PROJET FOG — DOCUMENT B

Installation et configuration du serveur

Mode opératoire et état de référence — Lab TSSR / Infrastructure Proxmox

Auteur	Julien BARBET
Version	1.0
Date	28 août 2026
Serveur	SRV-FOG-01 — VM 107 — 192.168.1.207
Système	Debian 12 Bookworm
Applicatif	FOG Project 1.5.10.2254
Hyperviseur	Proxmox VE 9.2.2
Domaine	lab.julien.local
Contexte	Lab personnel de formation

Sommaire

1. Objet et périmètre

Ce document décrit la construction du serveur de déploiement SRV-FOG-01 : la machine virtuelle qui l'héberge, le système Debian qui la porte, le volume dédié aux images, les mesures de durcissement, l'installation de FOG Project et le service ProxyDHCP qui rend l'amorçage réseau possible sans modifier le réseau existant.

Il s'arrête au serveur en état de fonctionner. Tout ce qui suit — la fabrication d'une image, son déploiement, l'installation logicielle — relève des documents C, D et E.

1.1 Découpage de la documentation

Document	Objet	Lecteur visé
A	Présentation du projet, choix structurants, périmètre	Décideur, responsable technique
B	**Installation et configuration du serveur**	**Administrateur système, repreneur du service**
C	Préparation du master Windows 11 et capture de l'image	Technicien poste de travail
D	Déploiement d'images : poste unitaire et flotte	Technicien de déploiement
E	Snapins et script de post-déploiement	Technicien, scripteur

1.2 État de référence

Les valeurs ci-dessous décrivent le serveur tel qu'il fonctionne. Elles servent de référence à toute reconstruction ou reprise du service.

Élément	Valeur
Hyperviseur	Proxmox VE 9.2.2 — GMKtec M5 Ultra, Ryzen 7 7730U, 32 Go
Machine virtuelle	VM 107 — SRV-FOG-01
Système	Debian GNU/Linux 12 (bookworm), noyau 6.1.0-52-amd64
Applicatif	FOG Project 1.5.10.2254
Branche du dépôt	`stable`
Commit installé	`9f3b5a896` — 11 août 2026
Emplacement du dépôt	`/root/fogproject`
Adresse IP	192.168.1.207/24, statique
Passerelle	192.168.1.254 (box opérateur)
Résolution DNS	192.168.1.201 (SRV-AD-01), puis 192.168.1.254

Élément	Valeur
Interface d'administration	http://192.168.1.207/fog/management
Accès distant	SSH sur le port 17251
Stockage des images	/images — 196 Go utiles sur disque dédié
Service d'amorçage	dnsmasq en mode ProxyDHCP

1.3 Nature des relevés

Les captures de ce document ne sont pas des captures d'installation : ce sont des **relevés de vérification**, effectués sur la machine en service. Chaque chapitre présente donc l'état constaté, accompagné de la commande qui permet de le reproduire.

> **Pourquoi ce choix de documentation est défendable**
>
> Une capture d'assistant d'installation prouve qu'un écran a été affiché ; elle ne prouve pas que le résultat est conforme. Un relevé `ufw status numbered` ou `df -h /images` prouve l'état réel du système au moment du contrôle, et reste reproductible par un tiers. Pour un document destiné à la reprise d'un service, la seconde forme est la plus utile : elle donne au repreneur les commandes de contrôle en même temps que l'état attendu.

1.4 Conventions

- Les commandes précédées de `root@SRV-FOG-01:~#` exigent les privilèges d'administration, obtenus par `su -` (voir chapitre 6).
- Les commandes précédées de `julien@SRV-FOG-01:~$` s'exécutent avec un compte non privilégié.
- Les commandes précédées de `root@pve:~#` s'exécutent sur l'hyperviseur.
- Les mots de passe sont masqués par `***`.

2. Architecture et choix techniques

2.1 Le besoin

Installer Windows poste par poste consomme environ une heure par machine, produit des configurations divergentes et ne se documente pas. Le déploiement d'images répond à ces trois problèmes : une installation de référence est fabriquée une fois, contrôlée, puis dupliquée à l'identique sur le parc. Le temps d'intervention par poste tombe à quelques minutes, et la conformité devient une propriété de l'image plutôt qu'une discipline individuelle.

Trois fonctions sont nécessaires pour cela : un serveur capable de démarrer un poste sans système d'exploitation, un stockage pour les images, et un mécanisme de personnalisation après clonage. FOG Project fournit les trois.

2.2 Comparaison avec la solution Microsoft

L'alternative naturelle en environnement Active Directory est le couple WDS (Windows Deployment Services) et MDT (Microsoft Deployment Toolkit). Le tableau suivant compare les deux approches sur les critères qui ont orienté la décision.

Critère	FOG Project	WDS + MDT
Nature de l'image	Copie des partitions du disque (Partclone). L'image reproduit le disque tel qu'il a été fabriqué.	Image de fichiers (WIM), rejouée par une séquence de tâches qui réinstalle Windows.
Système hôte	Debian, aucune licence	Windows Server, licence requise
Coût	Nul — GPL v3	Coût de la licence serveur
Intégration DHCP	Possible en ProxyDHCP : le DHCP existant n'est pas touché	Exige la maîtrise du DHCP, ou les options 66/67, ou le relais IP helper
Systèmes pris en charge	Indifférent au système : Windows, Linux, tout ce qui tient sur un disque	Windows uniquement
Diffusion multiple	Multicast natif par UDPCAST, piloté par groupes d'hôtes	Multicast natif WDS
Personnalisation après clonage	Client FOG installé dans l'image : renommage, jonction au domaine, snapins	Séquence de tâches MDT, plus riche mais plus complexe
Gestion des pilotes	Pas de gestion native structurée par modèle : l'image doit convenir au matériel cible.	Gestion native structurée des pilotes par modèle de machine
Pérennité de l'outil	Projet communautaire actif	MDT n'est plus développé par Microsoft, qui oriente vers Intune et Autopilot

FOG a été retenu pour trois raisons. La première est l'absence de contrainte sur le DHCP existant : le lab est adossé à une box opérateur dont le service DHCP ne peut pas être remplacé sans risque pour le réseau domestique. La seconde est le coût nul, licence comme matériel — le serveur tient

sur une machine virtuelle de 4 Go. La troisième est la simplicité du modèle : capturer un disque et le réécrire est un mécanisme que l'on peut expliquer et diagnostiquer entièrement, là où une séquence de tâches MDT introduit une couche d'abstraction supplémentaire.

> **La limite assumée : les pilotes**
>
> C'est le point faible de l'approche par image disque. Une image fabriquée sur un modèle de machine ne contient que les pilotes de ce modèle, et FOG n'offre pas de mécanisme natif structuré pour les injecter par modèle. Sur un parc hétérogène, il faut soit une image par modèle, soit une injection de pilotes en post-déploiement — deux réponses que MDT traite nativement et que FOG laisse à l'administrateur. Dans le périmètre de ce lab, où toutes les machines cibles sont des machines virtuelles au matériel identique, la question ne se pose pas. Sur un parc réel, elle serait le premier critère à réexaminer.

2.3 Rôles portés par le serveur

Un serveur FOG n'est pas un service unique : c'est un assemblage de services standards, orchestrés par une application web. Le tableau suivant donne la correspondance entre chaque service et sa fonction.

Service	Fonction dans la chaîne de déploiement
dnsmasq (ProxyDHCP)	Indique au poste qui démarre où trouver son fichier d'amorçage
TFTP	Sert le chargeur d'amorçage et le script iPXE, avant tout système
Apache + PHP-FPM	Interface d'administration, menu d'amorçage, dialogue avec le système temporaire du client
MariaDB	Base de données : hôtes, images, groupes, tâches, snapins
NFS	Expose `/images` au système temporaire du client pendant la capture et le déploiement
FTP	Transferts internes de FOG et gestion des fichiers d'image
FOGMulticastManager	Pilote les sessions de diffusion multiple (UDPCAST)
FOGScheduler	Déclenche les tâches planifiées
FOGImageReplicator	Réplique les images entre nœuds de stockage (un seul nœud ici)

2.4 Séquence d'amorçage réseau

C'est l'enchaînement le plus délicat de l'ensemble, parce qu'il fait intervenir deux serveurs DHCP qui répondent à la même requête sans se concurrencer. Le déroulement complet est le suivant.

Étape	Acteur	Échange
1	Poste client	Émet un DHCP Discover depuis l'adresse `0.0.0.0`, en s'annonçant comme `PXEClient` et en déclarant son architecture (option 93)

Étape	Acteur	Échange
2	Box — 192.168.1.254	Attribue l'adresse IP, le masque, la passerelle. Elle ne connaît aucun fichier d'amorçage.
3	dnsmasq — 192.168.1.207	Répond en mode proxy : aucune adresse, mais le nom du serveur d'amorçage et le fichier à charger, choisi selon l'architecture déclarée
4	Poste client	Télécharge `secureboot/snponly-shimx64.efi` en TFTP depuis 192.168.1.207
5	Poste client	Le chargeur démarre iPXE, qui relance une requête DHCP en se signalant cette fois par sa classe utilisateur `iPXE`
6	dnsmasq	Reconnaît la classe iPXE et répond cette fois `tftp://192.168.1.207/default.ipxe`
7	Poste client	Exécute le script iPXE, qui interroge l'interface web du serveur en HTTP et affiche le menu FOG
8	Poste client	Selon la tâche en attente, charge le système temporaire FOS, monte `/images` en NFS et lance Partclone

Pourquoi deux requêtes DHCP successives

La première requête provient du micrologiciel UEFI du poste, qui ne sait faire qu'une chose : télécharger un fichier en TFTP et l'exécuter. Ce fichier est iPXE, un environnement d'amorçage bien plus capable — HTTP, scripts, menus. Mais iPXE démarre sans configuration réseau héritée : il refait donc sa propre requête DHCP. Cette seconde requête est reconnaissable à sa classe utilisateur, ce qui permet à dnsmasq de lui répondre autre chose qu'au premier tour. Sans cette distinction, iPXE se verrait servir iPXE, indéfiniment.

Pourquoi ce serveur est en HTTP, et jusqu'à quand

Le fichier `snponly-shimx64.efi` est un chargeur signé par Microsoft, seul moyen d'amorcer un poste dont le Secure Boot est actif. Ce chargeur ne valide pas les chaînes de certificats auto-signées : un serveur FOG en HTTPS avec un certificat de lab devient inaccessible aux clients qui passent par lui. **Avec FOG 1.5.10 et le mode d'installation retenu ici, HTTPS n'a donc pas été activé, afin de préserver la chaîne Secure Boot.**

Deux précisions importantes. Cette contrainte est propre à cette version et à cette configuration : **elle doit être réévaluée lors d'un passage à FOG 1.6**, dont la gestion du transport et des certificats diffère. Et même en 1.5.10, l'exclusion n'est pas absolue : un certificat émis par une autorité reconnue du parc, ou l'enrôlement d'une clé MOK sur les postes, lèverait la contrainte. C'est un choix de configuration, pas une impossibilité technique.

Pourquoi le même fichier pour tous les clients UEFI 64 bits

La configuration sert `snponly-shimx64.efi` aux architectures `00007` et `00009`, sans distinguer les postes dont le Secure Boot est actif de ceux où il ne l'est pas. Ce n'est pas un raccourci : **un serveur DHCP ne peut pas connaître l'état du Secure Boot d'un client**, cette information ne figure dans aucune option de la requête. Servir à tous le chargeur compatible Secure Boot est la seule stratégie qui fonctionne dans les deux cas.

Le chargeur passe, le noyau non

Un essai mené avec Secure Boot réellement activé sur un poste de test délimite précisément la compatibilité. Le chargeur `snponly-shimx64.efi` est accepté, iPXE démarre, le ProxyDHCP répond, le menu FOG s'affiche. Puis le noyau et le système de fichiers initial sont téléchargés sans erreur — et leur exécution est **refusée pour violation de la politique de sécurité**.

La cause : le noyau FOS n'est pas signé par une clé que reconnaît le chargeur. **Les déploiements de ce lab sont donc menés Secure Boot inactif**, sur des machines créées sans clés pré-enregistrées. Le menu FOG s'affichant normalement dans les deux cas, l'écart ne se voit qu'en poussant le test jusqu'au chargement de FOS — d'où l'insistance du chapitre 9 sur ce point.

3. Configuration de la machine virtuelle

La configuration de la VM 107 est relevée sur l'hyperviseur :

```
root@pve:~# qm config 107
```

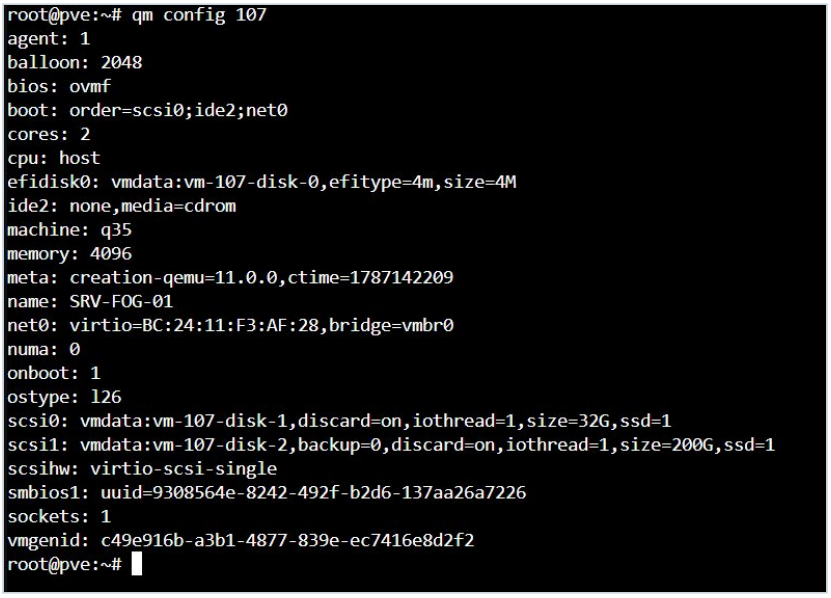

Figure 1 — Configuration complète de la VM 107 (SRV-FOG-01)

3.1 Paramètres et justifications

Paramètre	Valeur	Justification
cores / sockets	2 / 1	La charge du serveur est dominée par les entrées-sorties disque et réseau, pas par le calcul. Deux cœurs suffisent, y compris en multicast.
cpu	host	Expose le jeu d'instructions réel du processeur, ce qui accélère la compression ZSTD lors des captures.
memory / balloon	4096 / 2048 Mo	4 Go pour MariaDB, Apache et PHP-FPM. Le ballon à 2048 Mo autorise l'hyperviseur à récupérer jusqu'à 2 Go quand le serveur est au repos, ce qui compte sur un hôte de 32 Go partagé entre huit machines.
bios	ovmf	Micrologiciel UEFI, avec `efidisk0` de 4 Mo pour la NVRAM.
machine	q35	Jeu de composants moderne, PCI Express.
ostype	l26	Noyau Linux 2.6 ou supérieur : ajuste les horloges et périphériques virtuels.
boot	order=scsi0;ide2;net0	Disque système en tête. Le serveur ne doit jamais amorcer en réseau.

Paramètre	Valeur	Justification
onboot	1	Démarrage automatique avec l'hyperviseur : le service d'amorçage doit être disponible sans intervention après une coupure.
net0	virtio, pont vmbr0	Pilote paravirtualisé, le plus performant. Voir l'encadré ci-dessous.
scsi0	32 Go, discard=on, iothread=1, ssd=1	Disque système.
scsi1	200 Go, backup=0, discard=on, iothread=1, ssd=1	Volume des images, exclu des sauvegardes. Voir l'encadré ci-dessous.
scsihw	virtio-scsi-single	Un contrôleur par disque, condition d'efficacité de `iothread`.
agent	1	Agent QEMU : arrêt propre et remontée de l'adresse IP dans l'interface.

Pourquoi VirtIO ici, alors que les postes clients exigent E1000

La contrainte E1000 documentée dans le rapport C ne concerne que les machines qui **amorcent en réseau** : l'environnement de préamorçage iPXE ne reconnaît pas nativement une carte VirtIO, et le micrologiciel OVMF fige ses entrées d'amorçage à la première énumération du matériel. Le serveur FOG, lui, démarre sur son disque et n'exécute jamais de PXE. Rien ne s'oppose donc au pilote paravirtualisé, qui offre un meilleur débit — ce qui compte pour une machine dont le métier est de transférer des images de 16 Go.

Pourquoi backup=0 sur le disque des images

Le volume `/images` contient des données volumineuses et reconstructibles : une image perdue se recapture depuis le master en une vingtaine de minutes. L'inclure dans la sauvegarde vers Proxmox Backup Server ferait passer chaque sauvegarde du serveur de quelques gigaoctets à plus de deux cents, pour une donnée dont la valeur ne le justifie pas. L'exclusion porte sur le disque, pas sur la machine : le disque système, qui contient la base MariaDB et donc la configuration des hôtes, des groupes et des snapins, reste sauvegardé.

Ce que cette exclusion implique en cas de restauration

Restaurer SRV-FOG-01 depuis une sauvegarde rend un serveur fonctionnel, configuré, connaissant tous ses hôtes — mais dont le répertoire `/images` est vide. La procédure de reprise comporte donc une étape supplémentaire : recapturer les images depuis les masters, ou les restaurer depuis une copie externe. Ce point doit figurer dans le plan de reprise, faute de quoi la restauration paraît réussie alors qu'aucun déploiement n'est possible.

4. Installation du système Debian

L'installation est réalisée depuis l'image netinst de Debian 12, en mode expert graphique. Les écrans de l'assistant ne sont pas reproduits ici : ils n'apportent rien qu'un tableau de paramètres ne dise plus clairement.

4.1 Paramètres retenus

Étape de l'assistant	Valeur	Motif
Langue / pays / clavier	Français — France — AZERTY	Cohérence avec le reste du lab
Nom de machine	SRV-FOG-01	Convention du lab : rôle, puis numéro d'instance
Domaine	lab.julien.local	Suffixe DNS, sans jonction au domaine
Configuration réseau	Statique — 192.168.1.207/24	Un serveur d'amorçage doit être joignable à une adresse fixe : elle est inscrite en dur dans la configuration ProxyDHCP
Mot de passe root	Défini	Voir chapitre 6 : l'escalade se fait par `su -`
Compte utilisateur	julien	Compte d'administration courante, non privilégié
Partitionnement	Disque système entier, LVM non utilisé	Le redimensionnement se fait au niveau de l'hyperviseur ; LVM ajouterait une couche sans bénéfice ici
Sélection de logiciels	Utilitaires usuels du système + serveur SSH uniquement	Aucun environnement de bureau : le serveur n'est administré qu'à distance

4.2 Vérification de l'identité du système

```
julien@SRV-FOG-01:~$ hostnamectl; echo "---"; cat /etc/os-release | head -3; echo "---"; uptime
```

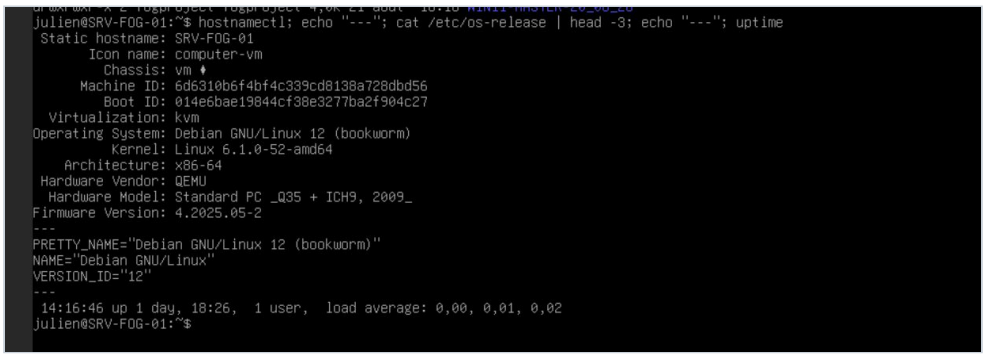

Figure 2 — Identité du système : nom, version de Debian, noyau, virtualisation

Quatre éléments sont à contrôler : le nom statique `SRV-FOG-01`, le système **Debian GNU/Linux 12 (bookworm)**, le noyau `6.1.0-52-amd64`, et la virtualisation `kvm` qui confirme que la machine est bien celle attendue sur l'hyperviseur.

> **Révision validée sur Debian 12 ; les essais sur Debian 13 ont échoué**
>
> La révision 1.5.10.2254 a été validée ici sur Debian 12 Bookworm. Le choix n'est pas un conservatisme : les essais menés sur Debian 13 Trixie ont échoué sur trois points distincts. Le paquet `sysv-rc-conf`, que le script tente d'installer, a disparu des dépôts. Une erreur liée à `libcurl4` interrompt la résolution des dépendances. Enfin la variable `php_ver` est codée en dur à la valeur 8.2 dans le script d'installation, alors que Trixie livre PHP 8.4 : les paquets demandés n'existent pas. Le problème a été suivi sous le numéro 797 du projet, désormais clos.
>
> **Ces constats valent pour la révision installée, à la date de l'essai.** Le comportement peut changer avec une révision ultérieure de FOG : la compatibilité doit être revérifiée avant toute réinstallation comme avant toute montée de version du système, plutôt que déduite de ce document.

4.3 Vérification de la configuration réseau

```
julien@SRV-FOG-01:~$ ip -4 addr show | grep -A2 "state UP"; echo "---"; ip route; echo "---";
cat /etc/resolv.conf
```

Figure 3 — Adressage, route par défaut et résolution de noms

Élément	Valeur constatée	Attendu
Interface	enp6s18, state UP	Interface unique
Adresse	192.168.1.207/24	Adresse fixe référencée dans dnsmasq
Route par défaut	via 192.168.1.254	Box opérateur
Domaine de recherche	lab.julien.local	Résolution des noms courts du lab
Serveur DNS primaire	192.168.1.201	SRV-AD-01, contrôleur de domaine
Serveur DNS secondaire	192.168.1.254	Box, pour la résolution externe en cas d'indisponibilité du DC

> **Pourquoi le contrôleur de domaine en DNS primaire**

Le serveur FOG n'est pas joint au domaine, mais il doit résoudre les noms internes : le script de post-déploiement vérifie la disponibilité du contrôleur avant de tenter une jonction, et les diagnostics passent par les noms plutôt que par les adresses. Interroger d'abord SRV-AD-01 donne accès à la zone `lab.julien.local` . La box en second assure la résolution des noms publics — nécessaire aux mises à jour du système et au téléchargement des sources de FOG.

Une chaîne de résolution à surveiller

Un résolveur secondaire n'est pas un basculement : il est interrogé quand le premier ne répond pas, mais aussi, selon les implémentations, quand le premier répond *négativement* avec un délai. Si SRV-AD-01 devient lent, certaines résolutions internes peuvent partir vers la box, qui ne connaît pas la zone du lab, et échouer silencieusement. C'est un compromis accepté ici pour garantir l'accès aux dépôts Debian en cas d'arrêt du contrôleur ; en production, la réponse serait deux contrôleurs de domaine et aucun résolveur externe.

5. Préparation du volume des images

Les images sont stockées sur un second disque virtuel, indépendant du disque système. Cette séparation répond à trois objectifs : exclure les images de la sauvegarde sans exclure la configuration, permettre l'agrandissement du stockage sans toucher au système, et éviter qu'un volume d'images saturé rende le système inutilisable.

5.1 État constaté

```
julien@SRV-FOG-01:~$ lsblk; echo "---"; df -h /images; echo "---"; grep images /etc/fstab
```

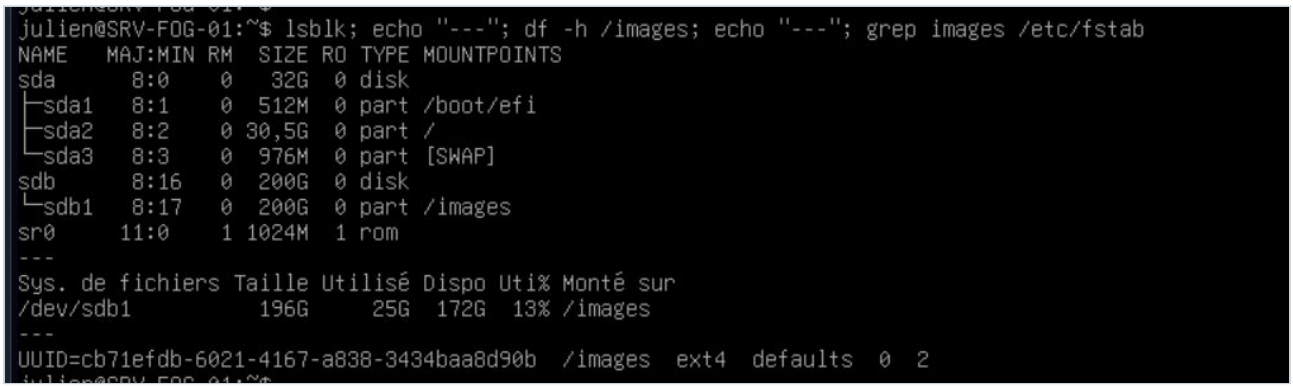

Figure 4 — Disques, occupation du volume des images et montage permanent

Élément	Valeur
Disque système	sda — 32 Go
sda1	512 Mo — partition système EFI, montée sur /boot/efi
sda2	30,5 Go — racine
sda3	976 Mo — swap
Disque des images	sdb — 200 Go
sdb1	200 Go — ext4, montée sur /images
Occupation	25 Go utilisés sur 196 Go — 13 %
Montage	UUID=cb71efdb-6021-4167-a838-3434baa8d90b /images ext4 defaults 0 2

5.2 Préparation du volume

Le disque a été partitionné en une partition unique, formaté en ext4, puis déclaré au montage permanent :

Prérequis — identification du disque cible. Les commandes qui suivent détruisent le contenu du disque visé. Le disque doit être identifié avant toute chose, par sa taille, son modèle et l'absence de point de montage :

```
root@SRV-FOG-01:~# lsblk -o NAME,SIZE,MODEL,TYPE,MOUNTPOINTS
root@SRV-FOG-01:~# lsblk -no MOUNTPOINTS /dev/sdb   # doit ne rien retourner
```

Action. Partitionnement, formatage et suppression de la réserve système :

```
root@SRV-FOG-01:~# fdisk /dev/sdb          # une partition primaire unique
root@SRV-FOG-01:~# mkfs.ext4 /dev/sdb1
root@SRV-FOG-01:~# tune2fs -m 0 /dev/sdb1
root@SRV-FOG-01:~# UUID=$(blkid -s UUID -o value /dev/sdb1)
root@SRV-FOG-01:~# mkdir -p /images
```

Action — déclaration du montage permanent. La ligne est sauvegardée puis ajoutée une seule fois, ce qui rend l'opération rejouable sans créer de doublon :

```
root@SRV-FOG-01:~# cp /etc/fstab /etc/fstab.bak-$(date +%F)
root@SRV-FOG-01:~# grep -q "$UUID" /etc/fstab ||     echo "UUID=$UUID /images ext4 defaults 0 2"
>> /etc/fstab
```

Contrôle avant montage, puis montage :

```
root@SRV-FOG-01:~# findmnt --verify --verbose
root@SRV-FOG-01:~# mount -a && df -h /images
```

Résultat attendu : `findmnt --verify` ne signale aucune erreur, `mount -a` ne retourne rien, et `df` affiche `/images` monté avec sa capacité pleine.

> **Trois précautions avant de lancer ces commandes**
>
> `fdisk` et `mkfs.ext4` sont des **opérations destructives** : elles effacent sans confirmation le contenu du disque visé, et une erreur de lettre porte sur le disque système.
>
> **Vérifier le disque, pas son nom.** Le contrôle `lsblk` ci-dessus confirme la taille (200 Go) et l'absence de point de montage. Un disque déjà monté n'est jamais le bon.
>
> **Ne jamais ajouter une ligne à `fstab` en aveugle.** Un `echo >>` répété lors d'une reprise duplique la ligne, ce qui fait échouer le montage au démarrage suivant. Le `grep -q ... ||` ci-dessus rend l'ajout idempotent, et `findmnt --verify` détecte une erreur de syntaxe **avant** qu'elle n'empêche le serveur de redémarrer.

> **Pourquoi monter par UUID et non par /dev/sdb1**
>
> Les noms `/dev/sdX` sont attribués dans l'ordre de détection des disques par le noyau. Ajouter un disque à la machine virtuelle, ou modifier l'ordre des contrôleurs, peut suffire à ce que le disque des images devienne `/dev/sdc`. La ligne de `fstab` pointerait alors vers le mauvais volume, ou vers rien. L'UUID est inscrit dans le système de fichiers lui-même : il suit le volume quoi qu'il arrive.

> **Pourquoi tune2fs -m 0**

Par défaut, ext4 réserve 5 % de la capacité à l'utilisateur root. Ce mécanisme protège les systèmes de fichiers système, où la saturation empêcherait les services et l'administrateur de fonctionner. Sur un volume de données pur, il immobilise 10 Go sans contrepartie. Le relevé confirme l'effet du réglage : 25 Go utilisés et 172 Go disponibles sur 196 Go utiles, soit une somme cohérente — avec la réserve par défaut, il manquerait une dizaine de gigaoctets à l'appel.

Le passage à 0 supprime aussi le filet de sécurité

La contrepartie doit être connue : plus rien n'empêche `/images` d'être rempli à 100 %. Une capture qui échoue faute de place laisse une image incomplète dans le répertoire, qu'il faut supprimer manuellement. La surveillance de l'occupation de ce volume est donc une tâche d'exploitation à part entière, et non un confort — c'est l'un des points à intégrer à la supervision.

5.3 Propriété du répertoire

L'installeur de FOG crée le compte système `fogproject` et lui attribue la propriété de `/images`. Ce point se vérifie après installation :

```
root@SRV-FOG-01:~# ls -ld /images
root@SRV-FOG-01:~# ls -l /images
```

Un répertoire d'image appartenant à un autre compte que `fogproject` est le symptôme d'une copie manuelle mal faite : la capture suivante échouera sur un refus d'écriture.

6. Durcissement du serveur

Trois mesures ont été appliquées avant l'installation de FOG : le déplacement du service SSH sur un port non standard, une politique de pare-feu en refus par défaut, et une séparation stricte entre le compte d'administration courante et les privilèges.

6.1 Service SSH

Le service d'accès distant écoute sur le port **17251**. La modification porte sur `/etc/ssh/sshd_config` :

```
Port 17251
PermitRootLogin no
```

La connexion s'établit ensuite en précisant le port :

```
ssh -p 17251 julien@192.168.1.207
```

> **Ce qu'apporte — et n'apporte pas — un port non standard**
>
> Déplacer SSH ne renforce pas le protocole : un attaquant qui balaie les 65 535 ports trouvera le service. Le bénéfice est ailleurs et il est réel : il supprime la quasi-totalité du bruit des robots, qui ciblent le port 22 sans exploration préalable. Les journaux d'authentification redeviennent lisibles, et une tentative qui y apparaît mérite alors une attention véritable. C'est une mesure d'hygiène opérationnelle, pas une mesure de sécurité au sens strict — elle ne dispense d'aucune autre. Le refus de connexion directe du compte root, lui, en est une : il oblige tout accès privilégié à passer d'abord par un compte nominatif, ce qui rend l'action traçable.

6.2 Comptes et escalade de privilèges

Le compte `julien` assure l'administration courante. Il **ne figure pas dans les sudoers**. L'obtention des privilèges passe par `su -` et le mot de passe du compte root :

```
julien@SRV-FOG-01:~$ su -
Mot de passe : ***
root@SRV-FOG-01:~#
```

> **Pourquoi su - plutôt que sudo, et ce que cela coûte**
>
> `sudo` et `su -` répondent à deux besoins différents. `sudo` prend tout son sens dès qu'il y a plusieurs administrateurs : chacun s'authentifie avec son propre mot de passe, la journalisation est nominative, et les droits peuvent être limités à certaines commandes. `su -` exige la connaissance du mot de passe root, qui devient un secret partagé — c'est son principal défaut, et il croît avec le nombre d'administrateurs.
>
> Sur ce serveur, l'administrateur est unique. Le secret partagé n'existe pas, et l'absence du compte dans les sudoers supprime un chemin d'élévation : une session `julien` compromise ne donne pas

> les privilèges sans un second mot de passe. Le choix est cohérent avec le contexte. **Il ne le resterait pas à deux administrateurs** : la bascule vers `sudo`, avec journalisation nominative, serait alors la bonne réponse.

6.3 Pare-feu

UFW est actif, en refus par défaut sur le trafic entrant. Dix-huit règles ont été ouvertes, dont la plupart sont imposées par le fonctionnement de FOG et détaillées au chapitre suivant.

Action — mise en place de la politique :

```
root@SRV-FOG-01:~# apt install -y ufw
root@SRV-FOG-01:~# ufw default deny incoming
root@SRV-FOG-01:~# ufw default allow outgoing
root@SRV-FOG-01:~# ufw allow from 192.168.1.0/24 to any port 17251 proto tcp
root@SRV-FOG-01:~# ufw allow from 192.168.1.0/24 to any port 80 proto tcp
root@SRV-FOG-01:~# ufw enable
```

Contrôle :

```
root@SRV-FOG-01:~# ufw status numbered
```

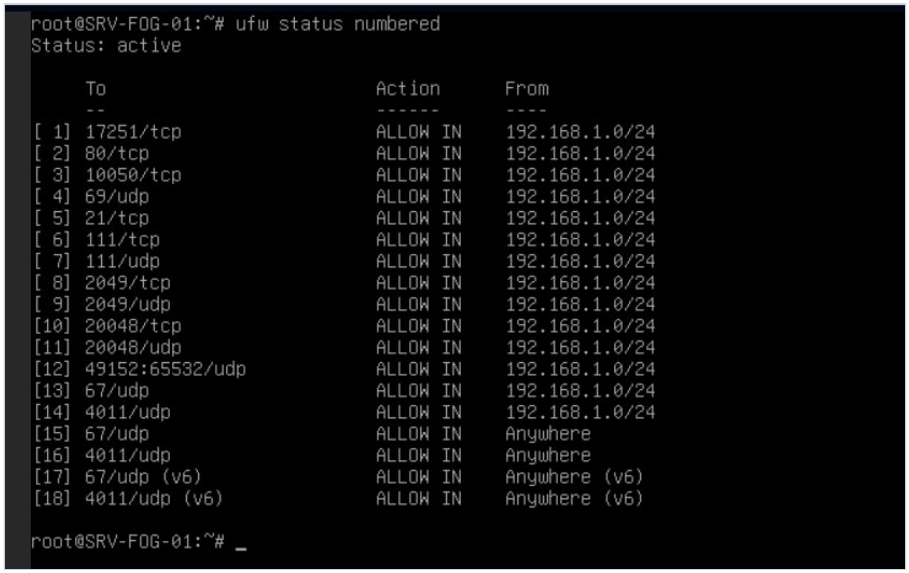

Figure 5 — État du pare-feu : dix-huit règles, toutes en entrée

Les règles d'administration sont les suivantes :

N°	Port	Source	Usage
1	17251/tcp	192.168.1.0/24	Accès SSH d'administration
2	80/tcp	192.168.1.0/24	Interface web et dialogue des clients en cours de déploiement
3	10050/tcp	192.168.1.0/24	Agent Zabbix — ouverture anticipée, voir ci-dessous

Toutes les règles sont restreintes au réseau local `192.168.1.0/24`, à deux exceptions près, traitées au chapitre 8. Le principe retenu est qu'un service ne doit être joignable que depuis le périmètre où il a une raison d'être appelé.

> **Une règle ouverte avant le service : un choix assumé**
>
> La règle n° 3 autorise le port 10050, port d'écoute de l'agent Zabbix. Le relevé des ports en écoute du chapitre 7 montre qu'**aucun processus n'écoute sur ce port** au moment du contrôle : l'agent n'est pas encore déployé sur SRV-FOG-01. La règle a été posée **par anticipation**, en cohérence avec les autres serveurs du lab, qui remontent leurs métriques vers SRV-ZABBIX-01 (192.168.1.204).
>
> Une règle sans service derrière n'ouvre rien : le pare-feu laisse passer le paquet, que le noyau rejette faute de processus à l'écoute. La démarche est cohérente avec la manière dont l'agent est déployé sur les postes du domaine — par stratégie de groupe, la règle réseau étant posée indépendamment de l'installation. L'intégration de ce serveur à la supervision est d'autant plus attendue que l'occupation de `/images`, dont le chapitre 5 a montré qu'elle n'est plus amortie par la réserve ext4, est précisément une métrique à surveiller.

7. Installation de FOG

7.1 Déroulement

FOG n'est pas distribué sous forme de paquet Debian. L'installation consiste à récupérer les sources et à exécuter le script d'installation, qui installe lui-même toutes les dépendances : serveur web, base de données, serveurs TFTP, NFS et FTP.

```
root@SRV-FOG-01:~# apt update && apt install -y git
root@SRV-FOG-01:~# cd /root
root@SRV-FOG-01:~# git clone --branch stable     https://github.com/FOGProject/fogproject.git
root@SRV-FOG-01:~# cd fogproject
```

Contrôle — relever la branche et le commit avant d'installer. C'est ce couple, et non le seul numéro de version affiché par l'interface, qui identifie sans ambiguïté le code installé :

```
root@SRV-FOG-01:~/fogproject# git branch --show-current
stable
root@SRV-FOG-01:~/fogproject# git rev-parse --short HEAD
9f3b5a896
root@SRV-FOG-01:~/fogproject# git log -1 --format='%h %ci'
9f3b5a896 2026-08-11 11:59:17 +0000
```

Ces valeurs sont reportées dans l'état de référence du chapitre 1. Sans elles, une reprise ne peut pas reconstruire le même serveur : le numéro de version affiché par l'interface ne suffit pas à identifier le code.

Action. L'installeur s'exécute ensuite :

```
root@SRV-FOG-01:~/fogproject# cd bin
root@SRV-FOG-01:~/fogproject/bin# ./installfog.sh
```

Le script pose une série de questions dont les réponses déterminent la configuration. Les valeurs retenues :

Question	Réponse	Motif
Type d'installation	Normal Server	Serveur complet ; le mode Storage Node ne concerne que les nœuds secondaires
Interface réseau	enp6s18	Interface unique de la machine
Serveur DHCP par FOG	**Non**	Le DHCP reste assuré par la box. Voir chapitre 8.
Routeur / DNS pour DHCP	Sans objet	Conséquence de la réponse précédente
Internationalisation	Oui	Interface disponible en français
Certificat HTTPS	**Non**	Incompatible avec le chargeur Secure Boot — voir chapitre 2
Nom d'hôte du serveur	192.168.1.207	Adresse littérale : les clients en préamorçage ne disposent pas encore d'une résolution de noms fiable

Les réponses sont conservées par l'installeur dans `/opt/fog/.fogsettings`. Ce fichier est relu lors des mises à jour : une réinstallation ultérieure reprend les mêmes choix sans les redemander.

7.2 Initialisation de la base de données

L'installation ne se déroule pas d'une seule traite. À un moment précis, **l'installeur s'interrompt et attend une action dans le navigateur** : le schéma de la base de données doit être créé, ou mis à jour s'il s'agit d'une montée de version.

Étape	Action
1	L'installeur affiche une invite demandant d'ouvrir l'interface web et attend
2	Ouvrir `http://192.168.1.207/fog/management` depuis un poste du réseau
3	Valider la création ou la mise à jour du schéma proposée par la page
4	Revenir au terminal et confirmer pour laisser l'installeur terminer

> **Une étape facile à manquer, et bloquante**
>
> C'est le seul moment de l'installation où le terminal ne suffit pas. Un opérateur qui laisse l'installeur en attente sans ouvrir l'interface obtient un serveur dont les services démarrent mais dont la base est vide : l'interface web renvoie une erreur, aucun hôte ne peut être enregistré, et rien dans les journaux système ne désigne clairement la cause. **Cette étape doit figurer dans toute procédure de reprise.** Le port 80 doit donc être joignable depuis un poste du réseau avant de lancer l'installeur — ce que garantit la règle de pare-feu n° 2 posée au chapitre 6.

7.3 Services en fonctionnement

```
root@SRV-FOG-01:~# systemctl is-active FOGImageReplicator FOGMulticastManager FOGScheduler nginx
mariadb; echo "---"; systemctl is-active dnsmasq; echo "---"; php -v | head -1
```

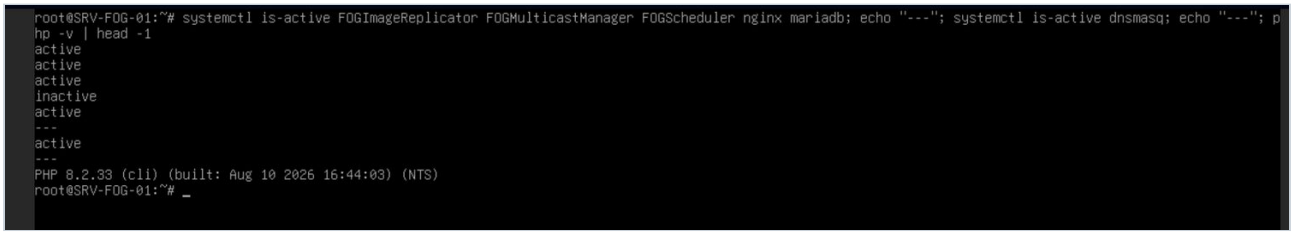

Figure 6 — État des services FOG, du service dnsmasq et version de PHP

Service	État	Rôle
FOGImageReplicator	active	Réplication entre nœuds de stockage. Actif sans objet ici : le serveur est le seul nœud.
FOGMulticastManager	active	Sessions de diffusion multiple. Indispensable au déploiement de flotte (document D).
FOGScheduler	active	Déclenchement des tâches planifiées

Service	État	Rôle
nginx	**inactive**	Non utilisé — voir l'encadré ci-dessous
mariadb	active	Base de données de FOG
dnsmasq	active	ProxyDHCP — chapitre 8
PHP	8.2.33 (cli)	Version imposée par l'installeur de FOG 1.5.10

> **Pourquoi nginx est inactif : le serveur web est Apache**
>
> L'installeur de FOG 1.5.10 déploie **Apache 2 associé à PHP-FPM**, et non Nginx. Le service `nginx` est présent sur le système mais n'a jamais été démarré ; son état `inactive` n'est pas une anomalie mais la conséquence du choix de l'installeur. Le relevé des ports en écoute ci-après le confirme : ce sont des processus `apache2` qui écoutent sur le port 80, et `php-fpm8.2` qui écoute en local sur le port 9000. Interroger `nginx` pour contrôler la santé du service web conduirait à un diagnostic faux — **la commande de contrôle correcte est `systemctl is-active apache2`**.

7.4 Ports en écoute

```
root@SRV-FOG-01:~# ss -tlnp
```

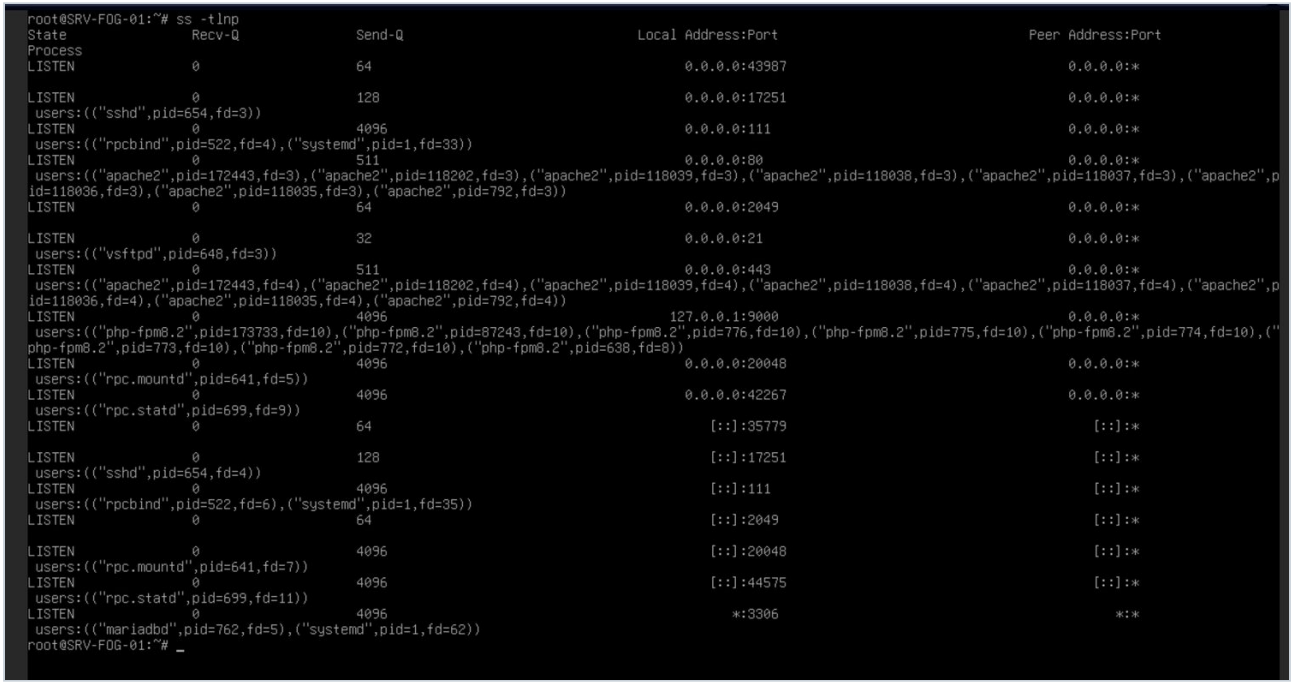

Figure 7 — Sockets TCP en écoute et processus associés

Port	Processus	Fonction
17251	sshd	Administration à distance
80	apache2	Interface web, menu d'amorçage, dialogue des clients FOS

Port	Processus	Fonction
443	apache2	Hôte virtuel TLS par défaut — non ouvert au pare-feu
9000 (127.0.0.1)	php-fpm8.2	Interpréteur PHP, joignable localement par Apache uniquement
3306	mariadb	Base de données
21	vsftpd	Transferts de fichiers internes à FOG
111, 2049, 20048	rpcbind, rpc.mountd, noyau	NFS : exposition de /images aux clients en déploiement

> **Pourquoi le port 443 écoute alors qu'il est fermé au pare-feu**
>
> Le paquet Apache de Debian active un hôte virtuel TLS par défaut, avec un certificat auto-signé, indépendamment de la configuration de FOG. Le service écoute donc, mais aucune règle UFW n'autorise le port 443 : la connexion est refusée avant d'atteindre Apache. Le résultat correspond à l'intention — le serveur est en HTTP, pour les raisons de compatibilité Secure Boot exposées au chapitre 2 — mais il l'obtient par le pare-feu plutôt que par la configuration du serveur web. Une désactivation explicite de l'hôte virtuel (`a2dissite default-ssl`) rendrait l'intention lisible dans la configuration elle-même, ce qui vaut mieux qu'un service qui écoute pour rien.

7.5 Règles de pare-feu imposées par FOG

Les règles 4 à 12 de la figure du chapitre 6 correspondent aux services que FOG met en œuvre pendant un déploiement. Chacune a une fonction précise :

N°	Port	Service	Moment où il intervient
4	69/udp	TFTP	Téléchargement du chargeur d'amorçage et du script iPXE, avant tout système
5	21/tcp	FTP	Transferts de fichiers internes de FOG
6, 7	111/tcp, 111/udp	rpcbind	Annuaire des services RPC, préalable à toute connexion NFS
8, 9	2049/tcp, 2049/udp	NFS	Lecture et écriture des images pendant capture et déploiement
10, 11	20048/tcp, 20048/udp	rpc.mountd	Montage des partages NFS par le client
12	49152:65532/udp	UDPCAST	Diffusion multiple vers plusieurs postes simultanément

> **Pourquoi NFS et pas HTTP pour transférer les images**
>
> Le système temporaire chargé sur le poste client (FOS) monte `/images` comme un système de fichiers local. Partclone y écrit alors directement, en flux, sans étape de téléchargement préalable ni espace tampon sur le poste — ce qui serait impossible puisque le disque du poste est précisément ce que l'on est en train de réécrire. NFS est le protocole qui permet cette écriture en flux ; HTTP servirait à télécharger un fichier, pas à monter un volume. C'est aussi pourquoi la plage de ports RPC doit être ouverte : sans `rpcbind` , le client ne peut même pas découvrir le service NFS.

Ces plages sont à revalider contre la configuration réelle

La plage `49152:65532/udp` couvre plus de seize mille ports. Elle correspond aux ports dynamiques d'UDPCAST, négociés au lancement de chaque session — mais **elle est nettement plus large que ce que documente le projet**. La documentation actuelle de FOG indique des plages sensiblement plus étroites : de l'ordre de `63100:63228/udp` pour UDPCAST, et `65000:65100/tcp` pour le FTP passif, que ce serveur n'ouvre pas.

Ces valeurs ne doivent pas être recopiées telles quelles. Elles décrivent une version qui n'est pas nécessairement la 1.5.10.2254 installée ici, et une règle de pare-feu posée sur une plage que le service n'utilise pas revient à ouvrir des ports pour rien tout en fermant ceux dont il a besoin.

Il ne s'agit pas de recopier ces valeurs à l'aveugle : elles varient selon la révision. La démarche est de **relever ce que cette installation utilise réellement**, puis de resserrer les règles en conséquence :

```
root@SRV-FOG-01:~# grep -E 'pasv_(min|max)_port|pasv_enable' /etc/vsftpd.conf
root@SRV-FOG-01:~# grep -riE 'udpcast|port' /opt/fog/.fogsettings
root@SRV-FOG-01:~# grep -riE 'MULTICAST|UDPCAST' /var/www/fog/lib/fog/ | head
```

La protection effective, en l'état, tient à la restriction au réseau local : aucune de ces plages n'est joignable depuis l'extérieur. Le resserrement reste une amélioration à mener, pas une correction urgente.

Le FTP passif et le suivi de connexion TFTP

Deux points méritent d'être vérifiés sur ce serveur. Le premier : aucune plage de **ports FTP passifs** n'est ouverte. Ce n'est pas gênant tant que le FTP n'est utilisé qu'en local par FOG lui-même, ce qui est le cas avec un nœud de stockage unique ; l'ajout d'un nœud secondaire rendrait l'ouverture nécessaire.

Le second : TFTP négocie ses transferts sur un port dynamique après la requête initiale sur le port 69. Avec un pare-feu à états, le module `nf_conntrack_tftp` est ce qui permet de suivre cette négociation sans ouvrir une large plage. Sa présence se vérifie par `lsmod | grep nf_conntrack_tftp`, et son chargement se rend persistant en l'inscrivant dans `/etc/modules`.

7.6 Interface d'administration et version installée

L'interface est accessible à l'adresse `http://192.168.1.207/fog/management`. La page *FOG Configuration* donne la version en service et celles publiées par le projet.

Figure 8 — Version de FOG en service et nœud de stockage DefaultMember

Élément	Valeur
Version en service	1.5.10.2254
Dernière version stable publiée	1.5.10.2253
Dernière version de la branche de développement	1.5.10.2433
Nœud de stockage	DefaultMember

L'alerte « version non à jour » ne doit pas déclencher de mise à jour

L'interface signale en rouge que le serveur n'exécute pas la version la plus récente. La comparaison des numéros dit l'inverse : la version en service, **2254, est supérieure à la stable annoncée, 2253**.

Ce que cet écart ne prouve pas. Il serait tentant d'en conclure que le serveur est installé depuis une branche de développement. Ce n'est pas établi. Dans le dépôt du projet, la branche par défaut est `stable` : elle porte la dernière version corrective, et c'est d'elle que la plupart des installations doivent partir. Un `git clone` sans argument récupère donc la branche recommandée, pas une branche de travail. Le dernier segment du numéro de version correspond aux correctifs, publiés de façon automatisée et fréquente ; le contrôleur de version compare à une valeur publiée qui peut être plus ancienne que le dépôt.

Ce qui a tranché la question. Le relevé du chapitre 7.1 donne la réponse : branche `stable`, commit `9f3b5a896` daté du 11 août 2026, soit neuf jours avant la capture du master. Le serveur exécute donc du code publié sur la branche recommandée, et non une révision de développement. **L'alerte de l'interface ne traduit qu'un décalage du contrôleur de version et ne doit déclencher aucune mise à jour.**

La démarche vaut au-delà de ce cas : c'est le couple branche et commit qui identifie le code installé, jamais le seul numéro affiché.

Changer de branche n'est pas une opération anodine

Les branches de FOG n'introduisent pas seulement du code : elles font évoluer le schéma de la base de données. Passer de la branche stable vers `dev-branch` se fait sans difficulté, mais le **retour en arrière est autrement plus délicat** — le schéma déjà migré peut ne plus correspondre à ce qu'attend la branche stable, et il faut alors attendre la publication suivante.

> Sur un serveur de déploiement en service, un changement de branche se traite donc comme une opération planifiée : sauvegarde préalable de la machine, fenêtre hors campagne de déploiement, et vérification que le retour arrière est possible avant de l'engager.

8. Service ProxyDHCP

C'est la pièce qui permet à l'ensemble de fonctionner sans modifier le réseau existant. Le principe : la box continue d'attribuer les adresses, et dnsmasq complète sa réponse en indiquant au poste où trouver son fichier d'amorçage.

> **Pourquoi ne pas déplacer le DHCP vers le contrôleur de domaine**
>
> La solution classique consiste à transférer le service DHCP vers SRV-AD-01 et à y déclarer les options 66 et 67, qui désignent le serveur et le fichier d'amorçage. Elle a été écartée pour une raison d'exploitation : le lab partage son réseau avec un usage domestique, et couper le DHCP de la box interrompt la connectivité de tous les équipements de la maison. Le mode ProxyDHCP obtient le même résultat fonctionnel **sans aucune modification du réseau existant** — ce qui, au-delà du contexte de ce lab, est aussi la réponse à donner en clientèle lorsqu'on ne maîtrise pas l'infrastructure réseau du site.

8.1 Configuration

Action — installation, contrôle de syntaxe et activation. `--test` valide le fichier avant tout redémarrage : un service d'amorçage arrêté sur une erreur de syntaxe rend le PXE muet.

```
root@SRV-FOG-01:~# apt install -y dnsmasq
root@SRV-FOG-01:~# dnsmasq --test
root@SRV-FOG-01:~# systemctl enable --now dnsmasq
root@SRV-FOG-01:~# systemctl restart dnsmasq
root@SRV-FOG-01:~# systemctl is-active dnsmasq
```

Contrôle du contenu :

```
root@SRV-FOG-01:~# cat /etc/dnsmasq.d/*.conf
```

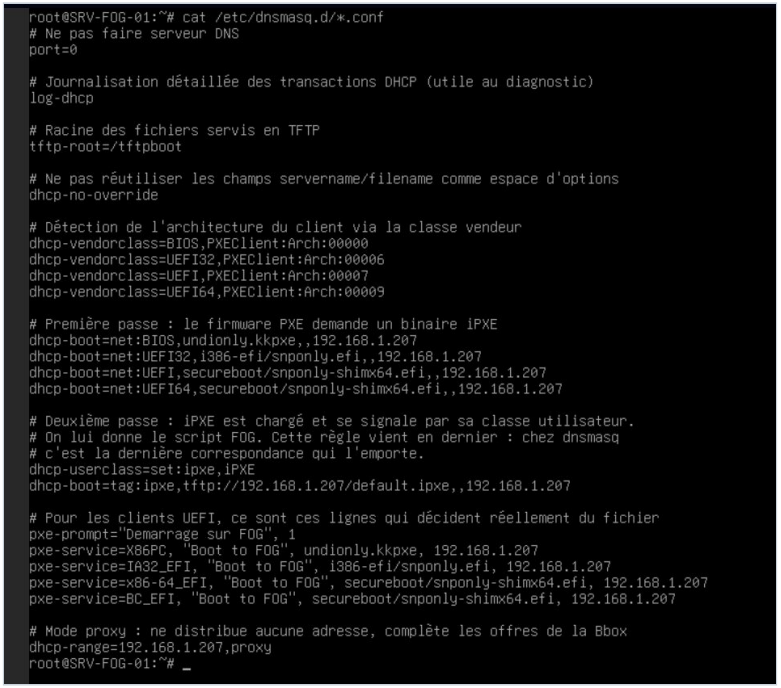

Figure 9 — Configuration ProxyDHCP commentée, dans /etc/dnsmasq.d/

Le contrôle des seules directives actives, commentaires et lignes vides exclus, confirme qu'aucune ligne n'a été neutralisée par inadvertance :

```
root@SRV-FOG-01:~# grep -vE '^#|^$' /etc/dnsmasq.d/*.conf
```

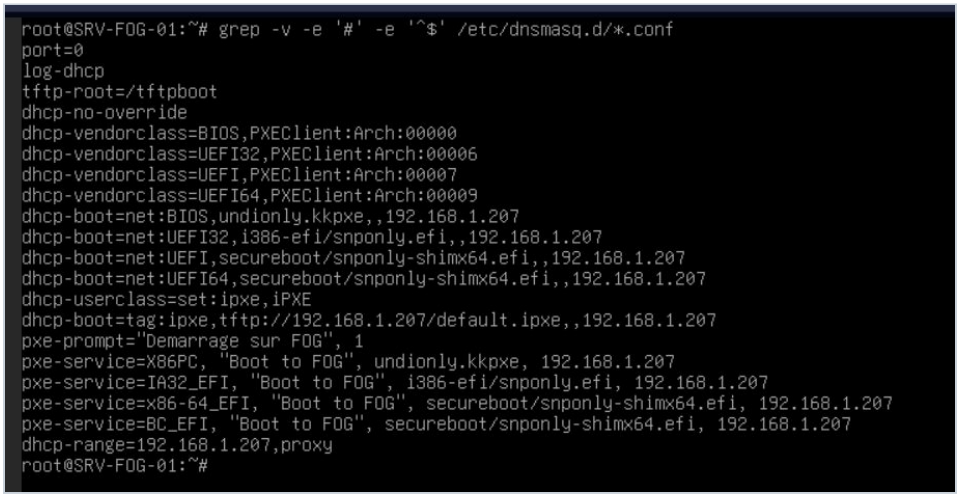

Figure 10 — Directives effectivement actives

Le fichier est reproduit ci-dessous. C'est cette forme, et non la capture, qui permet de reconstruire le service à l'identique.

```
# Ne pas faire serveur DNS
port=0

# Journalisation detaillee des transactions DHCP (utile au diagnostic)
log-dhcp

# Racine des fichiers servis en TFTP
tftp-root=/tftpboot

# Ne pas reutiliser les champs servername/filename comme espace d'options
dhcp-no-override

# Detection de l'architecture du client via la classe vendeur
dhcp-vendorclass=BIOS,PXEClient:Arch:00000
dhcp-vendorclass=UEFI32,PXEClient:Arch:00006
dhcp-vendorclass=UEFI,PXEClient:Arch:00007
dhcp-vendorclass=UEFI64,PXEClient:Arch:00009

# Premiere passe : le firmware PXE demande un binaire iPXE
dhcp-boot=net:BIOS,undionly.kkpxe,,192.168.1.207
dhcp-boot=net:UEFI32,i386-efi/snponly.efi,,192.168.1.207
dhcp-boot=net:UEFI,secureboot/snponly-shimx64.efi,,192.168.1.207
dhcp-boot=net:UEFI64,secureboot/snponly-shimx64.efi,,192.168.1.207

# Deuxieme passe : iPXE est charge et se signale par sa classe utilisateur.
# On lui donne le script FOG. Cette regle vient en dernier : chez dnsmasq
# c'est la derniere correspondance qui l'emporte.
dhcp-userclass=set:ipxe,iPXE
dhcp-boot=tag:ipxe,tftp://192.168.1.207/default.ipxe,,192.168.1.207

# Pour les clients UEFI, ce sont ces lignes qui decident reellement du fichier
pxe-prompt="Demarrage sur FOG", 1
pxe-service=X86PC, "Boot to FOG", undionly.kkpxe, 192.168.1.207
pxe-service=IA32_EFI, "Boot to FOG", i386-efi/snponly.efi, 192.168.1.207
pxe-service=x86-64_EFI, "Boot to FOG", secureboot/snponly-shimx64.efi, 192.168.1.207
```

```
pxe-service=BC_EFI, "Boot to FOG", secureboot/snponly-shimx64.efi, 192.168.1.207

# Mode proxy : ne distribue aucune adresse, complete les offres de la Bbox
dhcp-range=192.168.1.207,proxy
```

8.2 Lecture des directives

Directive	Effet
`port=0`	Désactive le serveur DNS de dnsmasq. Seule la fonction DHCP est utilisée ; laisser le DNS actif entrerait en conflit avec SRV-AD-01.
`log-dhcp`	Journalise chaque transaction. Indispensable au diagnostic d'un poste qui n'amorce pas.
`tftp-root=/tftpboot`	Désigne la racine des fichiers servis en TFTP, celle que FOG alimente.
`dhcp-no-override`	Interdit la réutilisation des champs `servername` et `filename` comme espace d'options. Certains micrologiciels UEFI lisent mal les réponses qui en usent.
`dhcp-vendorclass=…`	Quatre lignes qui associent une étiquette à chaque architecture cliente déclarée : BIOS, UEFI 32 bits, UEFI, UEFI 64 bits.
`dhcp-boot=net:…`	Quatre lignes qui, à chaque étiquette, associent un fichier d'amorçage et l'adresse du serveur TFTP.
`dhcp-userclass=set:ipxe,iPXE`	Pose l'étiquette `ipxe` lorsque la requête provient d'iPXE lui-même et non du micrologiciel.
`dhcp-boot=tag:ipxe,tftp://…/default.ipxe`	Sert le script FOG aux requêtes ainsi étiquetées.
`pxe-prompt`, `pxe-service`	Menu et libellés présentés par le micrologiciel. Pour les clients UEFI, ce sont ces lignes qui déterminent réellement le fichier chargé.
`dhcp-range=192.168.1.207,proxy`	Bascule dnsmasq en mode proxy : il ne distribue aucune adresse et se contente de compléter les offres de la box.

> **Pourquoi une configuration en deux passes**
>
> Les deux directives `dhcp-boot` ne s'adressent pas au même interlocuteur. La première série répond au micrologiciel UEFI du poste et lui sert un binaire iPXE, choisi selon l'architecture annoncée. La seconde répond à iPXE, une fois qu'il a démarré et qu'il refait sa propre requête : il se signale alors par sa classe utilisateur, et reçoit cette fois `default.ipxe`, le script qui affiche le menu FOG.
>
> L'ordre compte : chez dnsmasq, c'est la **dernière correspondance** qui l'emporte, d'où le placement de la règle étiquetée `ipxe` après les autres. Sans cette seconde passe, iPXE relancerait une requête à laquelle personne ne répondrait de manière utile, et retomberait sur la box — qui ne connaît aucun fichier d'amorçage. Le symptôme observé est alors un poste qui charge iPXE, puis abandonne.

8.3 Ouverture des ports DHCP

Les règles 13 à 18 du pare-feu concernent les ports utilisés par dnsmasq. Elles se lisent par paires :

N°	Port	Source	Effet réel
13, 14	67/udp, 4011/udp	192.168.1.0/24	**Ne correspondent jamais** — voir l'encadré
15, 16	67/udp, 4011/udp	Anywhere	Ce sont ces règles qui laissent passer le trafic
17, 18	67/udp, 4011/udp	Anywhere (v6)	Équivalents IPv6, sans usage ici

Une règle filtrée par sous-réseau ne peut pas voir un DHCP Discover

C'est le point le plus contre-intuitif de cette installation, et il a coûté un diagnostic. Un poste qui démarre n'a **pas encore d'adresse IP** : sa requête DHCP Discover part de l'adresse source `0.0.0.0` vers l'adresse de diffusion `255.255.255.255`. Une règle UFW qui filtre sur la source `192.168.1.0/24` ne peut donc jamais correspondre — l'adresse `0.0.0.0` n'appartient pas à ce sous-réseau.

Le symptôme est trompeur : le pare-feu paraît correctement configuré, le service dnsmasq est actif, et pourtant aucun poste n'amorce. **Les ports 67/udp et 4011/udp doivent être ouverts à `Anywhere`**. Les règles 13 et 14 subsistent du premier essai ; elles sont sans effet et peuvent être supprimées, à condition de retirer les bonnes : `ufw status numbered` puis `ufw delete` en commençant par le numéro le plus élevé, la suppression d'une règle renumérotant toutes les suivantes.

Pourquoi le port 4011 en plus du 67

Le port 67 reçoit la requête initiale, diffusée à l'ensemble du réseau. Certains micrologiciels PXE, une fois qu'ils ont identifié un serveur d'amorçage, lui adressent une requête complémentaire en unicast sur le port 4011 pour obtenir le nom exact du fichier à charger. Un serveur ProxyDHCP doit donc écouter sur les deux, faute de quoi une partie seulement du parc amorcera — le comportement dépendant du fabricant de la carte réseau.

9. Validation et exploitation

9.1 Contrôles de mise en service

Les huit contrôles suivants établissent qu'un serveur FOG est opérationnel. Chacun renvoie à la commande qui le vérifie.

Point	Commande	Résultat attendu
Système	`cat /etc/os-release`	Debian GNU/Linux 12 (bookworm)
Adressage	`ip -4 addr show`	192.168.1.207/24 sur enp6s18
Volume des images	`df -h /images`	Monté, occupation inférieure à 80 %
Montage permanent	`grep images /etc/fstab`	Ligne UUID présente
Serveur web	`systemctl is-active apache2`	active
Services FOG	`systemctl is-active FOGMulticastManager FOGScheduler mariadb`	active pour les trois
ProxyDHCP	`systemctl is-active dnsmasq`	active
Pare-feu	`ufw status numbered`	Status: active, 67/udp et 4011/udp ouverts à Anywhere
Interface web	http://192.168.1.207/fog/management	Accessible, version 1.5.10.2254, nœud DefaultMember présent

9.2 Contrôle de bout en bout

Les contrôles précédents établissent que les services fonctionnent. Ils n'établissent pas que la chaîne d'amorçage fonctionne : seul le démarrage effectif d'un poste le prouve. Le test de bout en bout consiste à démarrer une machine de test en amorçage réseau et à observer l'apparition du menu FOG. Il est décrit dans le document D.

En cas d'échec, la journalisation de dnsmasq donne la réponse en quelques lignes :

```
root@SRV-FOG-01:~# journalctl -u dnsmasq -f
```

La lecture attendue est la suivante : une requête reçue depuis `0.0.0.0`, une réponse proxy annonçant `snponly-shimx64.efi`, puis une seconde requête portant la classe utilisateur `iPXE` et une réponse annonçant `default.ipxe`. L'absence de la première ligne oriente vers le pare-feu ; l'absence de la seconde vers la configuration en deux passes.

9.3 Test de recette

Les contrôles précédents portent sur des états. Le test de recette porte sur le **service rendu** : il n'est réussi que si un poste vierge peut être installé de bout en bout après un redémarrage complet du serveur. C'est le seul contrôle qui engage réellement la mise en service.

Étape	Action	Résultat attendu
1	`reboot` du serveur	Tous les services remontent seuls : le contrôle 9.1 repasse sans intervention
2	Démarrer un poste de test en amorçage réseau	Le menu FOG s'affiche
3	Laisser le poste charger le système temporaire	FOS démarre complètement — l'affichage du menu ne suffit pas à le prouver
4	Lancer un déploiement ou une capture de test	Le transfert se déroule et se termine sans erreur
5	Relire les journaux du serveur	Aucune erreur dans dnsmasq, Apache et les services FOG

```
root@SRV-FOG-01:~# journalctl -u dnsmasq --since "1 hour ago" | grep -i error
root@SRV-FOG-01:~# tail -50 /var/log/apache2/error.log
root@SRV-FOG-01:~# tail -50 /opt/fog/log/fogreplicator.log
```

> **L'étape 3 est celle que l'on saute, et c'est la mauvaise**
>
> Voir apparaître le menu FOG prouve seulement que la chaîne DHCP, TFTP et iPXE fonctionne. Le chargement de FOS met en jeu d'autres éléments : le service web qui répond au script iPXE, le noyau et le système de fichiers initial servis par HTTP, puis le montage NFS de `/images`. **Un serveur peut afficher un menu parfaitement et ne déployer aucun poste.**
>
> C'est exactement ce qu'a montré l'essai sous Secure Boot actif décrit au chapitre 2 : menu affiché sans anomalie, puis noyau refusé pour violation de la politique de sécurité. Un contrôle arrêté à l'apparition du menu aurait conclu à un fonctionnement normal.

9.4 Tâches d'exploitation

Fréquence	Tâche	Motif
Hebdomadaire	Contrôle de l'occupation de /images	La réserve ext4 ayant été supprimée, la saturation n'est plus amortie
Hebdomadaire	Suppression des images obsolètes	Chaque capture crée une image ; les versions antérieures restent
Mensuelle	Mise à jour du système Debian	Correctifs de sécurité
Ponctuelle	Mise à jour de FOG	À ne pas engager pendant une campagne de déploiement, ni sans avoir vérifié la compatibilité avec la version de Debian installée

Fréquence	Tâche	Motif
Ponctuelle	Vérification de la sauvegarde du disque système	La base MariaDB porte toute la configuration ; le disque des images est exclu de la sauvegarde

> **Trois points à connaître avant toute intervention**
>
> Aucun ne compromet le fonctionnement, mais chacun induit en erreur qui les découvre sans explication.
>
> **Supervision.** Le port 10050 est ouvert alors qu'aucun agent Zabbix n'écoute : ouverture anticipée, en cohérence avec le reste du lab. Le serveur n'est pas encore supervisé — ce qui reste à faire, l'occupation de `/images` étant la métrique la plus utile à remonter.
>
> **Règles de pare-feu sans effet.** Les règles 13 et 14 filtrent les ports DHCP sur le sous-réseau source et ne peuvent jamais correspondre (chapitre 8). Elles sont inoffensives mais rendent la lecture du jeu de règles ambiguë, et peuvent être supprimées.
>
> **Alerte de version.** L'interface signale une version non à jour alors que le serveur est en avance d'une révision sur la stable (chapitre 7.5). Ne pas déclencher de mise à jour sur cette base.

9.5 Références

Les sources consultées pour l'ensemble du dossier sont rassemblées au document A, chapitre « Références ». Celles qui font autorité sur le contenu du présent volume sont les suivantes.

Source	Points concernés
docs.fogproject.org — installation du serveur	Branches du dépôt, déroulement de l'installeur, initialisation de la base de données
github.com/FOGProject/fogproject — fichier README	Rôle respectif des branches `stable`, `master` et `dev-branch` ; format du numéro de version
docs.fogproject.org — exigences réseau et pare-feu	Plages de ports à comparer à la configuration observée (voir 7.5)
docs.fogproject.org — transport et PKI	Évolution du traitement HTTPS et Secure Boot en version 1.6
Suivi des anomalies du projet, n° 797	Échec de l'installeur sur Debian 13
Pages de manuel dnsmasq, ufw, tune2fs, findmnt	Directives de configuration et options employées

9.6 Éléments à conserver hors du serveur

La reconstruction du serveur suppose de disposer des éléments suivants, qui ne doivent pas résider uniquement sur la machine :

- Le présent document, qui porte les valeurs de configuration
- Le contenu de `/etc/dnsmasq.d/`

- Le contenu de `/opt/fog/.fogsettings`
- La branche et le commit du dépôt, relevés au chapitre 7.1
- Les sauvegardes du disque système vers Proxmox Backup Server

Les **secrets** forment un cas à part et ne relèvent pas du même traitement : mot de passe root, mot de passe de l'interface FOG, identifiants du compte de jonction au domaine, et le cas échéant les clés Secure Boot ou MOK. Ils ne doivent figurer ni dans ce document, ni dans un fichier déposé à côté des sauvegardes, mais dans un **gestionnaire de secrets** — l'exigence n'est pas qu'ils soient conservés hors du serveur, mais qu'ils soient conservés chiffrés, avec un accès tracé et révocable.

Les images elles-mêmes n'en font pas partie : elles se recapturent depuis les masters, dont la fabrication est décrite au document C.