

FAI Guide (Fully Automatic Installation)

Thomas Lange

Contents

1	Résumé	1
2	Introduction	2
2.1	Disponibilité	2
2.2	Motivation	2
2.3	Comment fonctionne FAI	3
2.4	Caractéristiques	3
2.5	La durée d'installation	4
3	Quickstart - Pour l'utilisateur impatient	4
3.1	Ma première installation	4
3.2	Ma première installation à l'aide d'un serveur	4
4	Vue d'ensemble et Concepts	5
4.1	Terminologie	6
4.2	Le concept de classe	7
5	Configurer votre faiserver	8
5.1	Installer les paquets FAI	8
5.2	Créer le nfsroot	9
5.3	Création de l'espace de configuration	9
5.4	Configurer les démons réseau	10
5.4.1	Configuration du démon DHCP	10
5.4.2	Ajout d'une entrée d'hôte au DHCP	11
5.4.3	TFTP	11
5.4.4	NFS	11
5.5	Création de la configuration PXELINUX	11
5.6	Serveur personnalisé	12
6	Planifiez votre installation	12
6.1	L'espace de configuration et ses sous-répertoires	12
6.2	Définition des classes	13
6.3	Définition des Variables	14
6.4	Configuration du disque dur	15
6.5	Extraction du fichier de base	15
6.6	Pré-ensemencement Debconf	15
6.7	Accès au dépôt de paquets	15
6.8	Configuration des paquets logiciels	15

6.9	Scripts de personnalisation	17
6.9.1	Scripts shell	18
6.9.2	Scripts cfengine	18
6.10	Les hooks, les « crochets »	18
6.11	Options (flags) de FAI	19
7	FAI installe votre planification	20
7.1	La première phase d'une installation	20
7.2	Messages de démarrage	20
7.3	Redémarrage de l'ordinateur dans le nouveau système	22
7.4	Démarrage de FAI (tâche confdir)	22
7.5	Définition de classes et de variables (tâches defclass et defvar)	22
7.6	Partitionnement des disques locaux, création des systèmes de fichiers (tâche partition)	22
7.7	Pré-ensemencement Debconf (tâche debconf)	22
7.8	Installation de paquets logiciels (tâche instsoft)	23
7.9	Personnalisation spécifique au site (tâche configure)	23
7.10	Enregistrement des fichiers journaux (tâche savelog)	23
7.11	Redémarrage du nouveau système installé	23
8	Sujets avancés pour FAI	24
8.1	Vérification des paramètres reçus des serveurs DHCP	24
8.2	Surveillance de plusieurs clients d'installation	24
8.3	Collecte d'adresses Ethernet pour plusieurs hôtes	24
8.3.1	Débogage du trafic réseau	24
8.4	Détails du démarrage PXE	25
8.5	Personnaliser la configuration de son serveur d'installation	25
8.6	Création d'un CD ou d'une clé USB FAI	26
8.7	Création d'images de disque VM à l'aide de FAI	27
8.8	Création d'une image vive amorçable	27
8.9	Construire des images disque d'architecture différente (cross-architecture)	27
8.10	Système de sauvetage FAI	28
8.11	FAI sans NFS	28
8.12	Installation d'autres distributions à l'aide d'un nfsroot Debian	29
8.13	Création d'environnements chroot et de virtualisation	29
8.14	Utilisation de FAI pour les mises à jour	29
8.15	Comment installer un système d'exploitation 32 bits à partir d'un système d'exploitation 64 bits	30
8.16	Arrêt d'urgence de l'installation quand une erreur se produit	31

9	Divers conseils et détails	31
9.1	La liste des tâches	31
9.2	Tests automatisés	33
9.3	Découvrir automatiquement	33
9.4	Modification du périphérique d'amorçage	33
9.5	Comment créer un miroir Debian local	34
9.6	Petits conseils	34
9.7	flag_reboot (FAI_FLAGS)	35
9.8	Fichiers journaux	36
9.9	Comment utiliser HTTP pour le démarrage PXE	36
10	Dépannage	37
10.1	Erreurs d'amorçage	37

Une version plus ancienne de ce guide en français est disponible à <https://fai-project.org/fai-guide-fr>.

1 Résumé

FAI est un système non-interactif pour installer, configurer et gérer des systèmes Linux et les configurations de logiciels sur des ordinateurs comme sur des machines virtuelles et des environnements chroot, depuis des petits réseaux jusqu'à de grandes infrastructures et de grands clusters.

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

(c) 2000-2025 Thomas Lange

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 License Générale Publique GNU.

Une copie de la Licence Générale Publique GNU 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

<https://fai-project.org>

Wiki FAI

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

Téléchargement

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

Entrée pour *sources.list*

deb <https://fai-project.org/download> bookworm koeln

Pages de manuel

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

Listes de diffusion

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

Retour d'information

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

Boguess

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

Changements visibles par l'utilisateur

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

Arbre source via git

git clone [git://github.com/faiproject/fai.git](https://github.com/faiproject/fai.git)

Voir l'arbre source avec https

<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 d'un seul coup?

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 étaient faites automatiquement. Un processus d'installation avec interaction manuelle ne peut pas passer à l'échelle. Mais les grappes (clusters) ont l'habitude de croître au fil des ans. Pensez à long terme plutôt que de planifier seulement 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 pouvais épargner beaucoup de temps non seulement pour chaque nouvel ordinateur de SUN, mais aussi pour la réinstallation des postes de travail existants. Par exemple,

¹Solaris 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>

j'ai dû construire un LAN temporaire avec quatre stations de travail SUN pour une conférence, qui a duré seulement quelques 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 elle a été terminée, j'ai simplement réintégré 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 qu'un disque dur endommagé ait 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 paquets logiciels 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 sur le serveur d'installation. Les fichiers de configuration sont répartis 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. Ainsi, 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

- On peut faire une installation entièrement automatisée.
- Installation très rapide sans surveillance.
- Système flexible grâce à un concept de classe simple.
- Mise à jour des systèmes en cours d'exécution sans réinstallation.
- Images « Cloud » pour un environnement de virtualisation.
- Les hôtes peuvent démarrer à partir d'une carte réseau, d'un CD, d'une clé USB.
- Création simple d'une image ISO vive ou pour l'installation.
- Prise en charge de PXE avec la méthode de démarrage DHCP.
- Support des systèmes de fichiers ReiserFS, ext3/ext4, btrfs et XFS.
- Support logiciel RAID et LVM. Support LUKS.
- Détection automatique du matériel.
- On peut déployer Debian, Ubuntu, Rocky Linux, CentOS, AlmaLinux, SuSE
- Possibilité de connexion à distance via ssh lors du processus d'installation.
- 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.
- Prise en charge des scripts Shell, Perl, Python, Ruby, expect et CFEngine 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 de clients sans disque.
- Prise en charge trans-architecture, par exemple pour des appareils embarqués (embedded devices).
- Ajoutez facilement vos propres fonctions via des hooks ou modifiez le comportement par défaut.
- Prise en charge du clonage de machines utilisant des images de disque.

2.5 La durée d'installation

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

CPU	RAM	Disque	Logiciels installés	durée
i7-3770T 2.50GHz	8GB	SSD	6 GB logiciel	8.5 min
Core-i7 3.2GHz	6GB	SATA disk	4.3GB logiciel	7 min
Core-i7 3.2GHz	6GB	SATA disk	471 MB logiciel	77sec
Intel Core2 Duo	2GB	SATA disk	3 GB logiciel	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 <https://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 à partir du CD est identique à celle exécutée dans un environnement réseau à l'aide du serveur d'installation FAI, et peut être aussi effectuée à l'aide d'une clé USB sur un ordinateur physique.

On peut aussi créer soi-même une image FAI (installation entièrement automatique) à la page web <https://fai-project.org/FAIme> sans avoir besoin d'installer FAI sur son ordinateur.

3.2 Ma première installation à l'aide d'un serveur

Nous allons créer un réseau privé et lancer deux machines virtuelles. L'une deviendra votre propre serveur FAI, l'autre sera un client d'installation.

Si vous avez l'intention d'utiliser VMware ou VirtualBox, assurez-vous que votre client utilise une connexion réseau pontée (bridged network). On trouve une description détaillée dans le wiki FAI ³. on ne peut pas utiliser des interfaces réseau pontées via le réseau sans fil, car la plupart des cartes réseau WiFi ne prennent pas en charge cette fonctionnalité.

Quand on utilise Qemu/KVM et le script `fai-kvm` on peut créer le réseau à l'aide de la commande `fai-mk-network`. Il faut commencer par installer quelques paquets logiciels

```
# apt-get install qemu-system-gui qemu-kvm qemu-utils
```

La commande ci-dessous met en place un réseau privé avec un pont logiciel et plusieurs TAPs réseau ⁴ appartenant à l'utilisateur `<nom_d_utilisateur>`.

³https://wiki.fai-project.org/index.php/VirtualBox_for_your_first_FAI_installation

⁴https://fr.wikipedia.org/wiki/TAP_r%C3%A9seau


```
fai-mk-network <nom_d_utilisateur>
```

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. Faites attention. Par défaut, `fai-kvm` créera les images de disque pour les machines virtuelles dans `+ /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+`.

Démarrez le premier hôte virtuel, qui deviendra le serveur FAI ⁵ :

```
fai-kvm -Vn -s20 -u 1 cd faicd64-small.iso
```

Dans le menu grub; sélectionnez `faiserver, using internal DHCP and a 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 un 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` n'y 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 de TAPs réseau 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 fonctionnel 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 impliquant peu d'efforts. 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 des scripts CFEngine, shell (bash et zsh), Perl, Python, Ruby et expect.

Les personnes intéressées par 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 ferme 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 à ce sujet.

⁵Cette installation utilisera environ 2GB d'espace dans `/tmp`.

4.1 Terminologie

Commençons par décrire quelques termes utilisés dans ce manuel.

Le serveur d'installation

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*. C'est l'hôte où le paquet *faiserver* est installé.

Le client d'installation

Un hôte qui sera installé à l'aide de FAI et une configuration fournie par le serveur d'installation. Aussi appelé client pour faire court. 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.

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 là. Il est également appelé *config space* pour faire court. Il comprend des informations sur :

- Organisation du disque dur dans un format similaire à fstab
- Systèmes de fichiers locaux, leurs types, points de montage et options de montage
- Paquets 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 entre autres. En invoquant `fai-mk-configspace` on copie ces exemples vers `/srv/fai/config`. Il est recommandé d'étudier ces fichiers de configuration et ces scripts pour mieux comprendre comment FAI fonctionne.

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 1.1Go d'espace disque libre.

TFTP

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

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, déterminer les paquets logiciels qui seront installés.

profil

Un profil FAI est juste une liste de classes FAI assignées 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`, et un autre incluant la classe `NGINX`, pour ensuite installer la solution de serveur web correspondante.

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          # trouver l'espace de configuration
setup            # quelques initialisations, lancement de sshd à la demande ←
defclass         # définition des classes FAI
defvar           # définition de variables
action           # évaluer FAI_ACTION
```

```

install          # Démarrer l'installation
partition        # partitionner les disques durs, créer les systèmes de fichiers ←
mountdisks       # monter les systèmes de fichiers
extrbase         # extraire le fichier base.tar.xz
debconf          # pré-ensemencer le deconf de Debian
repository       # préparer l'accès aux dépôts de paquets
updatebase       # mettre en place les outils de gestion de paquets ←
                 et mettre les paquets à jour
instsoft         # installer les paquets logiciels
configure        # lancer les scripts de personnalisation
finish           # faire un peu de ménage, afficher les statistiques ←
                 d'installation
tests            # appeler des tests s'il y en a
chboot           # lancer fai-chboot sur le serveur d'installation
savelog          # enregistrer les fichiers de journalisation ←
                 localement et à distance
faiend           # redémarrage de l'hôte, éjection du CD si ←
                 nécessaire

```

Voilà des tâches qui sont exécutées seulement lorsqu'on fait une action différente

```

dirinstall       # installer un environnement chroot
softupdate       # ne faire que la configuration du système
sysinfo          # imprimer une information système détaillée
inventory        # imprimer un inventaire court du matériel

```

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 dans la liste des classes définies et utilise tous les fichiers de configuration correspondant à un nom de classe ⁶. La boucle suivante implémente cette fonction en pseudo-code shell :

```

for class in $all_classes; do
  if [ -r $config_dir/$class ]; then      # s'il existe un fichier nommé $class
    your_command $config_dir/$class      # invoquer une commande avec ce nom de fichier
    # sortir si on n'a besoin que du premier fichier qui convient
  fi
done

```

La beauté de ceci c'est qu'on peut ajouter une nouvelle alternative de configuration et qu'elle sera automatiquement utilisée par FAI sans modifier le code, si le fichier de configuration utilise un nom de classe.

⁶on peut aussi seulement utiliser le fichier de configuration qui a la priorité la plus élevée, comme l'ordre des classes définit une priorité de la plus basse à la plus élevée dans la liste des classes.

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.

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

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, elles peuvent définir quels paquets logiciels seront installés, ou quelles étapes de personnalisation seront exécutées.

Souvent, une configuration client est juste créée en modifiant ou en ajoutant 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 celui de votre client d'installation. 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 les paquets de FAI
 - b. Créer le `nfsroot`
 - c. Copier les exemples dans l'espace de configuration
 - d. Configurer les démons réseau
 - e. Créer les configurations PXELINUX
2. Démarrage et installation des clients

5.1 Installer les paquets FAI

- Installer la clé du dépôt de paquets du projet FAI.
- Ajouter l'URL du dépôt de paquets du projet FAI.
- Installer le paquet *fai-quickstart* sur le serveur d'installation.

Utilisez ces commandes pour l'installation :

```
# wget -O /etc/apt/trusted.gpg.d/fai-project.gpg https://fai-project.org/download/2 ↵  
BF8D9FE074BCDE4.gpg  
# echo "deb http://fai-project.org/download bookworm koeln" > /etc/apt/sources.list.d/fai. ↵  
list
```

ou

```
# apt install extrepo ; extrepo enable fai
```

et ensuite :

```
# apt-get update  
# aptitude install fai-quickstart
```

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

5.2 Créer le nfsroot

- Activez également le dépôt de paquets 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 miroir de paquets, qui va tenter de trouver un dépôt de paquets rapide pour vous.⁷ Ensuite, on peut lancer `fai-setup` (8)⁸ Et vérifiez si tout s'est bien passé. Le fichier de journal est écrit dans `/var/log/fai/fai-setup.log`.

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

- Voilà 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 <https://fai-project.org/logs/fai-setup.log>.

```
FAI packages and related packages inside the nfsroot:
dracut                059-4
dracut-live           059-4
dracut-network        059-4
dracut-squash         059-4
fai-client            6.2
fai-nfsroot           6.2
fai-setup-storage     6.2
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 192.168.33.250/24(async,ro,no_subtree_check)
Adding line to /etc/exports: /srv/fai/nfsroot 192.168.33.250/24(async,ro,no_subtree_check, ←
no_root_squash)
Reloading nfs-kernel-server configuration (via systemctl): nfs-kernel-server.service.

Your initial config space is now located in /srv/fai/config
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
```

- `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 configuration⁹.

```
$ fai-mk-configspace
```

⁷Si vous voulez un miroir plus rapide, ajustez l'URL dans */etc/fai/apt/sources.list* et `FAI_DEBOOTSTRAP` dans */etc/fai/nfsroot.conf* avant de lancer `fai-setup`.

⁸Cela invoque `fai-make-nfsroot` (8) en interne.

⁹Il n'est pas nécessaires que ces fichiers appartiennent au compte root.

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.

gnomehost

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

ubuntuhost::\n" Un bureau Ubuntu sera installé et la démo du compte est créée.

autres noms d'hôtes

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 démon DHCP et d'un démon TFTP fonctionnels. Le paquet *fai-quickstart* a déjà installé les paquets logiciels pour ces démons. En outre, le paquet du serveur NFS pour l'exportation du nfsroot et de l'espace de configuration a été installé.

5.4.1 Configuration du démon DHCP

Idéalement, 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. Il est possible d'éviter cela optionnellement 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 d'assigner `filename` (DHCP option 67) à la valeur `fai/pxelinux.0` et d'assigner `next-server` (DHCP option 66, also aussi nommée TFTP server name) et `server-name` à la valeur du nom de votre serveur d'install. Toutes les autres informations sont uniquement des données liées au réseau, qui sont utilisées dans presque toutes les configurations DHCP. Ajustez ces paramètres de réseau à vos besoins locaux.

```
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.

```
# invoke-rc.d isc-dhcp-server restart
```

Si vous disposez de plusieurs interfaces réseau, vous pouvez définir l'interface que le serveur écouterait 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 l'associez 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* ¹⁰.

```
192.168.33.100    demohost
```

L'association de l'adresse MAC à l'adresse IP est effectuée 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 l'adresse MAC que vous avez trouvée.

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

Après l'invocation de cette commande, voici à quoi l'entrée d'hôte dans *dhcpd.conf* ressemblera :

```
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 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 utilisons la commande `fai-chboot (8)` pour créer une configuration pxelinux pour chaque client d'installation. Cela comprend des informations sur le noyau, le fichier `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`.

```
$ fai-chboot -IFv -u nfs://faiserver/srv/fai/config demohost

Booting kernel vmlinuz-4.19.0-5-amd64
append initrd=initrd.img-4.19.0-5-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.

¹⁰On peut aussi ajouter cela dans son serveur de noms de domaine (DNS)

5.6 Serveur personnalisé

Le faiserver 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 qu'il y manque 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 lancées par les utilisateurs ?
- Les utilisateurs ont-ils besoin d'un système de mise en file d'attente ?
- Quels logiciels faut-il installer ?
- Quels démons devraient être lancés, et à quoi devrait ressembler leur configuration ?
- Quels systèmes de fichiers distants doivent être montés ?
- Comment effectuer les sauvegardes ?

Vous devez également penser à des comptes d'utilisateur, des imprimantes, un système de courrier, des travaux de cron, des cartes graphiques, au double amorçage, au NIS, au NTP, au fuseau horaire, à la disposition de clavier, à l'exportation et au montage des annuaires via NFS et à 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 des informations qui disent comment exactement configurer un client. L'espace central de configuration pour tous les clients d'installation est situé sur le serveur d'installation dans `/srv/fai/config` et ses sous-répertoires. Il sera monté par les clients sous `/var/lib/fai/config`. La commande principale d'installation `fai (8)` utilise tous ces sous-répertoires dans l'ordre listé 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 des paquets. 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 celle du *nfsroot*, vous pouvez mettre un fichier tar dans le sous-répertoire *basefiles/* et le nommer selon 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/HTTPS 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=https://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-selection*.

package_config/

Les fichiers avec des noms de classe contiennent des listes de paquets à installer ou à désinstaller par *install_packages* (8). Les fichiers nommés *<CLASS>.gpg* sont ajoutés à la liste des clés utilisées par *apt* pour les dépôts de paquets approuvés.

pkgs/

Ce répertoire peut avoir des sous-répertoires nommés selon des classes. Vous pouvez mettre des fichiers *.deb* ou *.rpm* dans ces sous-répertoires. FAI installera ces paquets sans qu'il soit besoin de créer les métadonnées d'un dépôt de paquets.

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 un sous-répertoire 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 lancés pendant le processus d'installation. Ils peuvent 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 tâche *updatebase* et seulement si l'installation du client fait partie de la classe DEBIAN.

6.2 Définition des classes

Il existe différentes possibilités pour définir des classes :

1. Certaines classes par défaut sont définies pour chaque hôte : *DEFAULT*, *LAST* et son nom d'hôte.
2. Les classes peuvent être répertoriées dans un fichier (par défaut dans *class/<hostname>*).
3. Les classes peuvent être dynamiquement définies par des scripts.

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

Tous les noms de classes, sauf le nom d'hôte, sont écrits en majuscules. Ils ne doivent pas contenir de trait d'union, de dièse, de point-virgule ni de 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 à l'expression `^[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 défini comme une classe. Pour plus d'informations sur Les définitions de classe, lire les pages de manuel de `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 DEMO" ;;
    xfcehost)
        echo "FAIBASE GRUB DEMO XORG XFCE LVM";;
    faiserver)
        echo "FAIBASE DEBIAN DEMO FAISERVER" ;;
    ubuntuhost)
        echo "FAIBASE DEBIAN DEMO UBUNTU JAMMY JAMMY64 XORG";;
    *)
        echo "FAIBASE DEBIAN DEMO" ;;
esac
```

Les noms d'hôtes doivent rarement être utilisés pour les fichiers de configuration dans l'espace de configuration. Au lieu de ça, 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 à un nom d'hôte, mais peuvent être partagées entre différents hôtes.

L'ordre des classes est important car celui-ci définit la priorité des classes de faible à élevée.

6.3 Définition des Variables

La tâche *defvar* définit les variables pour le client d'installation. Les variables sont définies par les scripts dans `class/*.var`. Toutes les variables globales peuvent être définies dans `DEFAULT.var`. Pour un groupe d'hôtes utiliser un fichier de classe. Pour un hôte seul, 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 aussi être utiles pour votre installation :

FAI_ACTION

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

FAI_ALLOW_UNSIGNED

Si cette variable vaut 1, FAI permet l'installation de paquets à partir de dépôts non signés.

CONSOLEFONT

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

KEYMAP

Définit les fichiers de disposition du clavier dans `/usr/share/keymaps` et `$FAI/files`. Vous n'avez pas besoin de spécifier le chemin complet, car 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 mot de passe donné. Par exemple, pour générer le hachage MD5 un mot de passe, utilisez :

```
$ echo "votre_mot_secret" | mkpasswd -m yescrypt -s
```

UTC

Règle l'horloge matérielle à UTC si *UTC=yes*. Sinon, règle l'horloge à l'heure locale. Voir `clock(8)` pour plus d'informations.

TIMEZONE

C'est le fichier sous `/usr/share/zoneinfo/` qui 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 (écrits dans `/etc/modules`).

6.4 Configuration du disque dur

L'outil `setup-storage(8)` lit un fichier dans `$FAI/disk_config` pour la configuration du disque. Ce fichier décrit comment tous les disques Locaux seront partitionnés, quels types de systèmes de fichiers doivent être créés (comme ext3/4, xfs, btrfs), et où ils seront montés. Vous pouvez aussi créer des configurations RAID logiciel et LVM en utilisant ce fichier de configuration. Il est aussi possible de préserver le partitionnement du disque ou de préserver les données sur certaines partitions.

Pendant le processus d'installation, tous les systèmes de fichiers locaux sont montés sous le répertoire `/target`. Par exemple, si vous spécifiez le point de montage `/home` dans un fichier de configuration de disque, ce sera le répertoire `/target/home` pendant le processus d'installation et ça deviendra `/home` pour le nouveau système installé.

6.5 Extraction du fichier de base

Un fichier de base n'est nécessaire que si on installe une distribution qui est différente de celle dans le `nfsroot`.

6.6 Pré-ensemencement Debconf

On peut utiliser le format décrit dans `debconf-set-selections(1)`.

6.7 Accès au dépôt de paquets

FAI supporte http, https, et NFS pour l'accès au miroir de paquets. Réglez la variable `$FAI_DEBMIRROR` pour l'utilisation de NFS.

6.8 Configuration des paquets logiciels

Avant l'installation de paquets, FAI va ajouter le contenu de tous les fichiers nommés `package_config/class.gpg` à la liste des clés apt. Si votre dépôt local est signé par votre keyid AB12CD34 vous pouvez facilement ajouter cette clé, ainsi FAI l'utilisera pendant l'installation. Utilisez cette commande pour créer le fichier `CLASS.gpg` :

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

Le script `install_packages(8)` installe les logiciels sélectionnés. Il lira tous les fichiers de configuration Dans `$FAI/package_config` dont le nom correspond à une classe définie. La syntaxe est très simple.

```
# an example package class

PACKAGES taskinst
german

PACKAGES install-norec
```

```
adduser nmap
less zstd

PACKAGES remove
gpm xdm

PACKAGES install GRUB_PC
grub-pc
```

Les commentaires commencent par une dièse (#) et se terminent à la fin de la ligne. Chaque commande de paquet commence par le mot *PACKAGES* suivi par un nom de commande, qui correspond à gestionnaire de paquets distinct comme apt-get, aptitude ou dnf par exemple. la commande définit quelle commande sera utilisé pour installer les paquets nommés après cette commande. La liste de toutes les commandes disponibles peut être listée en utilisant *install_packages -H*. Les gestionnaires de paquets pris en charges sont *aptitude*, *apt-get*, *smart*, *yast*, *dnf*, *rpm*, *zypper*

hold

Mettre un paquet en attente. Ce paquet ne sera pas pris en charge par dpkg, pas exemple il ne sera pas mis à niveau.

install

Pour installer tous les paquets (en utilisant apt-get) qui sont précisés 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 de frappe. Tout paquet qui n'existe pas, sera retiré de la liste des paquets à installer. Soyez donc attentif à 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 pécisés dans les lignes suivantes. Si on ajoute un + au nom du paquet, le paquet doit être installé.

taskinst

Installer 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

Installer tous les paquets avec la commande *aptitude*. Ce sera la valeur par défaut à l'avenir et pourra remplacer apt-get et taskinst. Aptitude peut aussi installer les tâches de paquets.

aptitude-r

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

unpack

Télécharger les paquets et les décompresser seulement. Ne pas configurer les paquets.

dselect-upgrade

Définir des sélections des paquets en utilisant les lignes suivantes puis installer ou supprimer les paquets précisés. Ces lignes sont le résultat de la commande *dpkg --get-selections*. Il est recommandé de ne pas utiliser ce format, puisque cela spécifie aussi tous les paquets qui sont installés seulement en raison d'une dépendance ou d'une 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 paquets 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)*.

On peut spécifier des paramètres supplémentaires pour le gestionnaire de paquets en ajoutant des paires *cle=valeur* après *PACKAGES <commande>*. les paires clé, valeur actuellement supportées sont *release=<nom>* qui ajoute *-t <nom>* durant l'installation des paquets.

Exemple:

```
PACKAGES install-norec release=testing
nvidia-smi
```

Cela installerait le paquet `nvidia-smi` de la distribution `testing`, en incluant les dépendances. N'oubliez pas d'ajouter une entrée dans *sources.list*. Vous pouvez aussi vouloir ajuster l'épinglage `apt` (`apt pinning`) (voir *apt_references(5)*)

Une ligne qui contient la commande *PRELOADRM*, télécharge un fichier à l'aide de `wget` (1) dans un répertoire avant d'installer les paquets. À l'aide d'une URL de type *file:*, ce fichier est copié de `$FAI_ROOT` vers le répertoire de téléchargement. Par exemple, le paquet `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*.

On peut ajouter une expression booléenne arbitraire qui utilise des classes FAI pour définir quand la liste des paquets devrait être installée. Ici, les paquets seront installés seulement si la classe `XORG` est définie mais la classe `MINT` n'est pas définie.

Exemple:

```
PACKAGES install UBUNTU && XORG && ! MINT
ubuntu-standard
ubuntu-desktop
```

L'ancienne façon d'ajouter des expressions logiques dans les lignes `PACKAGES` est encore valable : Il est possible d'ajouter une liste de noms de classes après la commande pour `apt-get`. Ainsi une telle commande *PACKAGES* sera exécutée seulement au moins une des classes correspondantes est définie (OU logique). On peut donc combiner plusieurs petits fichiers dans le fichier `DEFAULT`.

Si vous souhaitez supprimer un nom de paquet d'une certaine classe faisait partie de cette classe précédemment, vous ne devez pas supprimer le nom du paquet classe, mais plutôt de lui ajouter un tiret (-). Cela garantira que le paquet soit enlevé pendant une mise à jour sur des hôtes qui étaient installés 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.

Le concept de priorité des classes permet à une classe de priorité plus élevée (qui arrive plus tard dans la séquence des classes) d'écraser les sélections des paquets de priorité plus basse. Pour que cela fonctionne correctement, les classes de priorité plus élevée doivent utiliser les mêmes commandes *PACKAGES* (par exemple *PACKAGES install-norec* au lieu de seulement *PACKAGES install*) que celles qu'utilisent la classe qu'on essaie d'écraser. Cela est utile pour supprimer l'installation d'un paquet, par exemple, pour éviter d'installer le paquet *linuxlogo* installé par la classe `FAIBASE` :

```
# exemple montrant comment écraser :
#
# Dans FAIBASE il y a :
#   PACKAGES install-norec
#   linuxlogo
#
# On veut _ne pas_ installer linuxlogo, et il se trouve dans
# une section install-norec, donc il faut aussi utiliser install-norec.

PACKAGES install-norec
linuxlogo-
```

6.9 Scripts de personnalisation

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

Ces scripts écrivent leur sortie standard dans `scripts.log`. Le fichier `status.log` contient les noms de tous les scripts exécutés et leur statut de terminaison.

6.9.1 Scripts shell

La plupart des scripts sont des scripts Bourne shell. Les scripts shell sont utiles si la tâche de configuration n'a besoin d'appeler que quelques commandes shell ou créer un fichier à partir de zéro. Afin d'éviter d'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` (ET, 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 *LocateLineMatching*, *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.

On peut trouver plus d'informations dans la page de manuel `cfengine(8)` ou sur la page d'accueil de cfengine <https://www.cfengine.com>.

6.10 Les hooks, les « crochets »

Les crochets (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 crochets 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. Il vous suffit de 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 est nommé selon la tâche : `taskname.CLASSNAME` (par exemple, `repository.CENTOS` ou `savelog.LAST.sh`). Le nom de tâche spécifie quelle tâche précède 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 qu'on peut utiliser.

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 peuvent utiliser uniquement les classes suivantes: *DEFAULT \$HOSTNAME LAST*. Si un hook pour la classe *DEFAULT* doit être appelé uniquement dans le cas où aucun hook pour la classe `$HOSTNAME` n'est disponible, insérez ces lignes dans le hook par défaut :

```
hookexample.DEFAULT:
```

```
#!/bin/sh
```

```
# sauter le hook DEFAULT s'il existe un hook pour $HOSTNAME
scriptname=$(basename $0 .DEFAULT)
[-f $FAI/hooks/$scriptname.$HOSTNAME ] && exit
# suivent alors les actions pour la classe DEFAULT
.
.
.
```

Quelques exemples de ce à quoi les hooks pourraient être utiles :

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

6.11 Options (flags) de FAI

La variable *\$FAI_FLAGS* contient une liste d'options séparées par des espaces. Les options sont normalement définies dans le fichier *pxelinux.cfg* qui est censément créé par *fai-chboot(1)*. Les options suivantes sont connues :

verbose

Rend l'installation verbeuse. Cela doit toujours être la première option (le premier flag), de sorte que les définitions suivantes des options soient affichées verbeusement.

debug

Afficher des informations 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 affichées. Cette option n'est utile que pour les développeurs FAI.

ssh

Démarrer 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 le mot de passe *root* que 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 cette option est utile pour le débogage. Si vous voulez une installation qui ne soit pas interruptible, ne définissez pas cette option.

menu

Cela active un menu utilisateur pour sélectionner un profil. Tous les fichiers *class/*.profile* sont lus et un menu basé sur curses sera créé.

screen

Lancer FAI dans une session *screen(1)*. La session est nommée FAI. Si on se connecte via *ssh* à distance on peut se rattacher à la session en utilisant :

```
$ screen -x
```

tmux

Lancer FAI dans une session `tmux(1)`. La session est nommée FAI. Si on se connecte via ssh à distance on peut se rattacher à la session en utilisant :

```
$ tmux attach
```

reboot

Redémarrer le client d'installation une fois l'installation terminée sans taper Entrée sur la console. Si cette option n'est pas définie, et que `error.log` contient quelque chose, le client d'installation s'arrêtera et attendra que vous appuyez sur Entrée. Si aucune erreur ne s'est produite, le client redémarrera toujours automatiquement.

halt

Arrêter 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 cette option ne soit définie. Souvent, cette option est définie dans un fichier `class/*.var` en utilisant le paramètre `flag_initial=1`.

7 FAI installe votre planification

7.1 La première phase d'une installation

Après le démarrage du noyau, on monte le système de fichiers racine via NFS à partir du serveur d'installation et on démarre le script `/usr/sbin/fai` ¹¹. 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) depuis le serveur d'installation vers le chemin défini dans `$FAI` ¹²

7.2 Messages de démarrage

Lorsqu'on démarre le client d'installation à partir de la carte réseau avec PXE, on voit 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
```

¹¹ Comme le système de fichiers racine des clients est monté via NFS, `fai` se situe dans `/srv/fai/nfsroot/usr/sbin` sur le serveur d'installation.

¹² `$FAI` est une variable interne utilisée par les scripts FAI. Le chemin par défaut est `/var/lib/fai/config`.


```

SYSUID=
TFTP prefix: fai/
Trying to load pxelinux.cfg/C0A82164

Loading vmlinuz-6.1.0-17-amd64.....
Loading initrd.img-6.1.0-17-amd64.....ready.

```

À ce stade, le client d'installation a réussi à recevoir la configuration réseau via DHCP, le noyau et le fichier initrd via TFTP. Il démarre maintenant le noyau Linux et met en place initrd. Si tout se passe bien, l'initrd monte le nfsroot¹³ et les scripts FAI sont lancés. La première chose qu'on voit est le message de copyright en rouge de FAI.

```

-----
          Fully Automatic Installation  -  FAI
          6.2                (c) 1999-2024
          Thomas Lange      <lange@cs.uni-koeln.de>
-----

Calling task_confdir
Kernel currently running: Linux 6.1.0-17-amd64 x86_64 GNU/Linux
Kernel parameters: BOOT_IMAGE=vmlinuz-6.1.0-17-amd64 initrd=initrd.img-6.1.0-17-amd64 \
ip=dhcp rw root=192.168.33.250:/srv/fai/nfsroot rootovl FAI_FLAGS=verbose,sshd,createvt \
FAI_ACTION=install FAI_CONFIG_SRC=nfs://faiserver/srv/fai/config
Reading /tmp/fai/boot.log
FAI_FLAGS: verbose sshd createvt
Monitoring to server faiserver enabled.
FAI_CONFIG_SRC is set to nfs://faiserver/srv/fai/config
Configuration space faiserver:/srv/fai/config mounted to /var/lib/fai/config
Source hook: setup.DEFAULT.sh
setup.DEFAULT.sh      OK.
Calling task_setup
FAI_FLAGS: verbose sshd createvt
Press ctrl-c to interrupt FAI and to get a shell
Starting FAI execution - 20240117_194012
Calling task_defclass
fai-class: Defining classes.
Executing /var/lib/fai/config/class/01-classes.
01-classes            OK.
Executing /var/lib/fai/config/class/10-base-classes.
10-base-classes       OK.
Executing /var/lib/fai/config/class/20-hwdetect.sh.
ens3                  UP                52:54:00:11:23:01 <BROADCAST,MULTICAST,UP,LOWER_UP>
ens3                  UP                192.168.33.101/24 fe80::5054:ff:fe11:2301/64
New disklist: vda
20-hwdetect.sh        OK.
Executing /var/lib/fai/config/class/40-parse-profiles.sh.
40-parse-profiles.sh  OK.
Executing /var/lib/fai/config/class/41-warning.sh.
41-warning.sh         OK.
Executing /var/lib/fai/config/class/50-host-classes.
50-host-classes       OK.
Executing /var/lib/fai/config/class/60-misc.
60-misc               OK.
Executing /var/lib/fai/config/class/85-efi-classes.
85-efi-classes        OK.
List of all classes:  DEFAULT LINUX AMD64 DHCPD FAIBASE DEBIAN DEMO GRUB_PC demohost2 LAST

```

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

¹³/srv/fai/nfsroot depuis le serveur d'installation via NFS

La première tâche est appelée *confdir*, qui est chargée de accès à l'espace de configuration. Ici, on utilise un montage NFS depuis le serveur d'installation comme on peut 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 émet trois bips. Donc, faites attention quand vous entendez trois bips et que vous ne voulez pas effectuer une installation, ni 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* de préférence à la commande de redémarrage normal. Utilisez aussi *faireboot* si vous êtes connecté à distance. Si l'installation n'est pas terminée, utilisez *faireboot -s*, afin que les fichiers journaux soient aussi copiés sur le serveur d'installation.

Si l'installation s'est bien terminée, l'ordinateur doit démarrer un petit système Debian. Vous pouvez vous connecter en tant qu'utilisateur *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 `$FAI_CONFIG_SRC` ¹⁴ est utilisée pour accéder à l'espace de configuration FAI qui est alors disponible dans le répertoire `$FAI` ¹⁵. FAI ne se fonctionnera pas sans l'espace de configuration.

On peut accéder à l'espace de configuration à l'aide de diverses méthodes. Les méthodes supportées sont : `nfs:`, `file:`, `cvs:`, `svn:`, `git:`, `hg:`, `http:` et `detect:`. Voir `fai.conf(5)` pour une description détaillée de ces méthodes.

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 dans `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 est pris en compte pour définir des variables. Il doit contenir du code shell valide.

7.6 Partitionnement des disques locaux, création des systèmes de fichiers (tâche partition)

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

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 pour créer l'organisation des partitions du disque, du RAID logiciel, du LVM et pour créer les systèmes de fichiers. Lisez la page de manuel de `setup-storage(8)` pour une description détaillée et quelques exemples du format.

7.7 Pré-ensemencement Debconf (tâche debconf)

Les fichiers dans `$FAI/debconf` sont utilisés par le système de pré-ensemencement habituel de `debconf(7)` si les noms de fichier correspondent à un nom de classe.

¹⁴Celle-ci est définie dans la ligne de commande du noyau

¹⁵`/var/lib/fai/config`

7.8 Installation de paquets logiciels (tâche instsoft)

La commande `install_packages(8)` lit les fichiers de configuration à partir de `$FAI/package_config` d'une façon basée sur la classe et installe des paquets logiciels sur le nouveau système de fichiers.

Elle installe les paquets en utilisant `apt-get(8)`, `aptitude(1)`, `yum` ou d'autres gestionnaires de paquets sans besoin d'aucune interaction manuelle. Les dépendances des paquets sont également résolues par les gestionnaires de paquets.

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

De plus, FAI installera les paquets (*.deb* or *.rpm*) depuis les répertoires `$FAI/pkgs/<CLASSNAME>`.

7.9 Personnalisation spécifique au site (tâche configure)

Souvent, les configurations par défaut des paquets logiciels 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 `$FAI/scripts` inclut des exemples d'installation de machines Debian et Rocky Linux. Ils définissent le mot de passe root, ajoutent un compte utilisateur (défini par `$username`, avec `demo` par défaut), paramètrent le fuseau horaire, configurent le réseau avec DHCP ou en utilisant une adresse IP fixe, configurent 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'information sur ces scripts est disponible dans [\[cscripits\]](#).

7.10 Enregistrement des fichiers journaux (tâche savelog)

Lorsque toutes les tâches sont terminées, les fichiers journaux sont écrits dans `/var/log/fai/$HOSTNAME/install/`¹⁶ sur le nouveau système, et sur le compte sur le serveur d'installation si `$LOGUSER` est défini (il faut l'activer dans `/srv/fai/config/class-FAIBASE.var`). Il est également possible de spécifier un autre hôte comme destination pour cet enregistrement à l'aide de 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`).

De plus, deux liens symboliques seront créés pour indiquer le dernier répertoire où on a 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 journalisation 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 <https://fai-project.org/logs>.

7.11 Redémarrage du 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 redémarrerait 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`.

¹⁶`/var/log/fai/localhost/install/` est un lien vers ce répertoire.

8 Sujets avancés pour 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 clients d'installation

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 envoie ces messages dans sa sortie standard. On peut aussi utiliser une interface graphique, appelée `fai-monitor-gui(1)`.

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

8.3 Collecte d'adresses Ethernet pour plusieurs hôtes

Vous devez collecter toutes les adresses Ethernet (MAC) des clients d'installation et affecter un nom d'hôte et une adresse IP à chaque client. Pour collecter les adresses MAC, démarrez vos clients d'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 de l'absence de DHCP ou de TFTP), mais vous pouvez toujours collecter les adresses MAC.

Pendant que les clients d'installation démarrent, ils envoient des paquets de diffusion (broadcast) 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 envoyés quelques paquets de diffusion, arrêtez `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 associations NIS correspondantes). Avec ces informations, vous pouvez configurer votre démon DHCP (voir la section [bootdhcp](#)).¹⁷

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 dans les entrées de plusieurs fichiers journaux venant de `tftpd(8)` et `dhcpcd(8)` :

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

¹⁷Je recommande d'inscrire l'adresse MAC (les trois derniers octets devraient suffire si vous avez des cartes du même vendeur) et le nom d'hôte devant chaque boîtier.

8.4 Détails du démarrage PXE

Ici, nous décrivons les détails du démarrage PXE, qui sont nécessaires seulement 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*, disponibles 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. Voici la séquence d'une amorce PXE :

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

Exemple d'un fichier *pxelinux.cfg* :

```
default fai-generated

label fai-generated
kernel vmlinuz-6.1.0-17-amd64
append initrd=initrd.img-6.1.0-17-amd64 ip=dhcp root=/srv/fai/nfsroot rootovl 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. FAI utilise le fichier binaire *lpxelinux.0* qui s'occupe du chargement du noyau et du fichier *initrd* via FTP ou HTTP. La commande *fai-chboot* (8) prend cela en charge avec l'option *-U*.

8.5 Personnaliser la configuration de son serveur d'installation

- Miroir de paquets logiciels local/plus rapide
- Loguser différent
- Mot de passe root local 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 du *nfsroot* utilise la commande *debootstrap* (8) . Elle a besoin de l'emplacement d'un miroir Debian et du nom de la distribution (comme *bullseye*, *bookworm*, *sid*) pour laquelle le système Debian de base devrait être construit. N'utilisez pas de distributions différentes ici et dans */etc/fai/apt/sources.list*. Cela créerait un *nfsroot* dysfonctionnel.

NFSROOT_ETC_HOSTS

Cette variable est nécessaire seulement 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 aussi 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`.¹⁸

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 l'utiliser comme compte local pour enregistrer des fichiers journaux. Les fichiers journaux de tous les clients d'installation sont enregistrés dans le répertoire de personnel de ce compte. Vous devriez changer le groupe principal de ce compte, afin que ce compte ait des droits d'écriture sur `/srv/tftp/fai` afin d'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 de noyau sur `nfsroot`, ajoutez les options `-k` ou `-K` à `fai-make-nfsroot`. Cela ne recréera pas votre `nfsroot`, mais mettra à jour seulement vos noyaux et les modules du noyau dans le `nfsroot` ou ajoutera des paquets supplémentaires dans le `nfsroot`.

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

Vous pouvez facilement créer un CD d'installation (ou une clé USB) de 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 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) écrira ce miroir, le `nfsroot` et l'espace de configuration sur un CD amorçable. Et voilà !

Pour créer facilement le CD d'installation, on peut utiliser la commande suivante (Pour plus de détails de configuration voir `fai-cd` (8)) :

```
faiserver# fai-cd -m <partialMirrorDir> fai-cd.iso
```

Ce CD d'installation contient toutes les données nécessaires à l'installation. La commande `fai-cd` (8) écrit 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 <https://fai-project.org/fai-cd>.

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

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

`mk-data-partition` est un outil qui étend une image ISO (qui sera copiée sur une clé USB) ou une clé USB contenant une image ISO avec une partition `ext4` ou `exFAT` et y place le label `MY-DATA`. Cette partition est automatiquement montée dans `/media/data` par FAI. On peut copier ses propres paquets `.deb` dans cette partition de données sous les sous-répertoires `pkgs/<CLASSNAME>`. FAI installera alors ces paquets si la classe équivalente est définie.

exemple d'usage de `mk-data-partition`:

```
# mk-data-partition -s 1G -c faicd-large.iso A B
```

La commande ci-dessus crée la partition de données avec une taille de 1 Go dans le fichier ISO et y copie les répertoires A et B. On peut écrire l'image ISO modifiée sur une clé USB.

Si l'image ISO est déjà sur la clé USB, qui est disponible sous `/dev/sdf`, on peut ajouter la partition de données de type `exFAT` qui utilisera tout l'espace disponible de la clé USB en utilisant cette commande :

```
# mk-data-partition -F /dev/sdf
```

¹⁸À des fins de débogage, il peut être utile d'entrer dans l'environnement `chroot` à la main à l'aide de cette commande : `faiserver# chroot /srv/fai/nfsroot bash`

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 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 fonctionnel. L'image disque peut également être convertie au format qcow2. Vous n'avez pas besoin de mettre en place le `nfsroot` de FAI quand vous utilisez seulement `fai-diskimage`. Mais vous avez besoin d'un fichier de base dans votre espace de configuration. Le réglage de la variable `$FAI_BASEFILEURL` permet de télécharger automatiquement un fichier de base approprié dans votre espace de configuration.

Voici par exemple comment créer une image de disque brute pour un hôte nommé `cloud3`, avec un petit ensemble de paquets logiciels :

```
# export FAI_BASEFILEURL=https://fai-project.org/download/basefiles/
# fai-diskimage -vu cloud3 -S2G -cDEBIAN,BOOKWORM64,AMD64,FAIBASE,GRUB_PC,DHCPC,DEMO,CLOUD, ↵
    LAST disk.raw
```

Cette commande créera une image de disque nommée `ubuntu.qcow2` pour une configuration de bureau Ubuntu 16.04 avec pour nom d'hôte `machin`.

```
# export FAI_BASEFILEURL=https://fai-project.org/download/basefiles/
# cl=DHCPC,UBUNTU,JAMMY,JAMMY64,AMD64,XORG,LAST
# fai-diskimage -Nvu machin -S5G -c$cl ubuntu.qcow2
```

Vous pouvez essayer quelques images de disques sans installer FAI, si vous visitez <https://fai-project.org/FAIme/cloud>

8.8 Création d'une image vive amorçable

La création d'images ISO vives amorçables est facile avec FAI. Il suffit de deux étapes. Premièrement, créez votre environnement vif à l'aide de la commande `fai dirinstall`. N'oubliez pas d'ajouter la classe `LIVEISO`. Ensuite, créez l'image ISO vive en utilisant `fai-cd` :

```
# cl="DEBIAN,BOOKWORM64,AMD64,FAIBASE,XFCE,XORG,DHCPC,DEMO,LIVEISO,LAST"
# LC_ALL=C fai -v dirinstall -u xfce33 -c $cl \
    -s file:///srv/fai/config/srv/xfce
# fai-cd -s500 -MH -d none -g /etc/fai/grub.cfg.live \
    -n /srv/xfce/live.iso
```

Le `nfsroot` n'est pas nécessaire pour une image ISO vive. Il n'y a pas actuellement d'image vive ISO disponible sur le serveur d'installation.

8.9 Construire des images disque d'architecture différente (cross-architecture)

À partir de FAI 5.4 il devient possible de construire une image disque pour des architectures différentes de l'hôte qui est utilisé. Par exemple on peut construire une image pour une architecture ARM 64-bit (`aarch64`) sur un hôte qui fonctionne sur une architecture `amd64`. Voici les étapes successives :

```
# apt install qemu-system-arm qemu-user-static fai-server fai-setup-storage fai-doc qemu- ↵
    utils
# fai-mk-configspace

# export FAI_BASEFILEURL=https://fai-project.org/download/basefiles/
# fai-diskimage -vu armhost -S2G -cDEFAULT,DHCPC,DEBIAN,ARM64,BUSTER_ARM64,FAIBASE,DEMO, ↵
    CLOUD,LAST arm64.raw
# chown your_user_id arm64.raw
# cp /var/log/fai/armhost/last/vmlinuz* vmlinuz
# cp /var/log/fai/armhost/last/initrd.img* initrd
```

Ensuite on peut lancer qemu en tant qu'utilisateur ordinaire :

```
> qemu-system-aarch64 -m 1000 -M virt,gic_version=3 -cpu cortex-a57 -drive file=arm64.raw, \
    if=virtio,index=1 -no-reboot -nographic -name ARM64 -net nic,name=eth0,model=virtio -net \
    user,name=eth0,-kernel vmlinuz -initrd initrd -append "console=ttyAMA0 rw ip=dhcp root \
    =/dev/vda1 net.ifnames=0"
```

Ça fonctionne de même pour d'autres architectures. Il faut garder en tête que les réglages réseau de qemu ont des performances dégradées si on n'utilise pas le pilote virtio comme ci-dessus ou des TAPs réseaux.

On peut trouver les fichiers de base pour plusieurs architectures à <https://fai-project.org/download/basefiles/>, ou utiliser `mk-basefile` pour créer son fichier propre.

8.10 Système de sauvetage FAI

Si vous assignez à la variable `$FAI_ACTION` la valeur *sysinfo* (par exemple en utilisant `fai-chboot -S`), le client n'installera pas de nouveau système, mais collectera beaucoup d'informations système. Si vous assignez à `$FAI_ACTION` la valeur *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 une autre console de terminal, si vous avez ajouté `createvt` à `$FAI_FLAGS`.

Vous voilà maintenant avec un système Linux qui fonctionne 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`, `lsmod`, `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 -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	4.0M	0	4.0M	0%	/dev
tmpfs	2.0G	0	2.0G	0%	/dev/shm
tmpfs	783M	18M	766M	3%	/run
tmpfs	5.0M	0	5.0M	0%	/run/lock
LiveOS_rootfs	783M	18M	766M	3%	/
192.168.33.250:/srv/fai/config	59G	23G	24G	49%	/var/lib/fai/config
/dev/mapper/vg1-root	7.3G	1.8G	5.2G	26%	/target
/dev/vda1	459M	53M	378M	13%	/target/boot
/dev/mapper/vg1-home	1.4G	72K	1.3G	1%	/target/home

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

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

8.11 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` et `-M` pour éviter de créer un miroir partiel à l'intérieur de l'image.

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


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

Pour demander alors l'image squashfs, ajoutez ce qui suit à votre ligne de commande du noyau, par exemple dans votre fichier de configuration pxelinux pour le client avec fai-chboot.

```
root=live:http://faiserver/cskoeln/squash.img FAI_CONFIG_SRC=file:///var/lib/fai/config
```

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

8.12 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 UBUNTU2204.tar.xz par exemple. Un client d'installation appartenant à la classe UBUNTU2204 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 dépôt de paquets.

L'outil `rinse(8)` est utilisé pour créer des fichiers de base pour des distributions comme Rocky Linux, CentOS, openSUSE, ou Fedora. Certains fichiers de base peuvent être téléchargés à partir de <https://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.

8.13 Création d'environnements chroot et de virtualisation

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 y a 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 la cible d'installation peut être spécifié à l'aide de `-u <nom-d-hote>`

On peut utiliser cela, par exemple, pour combiner FAI avec les outils *xen-tools*, qui vous aident à construire des domaines invités Xen. Les *xen-tools* sont très agréables pour générer des fichiers de configuration et bloquer des périphériques pour de nouveaux invités sur la base de commandes simples et/ou des fichiers de configuration, mais ils ne peuvent assigner qu'un seul rôle par installation pour la personnalisation. Les utilisateurs de FAI ont besoin de plus et en veulent, car ils ont l'habitude d'avoir le système de classes. Ils peuvent y arriver 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, il faut définir la variable `install=0` dans la configuration *xen-tools* pour cet hôte.

8.14 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 paquets logiciels et l'appel des scripts de personnalisation.

Pour lancer un *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` (ça se fait dans un fichier `class/*.var`) si FAI doit enregistrer les fichiers journaux sur une machine distante.

C'est à vous de décider, comment démarrer un `softupdate` sur un plus grand nombre d'hôtes. Vous pouvez faire le `softupdate` sur régulièreent 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
```

Au lieu de ça, utilisez la commande `ainsl`(1) dans un script shell ou utilisez la fonction de `cfengine` `AppendIfNoSuchLine`.

Toutes les commandes du script de personnalisation doivent être capables de modifier le système de fichiers cible qui'il soit disponible dans `/target` lors de l'installation initiale ou qu'il soit 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, on a 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 avant les 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. Elle contient l'action actuellement exécutée: `install`, `softupdate`, `dirinstall`, `sysinfo`, `inventory` ou votre propre action définie.

8.15 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 ce `nfsroot` 32 bits sur un serveur d'installation qui fonctionne sous `amd64` est 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) qui est 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
```

Et voilà !

8.16 Arrêt d'urgence de l'installation quand une erreur se produit

Chaque tâche, chaque hook peut appeler la fonction `task_error` pour envoyer une valeur d'erreur à l'installation. Cette erreur apparaîtra dans le fichier de journalisation et peut être visionnée dans le `fai-monitor-gui` (1). La valeur de l'erreur est aussi examinée en tenant compte de la variable `$STOP_ON_ERROR` qui est réglée à 700 par défaut. Si la valeur d'erreur est supérieure à `$STOP_ON_ERROR`, l'installation s'arrêtera immédiatement. Dans un hook, ajoutez simplement un appel comme celui-ci :

```
task_error <valeur>
```

C'est commode d'appeler `task_error` avec `$?` comme deuxième paramètre, par exemple si on veut émettre une valeur d'erreur de 123 on peut faire ainsi

```
<some command>
task_error 123 $?
```

Alors, l'erreur de tâche n'est mise en place que si la commande précédente a échoué avec une certaine erreur. La même syntaxe peut être utilisée par les scripts dans le sous-répertoire de classe qui se terminent en `.sh`.

Un cas particulier est celui des scripts dans le sous-répertoire `class/` qui ne se termine pas en `.sh`. Dans ces scripts il faut appeler `task_error` avec 1 comme troisième paramètre. Par exemple on peut utiliser deux variantes pour régler la valeur à 123

```
task_error 123 $? 1
```

ou

```
task_error 123 1 1
```

La dernière commande met toujours l'erreur de tâche à 123 sans tenir compte de `$?`. Si la valeur d'erreur est supérieure à `$STOP_ON_ERROR`, l'installation sera immédiatement stoppée.

Les valeurs d'erreur sont regroupées dans quatre catégories

normale ou messages d'information~:	1xx, 2xx
avertissements~:	3xx, 4xx
erreurs mineures~:	5xx, 6xx
erreurs~:	7xx, 8xx

Le `fai-monitor-gui` possède 4 icônes différentes pour ces catégories d'erreur.

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`). Certaines sont des scripts shell externes situés dans `/usr/lib/fai/`. Elles sont appelées via un sous-programme supérieur nommé `task`. Ce sous-programme appelle les hooks si disponibles, puis appelle la tâche (définie comme `task<nom>_()`). Une tâche et ses hooks peuvent être ignorés à la demande en utilisant la commande `skiptask()`.

Voici 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. Les paramètres de réseau sont récupérés d'un serveur DHCP et sont enregistrés dans `boot.log`. La configuration pour la résolution DNS est créée.

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 de 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 ssh est lancé à la demande pour les connexions à distance.

defclass

Appels de `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

Source 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`. Elle affiche des informations sur le matériel détecté et monte les disques durs locaux en lecture seule sur `/target/partitionname` ou en tenant compte 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 <https://fai-project.org/logs/>.

dirinstall

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

softupdate

Cette tâche, exécutée à l'intérieur d'un système en cours d'exécution via la commande en ligne `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 les partitions et périphériques 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

Monte 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 puisse y être introduit. Par défaut, le fichier tar de 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 récupérer le fichier de base via FTP, HTTPS ou HTTP si celle-ci est définie.

debconf

Appelle `fai-debconf(1)` pour définir les valeurs de la base de données de pré-ensemencement de `debconf`.

repository

Prépare l'accès au dépôt de paquets en préparant la configuration `apt`. Cela peut également ajouter des clés de dépôts en tenant compte des classes à partir de fichiers comme `CLASSNAME.gpg` 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 contrefait 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 soit démarré pendant l'installation.

instsoft

Installe les paquets logiciels 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

Attend que les travaux en arrière-plan se terminent (par exemple, qu'`emacs` compile des fichiers lisp) puis 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'un petit media vif autodiscover FAI (CD, USB, etc.), qui peut être créé 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 fait l'option `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 lancer `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 utilitaire `mkdebmirror` ¹⁹ peut être utilisé pour créer votre propre miroir Debian local. Ce script utilise la commande `debmirror (1)`. Un miroir Debian partiel pour l'architecture amd64 pour Debian 11 et 12 (noms de code bullseye et bookworm) sans les paquets source nécessite environ 180 Go 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 <https://wiki.fai-project.org/index.php/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 vers le 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`. Elle est définie après l'appel de `set_disk_info`.

¹⁹On peut trouver ce script dans `/usr/share/doc/fai-doc/examples/utills/`

- Il y a de nombreuses fonctions pour générer un liste personnalisée de disques. Voir `fai-disk-info` pour un exemple.
 - `set_bootstick()`
 - `grepv_disks()`
 - `grep_disks()`
 - `notmatchdisks()`
 - `matchdisks()`
 - `smallestdisk()`
 - `largestdisk()`
 - `all_disks_by_size()`
 - `all_disks_and_size()`
 - `once_only()`
 - `checkdisk()`
 - `disks_by_id()`
- Utilisez `fai-divert -a` si un script `postinst` appelle un programme de configuration, par exemple le script `postinst` pour le paquet `apache` appelle `apacheconfig`, qui nécessite une entrée manuelle. Vous pouvez contrefaire le programme de configuration pour que l'installation puisse être entièrement automatique.
- Parfois, l'installation semble s'arrêter, mais souvent il y a 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 ?
Lire la page de manuel de `fai-class(8)`. Si vous voulez définir quelques classes supplémentaires (par exemple A; B, C) dans la 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` de noyau à l'aide de `make-kpkg(8)` et utilisez l'option `--initrd`. Copiez ce paquet Debian dans un dépôt local et ajoutez-le à `/etc/fai/sources.list`. Ajoutez le nom de votre paquet à `/etc/fai/NFSROOT`. Ensuite appelez

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

- Comment utiliser le `nfsroot` comme système pour les clients sans disque ?
https://wiki.fai-project.org/index.php/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-buster` et `/etc/fai-bookworm`. Ensuite, vous devez définir les variables `$NFSROOT` dans différents répertoires et lancer

```
faiserver#fai-make-nfsroot -C /etc/fai-buster
```

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 `faiend`. Ceci est vrai pour les installations par réseau ainsi que pour les installations par media amorceable.

9.8 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`.

Des exemples de fichiers journaux d'ordinateurs installés avec succès sont disponibles sur <https://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)`.

scripts.log

Sortie de tous les scripts, qui sont 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 on démarre, 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 dans tous les fichiers journaux 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 devrez rechercher plus de détails dans tous les fichiers journaux.

9.9 Comment utiliser HTTP pour le démarrage PXE

`fai-make-nfsroot` utilise maintenant le fichier binaire `lpxelinux.0` qui prend en charge déjà le transfert du noyau et du fichier `initrd` via http (en plus de tftp). Il vous suffit d'activer un 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 l'option `debug` activée.

```
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 le fichier `initrd` dans le `nfsroot` (`lsinitrd`) pour voir 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`` puis reconstruisez le fichier `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+=`.

Voici 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
```