d'une manière basée sur la classe. Il est possible d'avoir plusieurs scripts de différents types (`shell`, `cfengine`, ...) à exécuter pour une classe.

L'ensemble de scripts par défaut dans `$FAI/scripts` inclut des exemples d'installation de machines Debian et CentOS. Ils définissent le mot de passe `root`, ajoutent un compte utilisateur démo, paramètrent le fuseau horaire, configurent le réseau pour DHCP ou utilisent une adresse IP fixe, la configuration `grub` et plus encore. Ils devraient faire un travail raisonnable pour votre installation. Vous pouvez les modifier ou ajouter de nouveaux scripts pour répondre à vos besoins locaux.

Plus d'informations sur ces scripts sont décrits dans [\[cscripts\]](#).

¹³ Il a été défini sur la ligne de commande du noyau

¹⁴ `/var/lib/fai/config`

7.10 Enregistrement des fichiers journaux (tâche savelog)

Lorsque toutes les tâches sont terminées, les fichiers journaux sont écrits dans `/var/log/fai/$HOSTNAME/install/`¹⁵ sur le nouveau système et sur le compte sur le serveur d'installation si `$LOGUSER` est défini. Il est également possible de spécifier un autre hôte comme enregistrement en enregistrant la destination via la variable `$LOGSERVER`. Si `$LOGSERVER` n'est pas défini, FAI utilise la variable `$SERVER` qui n'est définie que lors d'une installation initiale (par `get-boot-info`). Assurez-vous de définir `$LOGSERVER` dans un script `_class/*.var` si vous utilisez l'action `softupdate`.

De plus, deux liens symboliques seront créés pour indiquer le dernier répertoire écrit. Le lien symbolique `last` pointe vers le répertoire journal de la dernière action FAI exécutée. Les liens symboliques `last-install` et `last-sysinfo` pointent vers le répertoire avec la dernière action correspondante. Par défaut, les fichiers journaux seront copiés sur le serveur de journal à l'aide de `scp`. Vous pouvez utiliser la variable `$FAI_LOGPROTO` dans le fichier `fai.conf(5)` pour choisir une autre méthode d'enregistrement des journaux sur le serveur distant. Voici un exemple de structure de lien symbolique:

```
lrwxrwxrwx 1 fai fai 23 Dec 2 2013 last-sysinfo -> sysinfo-20131202_161237
drwxr-xr-x 2 fai fai 4096 Dec 2 2013 sysinfo-20131202_161237
drwxr-xr-x 2 fai fai 4096 Feb 14 2014 install-20140214_142150
drwxr-xr-x 2 fai fai 4096 Dec 2 11:47 install-20141202_113918
lrwxrwxrwx 1 fai fai 23 Dec 4 13:22 last-install -> install-20141204_131351
lrwxrwxrwx 1 fai fai 23 Dec 4 13:22 last -> install-20141204_131351
drwxr-xr-x 2 fai fai 4096 Dec 4 13:22 install-20141204_131351
```

Vous trouverez des exemples de fichiers journaux à l'adresse <http://fai-project.org/logs>.

7.11 Redémarrez le nouveau système installé

Avant de redémarrer, le client d'installation appelle `fai-chboot -d <hostname>` sur le serveur d'installation, pour désactiver sa propre configuration PXELINUX. Sinon, il redémarrera l'installation lors de la prochaine initialisation. Normalement, cela devrait démarrer le nouveau système installé à partir de son second périphérique d'amorçage, le disque dur local.

À la fin, le système est automatiquement redémarré si "reboot" a été ajouté à `$FAI_FLAGS`.

8 Chapitre avancés de FAI

8.1 Vérification des paramètres reçus des serveurs DHCP

Si le client d'installation démarre, vous pouvez vérifier si toutes les informations provenant du démon DHCP sont correctement reçues. Les informations reçues sont écrites dans `/tmp/fai/boot.log`. Un exemple de résultat d'une requête DHCP peut être trouvé dans les fichiers journaux d'exemple.

8.2 Surveillance de plusieurs installations clientes

Vous pouvez surveiller l'installation de tous les clients d'installation avec la commande `fai-monitor(8)`. Tous les clients vérifient si ce démon est en cours d'exécution sur le serveur d'installation (ou sur l'ordinateur défini par la variable `$monserver`). Chaque fois qu'une tâche démarre ou se termine, un message est envoyé. Le démon du moniteur FAI imprime ces messages à la sortie standard. Il ya aussi un frontend graphique disponible, appelé `fai-monitor-gui(1)`.

```
$ fai-monitor | fai-monitor-gui - &
```

¹⁵ `/var/log/fai/localhost/install/` est un lien vers ce répertoire.

8.3 Collecte d'adresses Ethernet pour plusieurs hôtes

Vous devez collecter toutes les adresses Ethernet (MAC) des clients à l'installation et affecter un nom d'hôte et une adresse IP à chaque client. Pour collecter les adresses MAC, démarrez vos clients pour l'installation. Vous pouvez déjà le faire avant que n'importe quel démon DHCP s'exécute dans votre sous-réseau. Ils échoueront à démarrer (en raison du manque de DHCP ou de TFTP manquant), mais vous pouvez toujours collecter les adresses MAC.

Pendant que les clients d'installation démarrent, ils envoient des paquets de diffusion au LAN. Vous pouvez enregistrer les adresses MAC de ces hôtes en exécutant simultanément la commande suivante sur le serveur:

```
faiserver# tcpdump -qtel broadcast and port bootpc >/tmp/mac.list
```

Une fois que les hôtes ont été envoyés, certains paquets de diffusion annule `tcpdump` en tapant `ctrl-c`. Vous obtenez une liste de toutes les adresses MAC uniques avec ces commandes:

```
faiserver$ perl -ane 'print "\U$F[0]\n"' /tmp/mac.list|sort|uniq
```

Après cela, vous n'avez qu'à assigner ces adresses MAC aux noms d'hôte et aux adresses IP (`/etc/ethers` et `/etc/hosts` ou aux cartes NIS correspondantes). Avec ces informations, vous pouvez configurer votre démon DHCP (voir la section [bootdhcp]). ¹⁶

8.3.1 Débogage du trafic réseau

Si le client ne peut démarrer correctement à partir de la carte réseau, utilisez `tcpdump` (8) pour rechercher des paquets Ethernet entre le serveur d'installation et le client. Recherchez également les entrées de plusieurs fichiers journaux effectués par `tftpd` (8) et `dhcpd` (8) :

```
faiserver$ egrep "tftpd|dhcpd" /var/log/*
```

8.4 Détails du démarrage PXE

Ici, nous décrivons les détails du démarrage PXE, qui ne sont nécessaires que si vous avez des problèmes lors du démarrage de vos clients d'installation.

Presque toutes les cartes réseau modernes prennent en charge l'environnement de démarrage PXE. PXE est l'environnement d'exécution de pré-lancement. Cela nécessite le chargeur de démarrage PXELINUX et une version spéciale du démon TFTP, disponible dans les paquets Debian `pxelinux` et `tftpd-hpa`. Le démarrage PXE nécessite également un serveur DHCP, afin que la carte réseau puisse configurer ses paramètres IP. Il s'agit de la séquence d'une amorce PXE:

- La carte réseau du client envoie son adresse MAC
- Le serveur DHCP répond à la configuration IP du client
- La carte réseau configure son IP
- Le client d'installation obtient le binaire `pxelinux.0` via TFTP
- Obtenez le fichier de configuration `pxelinux.cfg/C0A8210C` via TFTP
- C0A8210C est l'adresse IP du client en hexadécimal
- Cette configuration contient le noyau, `initrd` et les paramètres de ligne de commande supplémentaires du noyau, qui a été créé par `fai-chboot`.
- Obtenez le noyau et `initrd` via TFTP.

Exemple d'un fichier `pxelinux.cfg`:

¹⁶ Je recommande d'écrire les adresses MAC (les trois derniers octets suffiront si vous avez des cartes réseau du même fournisseur) et le nom d'hôte à l'avant de chaque châssis.

```
default fai-generated

label fai-generated
kernel vmlinuz-3.16.0-4-amd64
append initrd=initrd.img-3.16.0-4-amd64 ip=dhcp root=/srv/fai/nfsroot aufs FAI_FLAGS= ↵
verbose,sshd,createvt FAI_CONFIG_SRC=nfs://faiserver/srv/fai/config FAI_ACTION=install
```

Voir */usr/share/doc/syslinux/pxelinux.doc* pour des informations plus détaillées sur PXELINUX. Il existe un nouveau binaire *lpxelinux* qui prend également en charge le chargement du noyau et de *l'initrd* via FTP ou HTTP. La commande *fai-chboot(8)* prend en charge cette option avec l'option *-U*.

8.5 Personnalisation de la configuration de votre serveur d'installation

- Miroir de paquetage local/plus rapide
- Loguser différent
- Local root password dans *nfsroot*

La configuration du paquet FAI (et non les données de configuration pour les clients d'installation) est définie dans *fai.conf(5)*. Les définitions qui sont utilisées uniquement pour créer le *nfsroot* sont situées dans *nfsroot.conf(5)*. Vérifiez ces variables importantes dans *nfsroot.conf* avant d'appeler *fai-setup* ou *fai-make-nfsroot*.

FAI_DEBOOTSTRAP

La construction de *nfsroot* utilise la commande *debootstrap(8)*. Il a besoin de l'emplacement d'un miroir Debian et du nom de la distribution (*wheezy*, *jessie*, *stretch*, *sid*) pour lequel le système Debian de base devrait être construit. N'utilisez pas de distributions différentes ici et dans */etc/fai/apt/sources.list*. Cela créera un *nfsroot* brisé.

NFSROOT_ETC_HOSTS

Cette variable n'est nécessaire que si les clients n'ont pas accès à un serveur DNS. Cette variable multiligne est ajoutée à */etc/hosts* dans le *nfsroot*. Ensuite, les clients d'installation peuvent accéder à ces hôtes par leur nom sans utiliser DNS.

Le contenu de */etc/fai/apt/sources.list* est utilisé par le serveur d'installation et par les clients. Si votre serveur d'installation a plusieurs cartes réseau et différents noms d'hôte pour chaque carte (comme pour un serveur *Beowulf*), utilisez le nom du serveur d'installation qui est connu des clients d'installation.

Si vous avez des problèmes lors de l'exécution de *fai-setup*, ils proviennent habituellement de *fai-make-nfsroot(8)* qui est appelé par la commande précédente. L'ajout de *-v* vous donne une sortie plus détaillée qui vous aide à repérer l'erreur. La sortie est écrite dans */var/log/fai/fai-make-nfsroot.log*.¹⁷

L'installation crée également le compte *fai* (défini par *\$LOGUSER*) s'il n'est pas déjà disponible. Vous pouvez donc ajouter un utilisateur avant d'appeler *fai-setup(8)* à l'aide de la commande *adduser(8)* et utiliser ce compte local pour enregistrer des fichiers journaux. Les fichiers journaux de tous les clients d'installation sont enregistrés dans le répertoire de base de ce compte. Vous devriez changer le groupe principal de ce compte, donc ce compte a des droits d'écriture sur */srv/tftp/fai* pour appeler *fai-chboot* pour créer la configuration PXE pour les hôtes.

Lorsque vous apportez des modifications à *fai.conf*, *nfsroot.conf*, le *nfsroot* doit être reconstruit en appelant *fai-make-nfsroot(8)*. Si vous souhaitez uniquement installer un nouveau paquet kernel sur *nfsroot*, ajoutez les flags *-k* ou *-K* à *fai-make-nfsroot*. Cela ne recréera pas votre *nfsroot*, mais ne mettra à jour que vos noyaux et les modules du noyau dans le *nfsroot* ou ajoutera des paquets supplémentaires dans le *nfsroot*.

8.6 Création d'un CD FAI ou d'une clé USB

Vous pouvez facilement créer un CD d'installation (ou clé USB) pour votre installation réseau. Cela permettra d'effectuer la même installation et la même configuration à partir du CD sans avoir besoin du serveur d'installation. Par conséquent, vous

¹⁷ Pour le débogage, il peut être utile d'entrer l'environnement *chroot* manuellement à l'aide de cette commande. *faiserver# chroot /srv/fai/nfsroot bash*

devez créer un miroir partiel de tous les paquets Debian nécessaires à vos classes FAI (à l'aide de `fai-mirror(1)`). Ensuite, la commande `fai-cd(8)` mettra ce miroir, le `nfsroot` et l'espace de configuration sur un CD amorçable. C'est tout!

Ce CD d'installation contient toutes les données nécessaires à l'installation. La commande `fai-cd(8)` place le `nfsroot`, l'espace de configuration et un sous-ensemble du miroir Debian sur un CD-ROM. Un miroir de paquets partiel est créé à l'aide de la commande `fai-mirror(1)` qui contient tous les paquetages utilisés par les classes utilisées dans votre espace de configuration. Un échantillon d'image ISO est disponible à l'adresse <http://fai-project.org/fai-cd>.

Avec la commande `dd(1)`, vous pouvez également créer une clé USB bootable en écrivant simplement le contenu du fichier ISO sur votre clé USB (ici le stick est `/dev/sdf`).

```
faiserver# dd if=fai-cd.iso of=/dev/sdf bs=1M
```

Il ne s'agit pas d'un live CD du serveur d'installation.

8.7 Création d'images de disque VM à l'aide de FAI

En utilisant la commande `fai-diskimage(8)`, vous pouvez créer des images de disques de machines Linux qui peuvent être utilisées avec une machine virtuelle comme KVM, VMware, VirtualBox ou un service cloud comme OpenStack, GCE, EC2 et autres. Le processus d'installation exécute les tâches FAI normales sur une image de disque brut. Après l'installation, vous pouvez démarrer l'image disque et avoir un système en cours d'exécution. L'image disque peut également être convertie au format `qcow2`.

```
faiserver# export FAI_BASEFILEURL=http://fai-project.org/download/basefiles/
faiserver# fai-diskimage -u cloud3 -S 2G -cDEBIAN,JESSIE64,AMD64,FAIBASE,GRUB_PC,CLOUD,GCE ←
disk.raw
```

Crée le fichier `disk.raw` pour un hôte appelé `cloud3`, avec un petit ensemble de progiciels.

```
# export FAI_BASEFILEURL=http://fai-project.org/download/basefiles/
# cl=DEFAULT,DHCP,DEBIAN,AMD64,FAIBASE,GRUB_PC,UBUNTU,XENIAL,XENIAL64,XORG
# fai-diskimage -v -u foobar -S5G -c$cl ubuntu.qcow2
```

Crée une image de disque appelée `ubuntu.qcow2` pour un bureau Ubuntu 16.04 avec un nom d'hôte défini sur `foobar`.

Vous ne devez pas configurer le `nfsroot` lorsque vous utilisez uniquement `fai-diskimage`. Mais vous avez besoin d'un fichier de base dans votre espace de configuration. Vous pouvez télécharger une image de base Debian à partir de <http://fai-project.org/download/basefile> et copier ceci dans votre espace de configuration. Si vous avez déjà configuré le `nfsroot`, vous pouvez copier le fichier de base Debian depuis le `nfsroot` dans votre espace de configuration à l'aide de cette commande:

```
$ cp /srv/fai/nfsroot/var/tmp/base.tar.xz
$ /srv/fai/config/basefiles/JESSIE64.tar.xz
```

8.8 Système de sauvetage FAI

Si vous définissez la variable `$FAI_ACTION` sur `sysinfo` (par exemple en utilisant `fai-chboot -S`), le client n'installera pas de nouveau système, mais collectera beaucoup d'informations système. Si vous définissez `$FAI_ACTION` sur `inventory`, vous ne recevrez que quelques informations sur le matériel. Les deux actions peuvent être utilisées pour FAI comme un système de sauvetage.

Tapez `ctrl-c` pour obtenir un shell ou utilisez `Alt-F2` ou `Alt-F3` et vous obtiendrez un autre terminal de console, si vous avez ajouté `createvt` à `$FAI_FLAGS`.

Vous avez maintenant un système Linux en cours d'exécution sur le client d'installation sans utiliser le disque dur local. Utilisez-le comme système de secours si votre disque local est endommagé ou si l'ordinateur ne peut pas démarrer correctement à partir du disque dur. Vous obtiendrez un shell et vous pouvez exécuter diverses commandes (`dmesg`, `lsmmod`, `df`, `lspci`, ...). Regardez le fichier `journal` dans `/tmp/fai`. Vous y trouverez de nombreuses informations sur le processus d'amorçage.

FAI monte tous les systèmes de fichiers qu'il trouve sur les disques locaux en lecture seule. Il vous indique également sur quelle partition un fichier `/etc/fstab` existe. Lorsqu'une seule table de système de fichiers est trouvée, les partitions sont montées selon ces informations. Voici un exemple:

```

demohost:~# df
Filesystem      1K-blocks      Used Available Use% Mounted on
rootfs          4099064        414088   3645296   11% /
192.168.33.250:/srv/fai/nfsroot
                3905600        410976   3454944   11% /live/image
tmpfs           193464         3112     190352    2% /live/cow
aufs            4099064        414088   3645296   11% /

192.168.33.250:/srv/fai/config
                3905600        410976   3454944   11% /var/lib/fai/config
/dev/sda1       241116         74519    154149    33% /target
/dev/sda9       4364212       139888   4179988    4% /target/home
/dev/sda7       553376        16840    536536    4% /target/tmp
/dev/sda8       2221628       275936   1832840   14% /target/usr
/dev/sda6       577096        172924   374856    32% /target/var

```

Cette méthode peut être utilisée comme un environnement de secours! Si vous avez besoin d'un système de fichiers avec accès en lecture/écriture, utilisez la commande `rwmount`:

```
demohost# rwmount /target/home
```

8.9 FAI sans NFS

Pour démarrer dans FAI et commencer la séquence d'installation sans utiliser le protocole NFS. Vous démarrez la machine cliente en utilisant PXE comme d'habitude, puis récupérez une image contenant le `nfsroot` via `http`.

Pour créer une image, utilisez l'argument `-S` de `fai-cd`

```
faiserver# fai-cd -S squash.img
```

Déplacez cette image vers un répertoire à partir duquel elle peut être demandée via `http` (généralement un répertoire desservi par le serveur web)

Pour demander maintenant l'image `squashfs`, ajoutez ce qui suit à votre ligne de commande du noyau, `p`. Dans votre fichier de configuration `pxelinux` pour le client.

```
root=live:http://faiserver/cksoeln/squash.img
```

Remplacez `faiserver` par le nom de domaine ou IP de la machine à laquelle votre image de `squash` est servie.

8.10 Installation d'autres distributions à l'aide d'un `nfsroot` Debian

Vous pouvez installer toutes sortes de distributions Linux à partir d'un seul `nfsroot` Debian. Par conséquent, vous devez créer un fichier `base.tar.xz` de la distribution que vous souhaitez installer et le placer dans le répertoire `basefiles`. Puis nommez-le `UBUNTU1404.tar.xz` par exemple. Un client d'installation appartenant à la classe `UBUNTU1404` extrait ensuite ce fichier de base dans son système de fichiers vide. De plus, vous devez ajuster les `sources.list` ou les fichiers de configuration similaires nécessaires pour spécifier l'emplacement du référentiel de paquets.

L'outil `rinse` (8) est utilisé pour créer des fichiers de base pour la distribution comme `CentOS`, `openSUSE`, `Scientific Linux`, `Cern` ou `Fedora`. Certains fichiers de base peuvent être téléchargés à partir de <http://fai-project.org/download/basefiles/>.

Le script `mk-basefile` dans `/usr/share/doc/fai-doc/examples/simple/basefiles/` aide à créer ces fichiers de base.

8.11 Création d'environnements `chroot` et virtualiser

Si vous devez créer certains environnements `chroot`, ou un environnement de virtualisation où vous ne pouvez ni ne voulez exécuter un programme d'installation Debian normal pour accéder à un système opérationnel (par exemple, les domaines hôtes `Xen`), il ya l'action FAI `dirinstall`. En appelant

```
faiserver# fai <options> dirinstall <target-directory>
```

Et en utilisant l'option `-c <classes>` ou `-N` vous obtenez une installation FAI, sans l'action de partitionnement, directement dans le répertoire cible. Le nom d'hôte de l'installation cible peut être spécifié à l'aide de `-u <host-name>`

Ceci, par exemple, peut être utilisé pour combiner FAI avec l'outil *xen-tools*, qui vous aide à construire des domaines invités Xen. *xen-tools* est très agréable pour générer des fichiers de configuration et bloquer des périphériques pour de nouveaux invités basés sur des commandes simples et/ou des fichiers de configuration, mais ils ne peuvent assigner qu'un seul rôle par installation pour la personnalisation. Les FAI-utilisateurs ont besoin et veulent plus, car ils sont utilisés pour avoir le système de classe. Ils les obtiennent même dans les installations *xen-tools*, en utilisant le code suivant en tant que rôle *xen-tools* script:

```
#!/bin/sh
TARGET=$1
CMD="fai -N -v -u ${hostname} dirinstall $TARGET"
echo running $CMD
$CMD
```

Ensuite, vous voulez définir la variable `install=0` de la configuration *xen-tools* pour cet hôte.

8.12 Utilisation de FAI pour les mises à jour

FAI peut également effectuer des mises à jour de systèmes déjà en cours d'exécution, sans réinstallation à partir de zéro. C'est ce qu'on appelle *softupdate*. Un FAI *softupdate* ignore les tâches qui ne sont pas adaptées à la mise à jour d'un système en cours d'exécution, comme le partitionnement des disques durs et la création de systèmes de fichiers. Au lieu de cela, il exécute uniquement les tâches de mise à jour et d'installation des progiciels et de l'appel des scripts de personnalisation.

Pour exécuter un appel *softupdate*:

```
# fai -v -s nfs://faiserver/srv/fai/config softupdate
```

Par défaut, un *softupdate* utilise la liste des classes définies lors de l'installation initiale. Assurez-vous de définir la variable `$LOGSERVER` (effectuée dans un fichier `class/*.var`) si FAI doit enregistrer les fichiers journaux sur une machine distante.

C'est à vous, comment démarrer un *softupdate* sur un plus grand nombre d'hôtes. Vous pouvez faire le *softupdate* sur une base régulière via *cron* ou vous pouvez utiliser des outils comme `clusterssh(1)` pour démarrer un *softupdate* via un push sur une liste d'hôtes.

Gardez à l'esprit que les scripts de personnalisation sont exécutés chaque fois que vous faites un *softupdate*. Cela signifie qu'ils doivent être **idempotents**, c'est-à-dire que le résultat de leur fonctionnement doit toujours produire le même résultat, même lorsqu'ils fonctionnent plus d'une fois.

Par exemple, l'ajout d'une ligne à un fichier ne doit pas se faire via ce code:

```
$ echo "some strings" >> /etc/fstab
```

Utilisez plutôt la commande `ainsl(1)` dans un script shell ou utilisez la fonction `AppendIfNoSuchLine` de *cfengine*.

Toutes les commandes du script de personnalisation doivent être capables de modifier le système de fichiers cible s'il est disponible dans `/target` lors de l'installation initiale ou si c'est le système de fichiers normal relatif à `/` pendant le *softupdate*.

Voici quelques variables qui aident à écrire ces scripts:

\$target

Pointe vers le répertoire racine du client, qui est `/target` pendant l'installation et `/` pendant un *softupdate*.

\$FAI_ROOT

C'est la même valeur que `$target`. Pour des raisons historiques, nous avons ces deux variables dans FAI.

\$ROOTCMD

Dans le cas de l'installation, il s'agit d'un alias pour `chroot $target` en cas de *softupdate* c'est juste vide. Vous pouvez ajouter ceci aux commandes si vous avez besoin d'exécuter une commande dans le système de fichiers cible des clients via `chroot`.

§FAI_ACTION

Si vous devez appeler le code en fonction de l'action FAI effectuée, vous pouvez utiliser cette variable. Il contient l'action actuellement exécutée: *install*, *softupdate*, *dirinstall*, *sysinfo*, *inventory* ou votre propre action définie.

8.13 Comment installer un système d'exploitation 32 bits à partir d'un système d'exploitation 64 bits

Pour installer un ordinateur avec un système d'exploitation 32 bits, vous avez besoin d'un nfsroot i386. La création de cette nfsroot 32 bits sur un serveur d'installation exécutant amd64 est assez simple. Installez et configurez les paquets FAI. Copiez ensuite vos fichiers de configuration FAI dans un nouveau sous-répertoire.

```
faiserver# cp -a /etc/fai /etc/fai-i386
```

Modifiez la variable `§FAI_DEBOOTSTRAP_OPTS` dans `/etc/fai-i386/nfsroot.conf` et ajoutez l'option `--arch i386`. Choisissez également un répertoire différent pour votre nouveau nfsroot. Voici les deux lignes après l'édition.

```
NFSROOT=/srv/fai/nfsroot-i386
FAI_DEBOOTSTRAP_OPTS="--arch i386 --exclude=info --include=aptitude"
```

Appelez maintenant `fai-make-nfsroot` qui crée le nfsroot 32 bits dans `/srv/fai/nfsroot-i386`

```
faiserver# fai-make-nfsroot -v -C/etc/fai-i386
```

La création d'un miroir partiel utilisant `fai-mirror` (1) nécessaire à un CD amorçable ou une clé USB est également possible sur une architecture différente. Vous devez spécifier l'architecture lors de l'appel de `fai-mirror`.

```
$ fai-mirror -m800 -B -a i386 -v -cDEFAULT,DEBIAN,FAIBASE,I386 /srv/mirror-i386
```

C'est tout!

9 Divers conseils et détails

9.1 La liste des tâches

La plupart des tâches de l'installation sont définies comme des sous-routines qui sont définies dans `/usr/lib/fai/subroutines` (par exemple `task_instsoft`). Certains sont des scripts shell externes situés dans `/usr/lib/fai/`. Ils sont appelés via un sous-programme supérieur appelé `task`. Ce sous-programme appelle les hooks si disponibles, puis appelle la tâche (définie comme `task<name>_()`). Une tâche et ses hooks peuvent être ignorés à la demande en utilisant la commande `skiptask()`.

Suit maintenant la description de toutes les tâches, énumérées dans l'ordre dans lequel elles sont exécutées.

confdir

Les paramètres ajoutés au noyau peuvent définir des variables, le démon `syslog` est démarré. La liste des périphériques réseau est stockée dans `§netdevices`. Ensuite, des paramètres supplémentaires sont extraits d'un serveur DHCP. Le fichier de configuration du résolveur DNS est créé.

L'emplacement de l'espace de configuration est défini par la variable `§FAI_CONFIG_SRC`.

Ensuite, le fichier `§FAI/hooks/subroutines` est sourcé s'il existe. En utilisant ce fichier, vous pouvez définir vos propres sous-programmes ou remplacer la définition des sous-programmes FAI.

setup

Cette tâche définit l'heure du système, tous les `§FAI_FLAGS` sont définis et deux terminaux virtuels supplémentaires sont ouverts à la demande. Un démon de shell sécurisé est lancé à la demande pour les connexions à distance.

defclass

Appelez `fai-class` (1) pour définir des classes à l'aide de scripts et de fichiers dans `§FAI/class` et classes de `/tmp/fai/additional_classes` et la variable `§ADDCLASSES`. La liste de toutes les classes définies est stockée dans la variable `§classes` et enregistrée dans `/tmp/fai/FAI_CLASSES`.

defvar

Sourcez tous les fichiers *\$FAI/class/*.var* pour chaque classe définie. Si un hook a écrit quelques définitions de variables dans le fichier *\$LOGDIR/additional.var*, ce fichier est également sourcé.

action

En fonction de la valeur de *\$FAI_ACTION*, ce sous-programme décide de l'action FAI à exécuter. Les actions disponibles par défaut sont: *sysinfo*, *install*, *inventory*, *dirinstall* et *softupdate*. Si *\$FAI_ACTION* a une autre valeur, une action définie par l'utilisateur est appelée si un fichier *\$FAI/hooks/\$FAI_ACTION* existe. Ainsi, vous pouvez facilement définir vos propres actions.

sysinfo

Appelée lorsque aucune installation n'est effectuée mais que l'action est *sysinfo*. Il affiche des informations sur le matériel détecté et monte les disques durs locaux en lecture uniquement sur */target/partitionname* ou en regard d'un fichier *fstab* trouvé à l'intérieur d'une partition. Les fichiers journaux sont stockés sur le serveur d'installation.

inventory

Une courte liste des informations système est imprimée.

install

Cette tâche contrôle la séquence d'installation. Vous entendrez trois bips avant le début de l'installation. Le travail principal consiste à appeler d'autres tâches et à enregistrer la sortie dans */tmp/fai/fai.log*. Si vous avez des problèmes pendant l'installation, regardez tous les fichiers dans */tmp/fai/*. Vous trouverez des exemples de fichiers journaux à l'adresse <http://fai-project.org/logs/>.

dirinstall

Installez dans un répertoire, et non pas sur un disque local. Utilisez-le pour créer des environnements chrootés.

softupdate

Cette tâche, exécutée à l'intérieur d'un système en cours d'exécution via l'interface de ligne de commande *fai(8)*, effectue un *softupdate*. Voir le chapitre [\[softupdate\]](#) pour plus de détails.

partition

Appelle *setup-storage(8)* pour partitionner les disques durs et créer des systèmes de fichiers. La tâche écrit des définitions de variables pour la partition et le périphérique racine et de démarrage (*\$ROOT_PARTITION*, *\$BOOT_PARTITION*, *\$BOOT_DEVICE*) dans */tmp/fai/disk_var.sh* et crée un fichier *fstab* pour le nouveau système.

mountdisks

Montez les partitions créées en fonction du fichier */tmp/fai/fstab* créé par rapport à *\$FAI_ROOT*.

extrbase

Extrait un système minimal après lequel un chroot peut y être introduit. Par défaut, le fichier tar base */var/tmp/base.tar.xz* sera extrait. Les fichiers correspondant à un nom de classe dans *\$FAI/basefiles/* sont également utilisés pour décompresser un autre fichier tar selon les classes définies. Cela peut être utilisé pour installer des distributions Linux différentes de celles utilisées pour créer le *nfsroot*. Le fichier par défaut *base.tar.xz* est un instantané d'un système Debian de base créé par *debootstrap(8)*. Cette tâche utilise la variable *FAI_BASEFILEURL* pour extraire le fichier de base via FTP ou HTTP si elle est définie.

debconf

Appelle *fai-debconf(1)* pour définir les valeurs de la base de données de préconfiguration de *debconf*.

repository

Préparez l'accès au référentiel de paquets en préparant la configuration *apt*. Cela peut également ajouter des clés de référentiel via *apt-key(8)* en classe à partir de fichiers comme *CLASSNAME.asc* dans le répertoire *package_config*.

updatebase

Met à jour les paquets de base du nouveau système et met à jour la liste des paquets disponibles. Il falsifie également certaines commandes (appelées *diversions*) à l'intérieur du nouveau système installé à l'aide de *dpkg-divert(8)*, de sorte qu'aucun démon ne sera démarré pendant l'installation.

instsoft

Installe les progiciels souhaités en utilisant des fichiers de classe dans *\$FAI/package_config/*.

configure

Appelle les scripts dans `$FAI/scripts/` et ses sous-répertoires pour chaque classe définie.

tests

Appelle les scripts de test dans `$FAI/tests/` et ses sous-répertoires pour chaque classe définie.

finish

Démonte tous les systèmes de fichiers dans le nouveau système installé et supprime les diversions de fichiers à l'aide de la commande `fai-divert`.

chboot

Modifie la configuration PXE d'un hôte sur le serveur d'installation qui indique quelle configuration PXELINUX doit être chargée lors de la prochaine initialisation à partir de la carte réseau via TFTP. Par conséquent, la commande `fai-chboot` (8) est exécutée à distance sur le serveur d'installation.

savelog

Enregistre les fichiers journaux sur le disque local et sur le compte `$LOGUSER` sur `$LOGSERVER` (par défaut sur le serveur d'installation).

faiend

Attendez que les travaux en arrière-plan se terminent (par exemple, emacs compile des fichiers lisp) et redémarre automatiquement les clients d'installation ou attend la saisie manuelle avant le redémarrage.

9.2 Tests automatisés

Après l'exécution des scripts de personnalisation, FAI exécutera certains tests si disponibles. En utilisant ces tests, vous pouvez vérifier les erreurs de l'installation. Les scripts de test sont appelés via `fai-do-scripts` (1) et doivent ajouter leurs messages à `$LOGDIR/test.log`. Un module Perl comprenant des sous-routines utiles peut être trouvé dans `Faitest.pm`. Un test peut également définir une nouvelle classe pour exécuter d'autres tests lors du prochain démarrage via la variable `$ADDCLASSES`.

9.3 Découvrir automatiquement

Dans FAI 5.0, nous avons publié une fonctionnalité qui permet aux clients de rechercher le faiserver dans leur sous-réseau respectif. Cela soulève la nécessité de récupérer l'adresse MAC de chaque client et de configurer le démon DHCP.

Cela se fait en démarrant à partir d'une petite autodiscover FAI bootmedium (CD, USB, etc.), qui peut être créée via la commande:

```
faiserver# fai-cd -A autodiscover.iso
```

L'image a une taille d'environ 25 Mo et analyse le sous-réseau d'un serveur FAI. Par défaut, il affiche un menu avec tous les profils disponibles dans l'espace de configuration de la même manière que le drapeau de menu. Dans ce menu, vous pouvez sélectionner le type d'installation que vous souhaitez effectuer.

Pour que les clients puissent trouver le faiserver, le faiserver doit exécuter `fai-monitor`.

9.4 Modification du périphérique d'amorçage

La modification de la séquence d'amorçage s'effectue normalement dans la configuration du BIOS. Mais vous ne pouvez pas changer le BIOS d'un système Linux en cours d'exécution.

Ainsi, la séquence d'amorçage du BIOS restera inchangée et votre ordinateur devrait toujours démarrer en premier à partir de sa carte réseau et le deuxième périphérique d'amorçage devrait être le disque local. Ensuite, vous pouvez changer le périphérique d'amorçage du client en créant différentes configurations PXELINUX. Cela définira si une installation doit être effectuée, ou si le client doit démarrer à partir du disque local. Cela se fait à l'aide de `fai-chboot` (8).

9.5 Comment créer un miroir Debian local

Le script `mkdebmirror`¹⁸ peut être utilisé pour créer votre propre miroir Debian local. Ce script utilise la commande `debmirror(1)`. Un miroir Debian partiel pour l'architecture i386 et amd64 pour Debian 8.0 (aka jessie) sans les paquets source nécessite environ 56Go d'espace disque. L'accès au miroir via HTTP sera la méthode par défaut dans la plupart des cas. Pour afficher plus de résultats à partir du script, appelez `mkdebmirror -v`. Un compte root n'est pas nécessaire pour créer et maintenir le miroir Debian.

Pour utiliser l'accès HTTP au miroir Debian local, installez un serveur Web et créez un lien symbolique vers le répertoire local où se trouve votre miroir:

```
faiserver# apt-get install apache2
faiserver# ln -s /files/scratch/debmirror /var/www/html/debmirror
```

Créez un fichier `sources.list(5)` dans `/etc/fai/apt` qui donne accès à votre miroir Debian. Ajoutez également l'adresse IP du serveur HTTP à la variable `$NFSROOT_ETC_HOSTS` dans `nfsroot.conf` si les clients d'installation n'ont pas de résolution DNS.

9.6 Petits conseils

- Lorsque vous utilisez l'accès HTTP à un miroir Debian, la partition locale `/var` sur tous les clients d'installation doit être suffisamment grande pour conserver les paquets Debian téléchargés. N'essayez pas avec moins de 250 Moctets à moins que vous sachiez pourquoi. Vous pouvez limiter le nombre de paquets installés à la fois avec la variable `$MAXPACKAGES`.
- Vous pouvez supprimer le logo rouge sur le client d'installation en appelant simplement une fois `reset`. Il ne s'affichera pas si vous créez un fichier à l'aide de cette commande sur le serveur d'installation:

```
touch /srv/fai/nfsroot/.nocolorlogo
```

- Une liste des variables utilisées par FAI peut être trouvée à <http://wiki.fai-project.org/wiki/Variables>.
- Vous pouvez raccourcir certains scripts de personnalisation en utilisant une seule commande `fcopy fcopy -r /`.
- Si vous reconstruisez le `nfsroot`, vous allez créer une nouvelle clé hôte ssh dans le `nfsroot`. La connexion à un client d'installation peut échouer, car la clé hôte change. Vous pouvez utiliser ceci:

```
$ ssh -o StrictHostKeyChecking=no root@installclient
```

- Vous pouvez également supprimer l'entrée hôte de votre client d'installation dans votre fichier `~/.ssh/known_hosts` à l'aide de la commande `ssh-keygen -R`.
- Dans les tâches `chboot` et `savelog`, une connexion utilisant un shell sécurisé est ouverte au serveur FAI (voir [\[isavelog\]](#)). Pour garantir que cela fonctionne de manière non interactive, une entrée appropriée dans `NFSROOT/root/.ssh/known_hosts` doit être créée. Lors de l'utilisation de `fai-setup`, cela se fait automatiquement, mais il peut s'avérer nécessaire de l'éditer manuellement si le nom de votre serveur FAI n'a pas été correctement déterminé. Si vous trébuchez sur des connexions ssh qui nécessitent de taper "yes" pour accepter la clé hôte pendant l'installation, vérifiez le contenu de votre fichier `NFSROOT/root/.ssh/known_hosts`.
- Une liste de tous les disques durs locaux est stockée dans `$disklist`. Il est défini après l'appel de `set_disk_info`.
- Utilisez `fai-divert -a` si un script `postinst` appelle un programme de configuration, par exemple Le script `postinst` pour package `apache` appelle `apacheconfig`, qui nécessite une entrée manuelle. Vous pouvez fausser le programme de configuration pour que l'installation puisse être entièrement automatique.

¹⁸ Vous pouvez trouver le script dans `/usr/share/doc/fai-doc/examples/utills/` Version 5.3

- Parfois, l'installation semble s'arrêter, mais souvent il ya seulement un script postinstall d'un logiciel qui nécessite une entrée manuelle de la console. Passez à un autre terminal virtuel et regardez quel processus fonctionne avec des outils comme `top(1)` et `ps(1)`. Vous pouvez ajouter `debug` à `FAI_FLAGS` pour faire en sorte que le processus d'installation affiche toutes les sorties des scripts postinst sur la console et obtenir son entrée aussi à partir de la console.
- Comment puis-je définir des classes sur la ligne de commande du noyau?
Lisez la page de manuel de `fai-class(8)`. Si vous souhaitez définir des classes supplémentaires (par exemple A, B, C) sur la ligne de commande du noyau, ajoutez ceci: `ADDCLASSES=A,B,C`
- Comment utiliser un noyau personnalisé dans le nfsroot?
Construisez votre noyau personnalisé en construisant un paquet kernel à l'aide de `make-kpkg(8)` et utilisez l'option `--initrd`. Copiez ce paquet Debian dans un référentiel local et ajoutez-le à `/etc/fai/sources.list`. Ajoutez le nom de votre package à `/etc/fai/NFSROOT`. Ensuite appeler

```
# fai-make-nfsroot -k
```

- Puis-je utiliser un noyau 4.X?
Oui. L'utilisation de FAI 5.1 et `dracut 044+150-1 overlays` (au lieu d'`aufs`) est prise en charge, de même que le noyau 4.x. Lorsque vous utilisez Debian jessie, vous pouvez utiliser un noyau de backports ou consulter <https://lists.uni-koeln.de/pipermail/linux-fai/2016-March/011283.html>
- Comment utiliser le nfsroot comme système pour les clients sans disque?
http://wiki.fai-project.org/wiki/Use_nfsroot_for_diskless_clients
- Comment faire pour servir plusieurs arborescence nfsroot sur un serveur FAI?
Si vous souhaitez diffuser plusieurs répertoires nfsroot, vous devez créer des répertoires de configuration spécifiques dans `/etc` pour FAI, comme `/etc/fai-jessie` et `/etc/fai-stretch`. Ensuite, vous devez définir les variables `$NFSROOT` dans différents répertoires et exécuter

```
faiserver#fai-make-nfsroot -c /etc/fai-jessie
```

9.7 flag_reboot (FAI_FLAGS)

Si `flag_reboot` est défini, en ajoutant "reboot" à `$FAI_FLAGS`, votre ordinateur client redémarrera après la fin de la tâche. Ceci est vrai pour les installations de réseau ainsi que pour les installations de bootmedium.

9.8 CentOS reboot

Après l'installation, CentOS nécessite habituellement un redémarrage supplémentaire, en raison des correctifs de sécurité SELinux qui sont appliqués après l'installation.

9.9 Fichiers journaux

FAI crée plusieurs fichiers journaux. Pendant l'installation, ils sont stockés dans `/tmp/fai` sur le client d'installation lui-même. A la fin de l'installation, ils seront copiés sur le serveur d'installation (voir [\[isavelog\]](#)). Une fois le client d'installation redémarré dans son système nouvellement installé, vous pouvez trouver les journaux FAI dans `/var/log/fai`. Les fichiers journaux sont également créés lors de l'action `softupdate` ou `dirinstall`.

Sur le faiserver, vous pouvez trouver les fichiers journaux (distants) sous le répertoire `~fai`.

Les exemples de fichiers journaux des ordinateurs installés avec succès sont disponibles sur <http://fai-project.org/logs>. Ce sont quelques fichiers journaux qui sont créés par FAI.

FAI_CLASSES

Contient une liste de toutes les classes définies.

dmesg.log

La sortie de la commande `dmesg`. Contient des messages utiles de la mémoire tampon du noyau.

fai.log

Le fichier journal principal. Contient toutes les informations importantes. Vous devez **toujours** lire ce fichier.

boot.log

Une liste de variables de paramètres de réseau, principalement définis par le démon DHCP.

format.log

Sortie de l'outil de partition `setup-storage(8)`.

shell.log

La sortie de tous les scripts shell, utilisés pour la personnalisation.

variables.log

Une liste de toutes les variables shell qui sont disponibles au cours d'une installation.

error.log

Résumé des erreurs possibles dans tous les fichiers journaux.

disk_var.sh

Une liste des variables contenant des informations sur les périphériques et les partitions à partir desquelles la partition racine et une liste de périphériques de swap. Ces informations sont utilisées par certains scripts de personnalisation (par exemple `GRUB_PC/10-setup`).

Si le processus d'installation se termine, le hook `savelog.LAST.sh` recherche tous les fichiers journaux pour les erreurs courantes et les écrit dans le fichier `error.log`. Donc, vous devriez d'abord regarder dans ce fichier pour les erreurs. Le fichier `status.log` vous donne également le code de sortie de la dernière commande exécutée dans un script. Pour être sûr, vous devriez rechercher plus de détails dans tous les fichiers journaux.

9.10 Comment utiliser HTTP pour le démarrage PXE

```
cp /usr/lib/PXELINUX/lpxelinux.0 /srv/tftp/fai/pxelinux.0
```

Activer l'accès HTTP au répertoire tftp:

```
cd /var/www/html
ln -s /srv/tftp/fai
```

Ajoutez `-U URL` à l'appel `fai-chboot`. Par exemple:

```
fai-chboot -U http://faiserver/fai -IFv .....
```

10 Dépannage

10.1 Erreurs d'amorçage

Le message d'erreur suivant indique que votre client d'installation n'obtient pas de réponse d'un serveur DHCP. Vérifiez vos câbles ou démarrez le démon `dhcpcd(8)` avec le debug flag activé.

```
PXE-E51: No DHCP or BOOTP offers received Network boot aborted
```

Si vous ne voyez pas le message suivant, le noyau d'installation n'a pas pu détecter votre carte réseau, par exemple en raison d'un pilote manquant:

```
Starting dhcp for interface eth0
dhcp: PREINIT eth0 up
dhcp: BOND setting eth
```

Vérifiez l'initrd dans le nfsroot (`lsinird`) si le pilote du noyau de votre carte réseau est inclus et vérifiez si vous souhaitez ajouter le paquet *firmware-linux-nonfree* dans `/etc/fai/NFSROOT` et reconstruisez l'initrd en appelant `fai-make-nfsroot -k`. Vous pouvez également ajouter un pilote à `/srv/fai/nfsroot/etc/dracut.conf` dans la ligne `add_drivers+=`.

C'est le message d'erreur que vous verrez, lorsque votre carte réseau fonctionne, mais le serveur d'installation n'exporte pas le répertoire nfsroot vers les clients d'installation. Cela est souvent dû aux permissions NFS manquantes du côté serveur.

```
Starting dhcp for interface eth0
dhcp: PREINIT eth0 up
dhcp: BOND setting eth
mount.nfs: access denied by server while mounting 192.168.33.250:/srv/fai/nfsroot
.
.
dracut Warning: Could not boot
.
Dropping to debug shell
dracut:/#
```

Maintenant, vous êtes à l'intérieur du shell d'urgence de l'initrd qui a été créé par *dracut(8)*. Vous obtiendrez une invite du shell et pourrez consulter les fichiers journaux. Pour plus d'informations sur le débogage du processus de démarrage précoce à l'aide de *dracut*, consultez `dracut.cmdline(7)`

Utilisez la commande suivante sur le serveur d'installation pour voir quels répertoires sont exportés à partir du serveur d'installation (nommé `faiserver`):

```
$ showmount -e faiserver
```