PROJET FOG — DOCUMENT A

Présentation du projet

Contexte, choix techniques et architecture — Lab TSSR / Infrastructure Proxmox

Auteur	Julien BARBET

Version	1.0
Date	28 août 2026
Objet	Déploiement d'images sur poste de travail
Solution	FOG Project 1.5.10.2254
Serveur	SRV-FOG-01 — 192.168.1.207
Hyperviseur	Proxmox VE 9.2.2
Domaine	lab.julien.local
Contexte	Lab personnel de formation

1. Contexte et objectifs

1.1 Le problème posé

Installer un poste de travail à la main occupe un technicien pendant environ une heure : installation du système, paramètres régionaux, jonction au domaine, logiciels, réglages. Sur un parc, cette méthode pose trois problèmes distincts.

Le premier est le **temps**, qui croît proportionnellement au nombre de postes. Le deuxième est la **dérive de configuration** : deux postes installés par deux personnes, ou par la même personne à deux moments, finissent par différer sur des détails qui se paient plus tard en incidents difficiles à reproduire. Le troisième est l'**absence de trace** : une installation manuelle ne se documente pas d'elle-même, et l'état d'un poste ne peut être établi qu'en l'inspectant.

1.2 La réponse retenue

Le déploiement d'images inverse la logique. Une installation de référence est construite une fois, contrôlée, puis dupliquée telle quelle sur le parc. Le temps d'intervention par poste tombe à quelques minutes, la conformité devient une propriété de l'image plutôt qu'une discipline individuelle, et l'état attendu d'un poste est connu par construction.

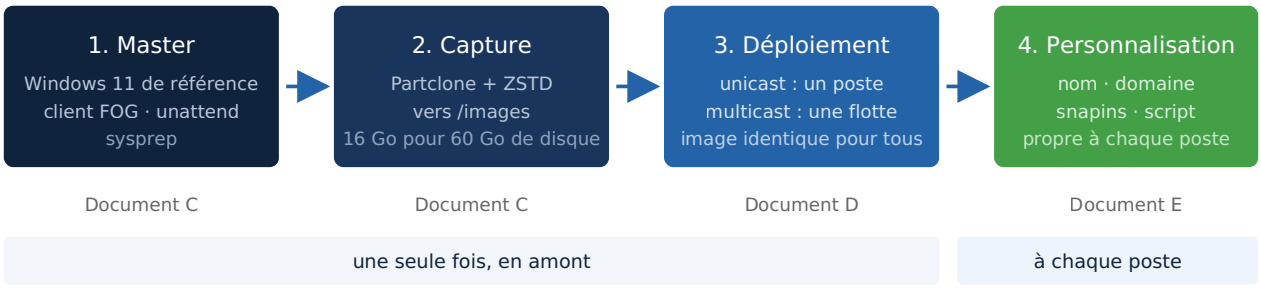

Figure 1 — Cycle de vie d'un poste, de la construction du master à sa mise en service

Le schéma fait apparaître la séparation qui structure tout le projet : les deux premières étapes n'ont lieu qu'une fois et produisent un objet réutilisable ; les deux dernières se rejouent à chaque poste. C'est cette séparation qui rend le coût marginal d'un poste supplémentaire faible.

1.3 Objectifs

Objectif	Traduction concrète
Réduire le temps d'installation	Passer d'environ une heure par poste à quelques minutes de transfert, sans intervention pendant le déroulement
Garantir l'uniformité	Tous les postes issus d'une même image sont identiques : mêmes versions, mêmes réglages, mêmes logiciels
Automatiser l'intégration au domaine	Nommage, jonction à `lab.julien.local` et placement dans la bonne unité d'organisation, sans saisie

Objectif	Traduction concrète
Permettre le déploiement simultané	Traiter une salle entière sans que le débit par poste s'effondre
Ne pas perturber le réseau existant	Aucune modification du service DHCP en place
Rester intervenable après coup	Pouvoir pousser un logiciel ou un correctif sur un parc en production, sans réinstaller

1.4 Périmètre du projet

Dans le périmètre	Hors périmètre
Serveur de déploiement et son système	Installation du domaine Active Directory
Fabrication et capture de l'image de référence	Stratégies de groupe (traitées à part)
Déploiement unitaire et de flotte	Supervision de l'infrastructure
Jonction automatique au domaine	Sauvegarde des données utilisateur
Distribution logicielle par snapins	Gestion du parc physique
Configuration post-déploiement des postes	Migration des profils depuis un ancien poste

1.5 Organisation de la documentation

Le projet est documenté en cinq volumes, découpés par lecteur et non par ordre chronologique. Le présent document donne le contexte et les choix ; les quatre autres sont des modes opératoires.

Doc	Objet	Lecteur
A	**Présentation du projet — ce document**	**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

2. Choix de la solution

2.1 Les options examinées

Trois approches permettent d'industrialiser l'installation de postes Windows. Elles ne reposent pas sur le même principe, ce qui explique qu'elles ne se comparent pas trait pour trait.

Solution	Principe	Éditeur
WDS + MDT	Réinstalle Windows depuis une image de fichiers (WIM), pilotée par une séquence de tâches	Microsoft
FOG Project	Copie et restaure les partitions d'un disque de référence	Communauté — GPL v3
Clonezilla SE	Copie et restaure les partitions, sans base de données ni gestion d'hôtes	Communauté — GPL v2

Clonezilla a été écarté rapidement. Il clone efficacement, mais ne gère ni inventaire d'hôtes, ni interface web, ni jonction au domaine, ni distribution logicielle : il faudrait reconstruire autour de lui ce que FOG fournit déjà. La comparaison utile est donc entre FOG et le couple Microsoft.

2.2 Comparaison FOG / WDS-MDT

Critère	FOG Project	WDS + MDT
Nature de l'image	Copie des partitions du disque. L'image reproduit le disque tel qu'il a été fabriqué.	Image de fichiers rejouée par une séquence qui réinstalle Windows.
Durée par poste	Transfert de l'image, quelques minutes	Réinstallation complète, sensiblement plus longue
Système hôte	Debian, aucune licence	Windows Server, licence requise
Intégration DHCP	Mode ProxyDHCP : le DHCP existant n'est pas touché	Exige la maîtrise du DHCP, ou les options 66/67, ou un relais
Systèmes pris en charge	Indifférent : tout ce qui tient sur un disque	Windows uniquement
Diffusion multiple	Multicast natif (UDPCAST), piloté par groupes d'hôtes	Multicast natif
Personnalisation après clonage	Client FOG embarqué dans l'image : renommage, jonction, snapins	Séquence de tâches, plus riche et plus complexe
Distribution logicielle continue	Snapins envoyés à un poste déjà en production	Relève d'un autre outil (SCCM, Intune)
Injection de pilotes	Absente : l'image doit convenir au matériel cible	Gestion structurée par modèle de machine

Critère	FOG Project	WDS + MDT
Pérennité de l'outil	Projet communautaire actif	MDT n'est plus développé ; Microsoft oriente vers Intune et Autopilot

2.3 La décision

FOG a été retenu sur trois critères, par ordre de poids.

L'absence de contrainte sur le DHCP. C'est le critère décisif. Le réseau est desservi par une box opérateur dont le service DHCP ne peut pas être remplacé sans interrompre la connectivité de tout ce qui en dépend. Le mode ProxyDHCP permet à FOG de répondre aux requêtes d'amorçage *en complément* de la box, sans lui retirer son rôle.

Le coût. Nul en licence, faible en ressources : deux cœurs, 4 Go de mémoire et 232 Go de disque suffisent au serveur, qui cohabite avec sept autres machines virtuelles sur un hôte de 32 Go.

La lisibilité du modèle. Copier un disque puis le réécrire est un mécanisme que l'on peut expliquer entièrement et diagnostiquer de bout en bout. Une séquence de tâches MDT introduit une couche d'abstraction supplémentaire, plus puissante mais plus coûteuse à maîtriser.

> **Ce que ce choix suppose du contexte**
>
> Les trois critères ci-dessus sont des critères *de situation*, pas de supériorité technique. Sur un parc hétérogène de plusieurs centaines de postes, avec une équipe formée et une infrastructure Microsoft déjà en place, la gestion structurée des pilotes et le séquencement de MDT — ou plus vraisemblablement aujourd'hui Intune et Autopilot — pèseraient davantage que l'indépendance vis-à-vis du DHCP. La décision doit donc être réexaminée si l'un des trois critères change.

> **La limite assumée : les pilotes**
>
> C'est le point faible de l'approche par image disque, et il doit être connu avant toute transposition. Une image fabriquée sur un modèle de machine ne contient que les pilotes de ce modèle. Sur un parc hétérogène, il faut soit une image par modèle — avec la charge de maintenance correspondante — soit une injection de pilotes en post-déploiement. MDT traite nativement ce problème ; FOG le laisse à l'administrateur. Dans ce projet, 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.**

3. Architecture

3.1 Vue d'ensemble

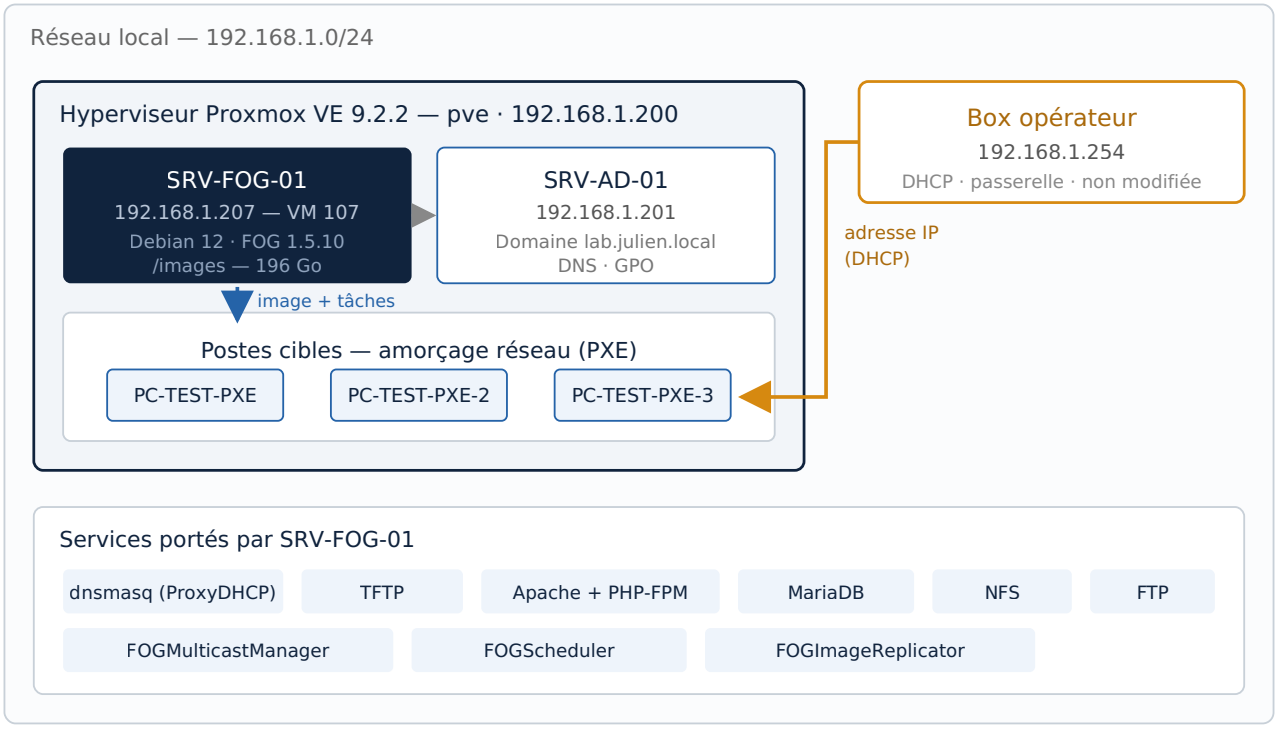

Figure 2 — Architecture de la solution et répartition des rôles

Trois éléments seulement participent au déploiement : la box, qui attribue les adresses ; le serveur FOG, qui indique aux postes quoi charger et leur sert l'image ; le contrôleur de domaine, qui reçoit les postes une fois installés. Aucun équipement réseau n'a été ajouté ni reconfiguré.

3.2 Composants

Élément	Rôle	Adresse
Box opérateur	DHCP et passerelle — inchangée	192.168.1.254
pve	Hyperviseur Proxmox VE 9.2.2	192.168.1.200
SRV-FOG-01	Serveur de déploiement — VM 107	192.168.1.207
SRV-AD-01	Contrôleur de domaine, DNS, stratégies de groupe	192.168.1.201
PC-MASTER-W11	Poste de référence — VM 211	adresse dynamique
PC-TEST-PXE 1 à 3	Postes cibles de validation	adresse dynamique

3.3 Les services du serveur

Un serveur FOG n'est pas un service unique mais un assemblage de services standards, orchestrés par une application web. Cette composition explique la longueur de la liste de ports à ouvrir : chacun correspond à une étape précise du déploiement.

Service	Fonction	Moment
dnsmasq	ProxyDHCP : désigne le fichier d'amorçage	Au démarrage du poste
TFTP	Sert le chargeur et le script iPXE	Avant tout système
Apache + PHP-FPM	Interface d'administration, menu, dialogue avec le client	Administration et déploiement
MariaDB	Hôtes, images, groupes, tâches, snapins	En permanence
NFS	Expose `/images` au système temporaire du poste	Capture et déploiement
FTP	Transferts de fichiers internes	Gestion des images
FOGMulticastManager	Sessions de diffusion multiple	Déploiement de flotte
FOGScheduler	Tâches planifiées	Déploiements différés

> **Pourquoi NFS plutôt qu'un téléchargement**
>
> Le système temporaire chargé sur le poste monte `/images` comme un volume local, et Partclone y lit ou y écrit en flux. Télécharger l'image d'abord serait impossible : le disque du poste est précisément ce que l'on est en train de réécrire, il n'existe aucun espace tampon disponible. C'est ce détail qui impose l'ouverture des ports RPC en plus du port NFS lui-même.

3.4 Dimensionnement

Ressource	Alloué	Justification
Processeur	2 cœurs	La charge est dominée par les entrées-sorties, pas par le calcul
Mémoire	4 Go (ballon à 2 Go)	MariaDB, Apache et PHP-FPM ; l'hyperviseur récupère la moitié au repos
Disque système	32 Go	Système, application et base de données
Disque des images	200 Go, exclu des sauvegardes	Environ 16 Go par image ; 25 Go occupés à ce jour

> **Ce que l'exclusion des images implique en cas de sinistre**
>
> Le disque des images est volontairement exclu de la sauvegarde : son contenu est volumineux et reconstructible depuis les masters. La conséquence doit figurer au plan de reprise. Restaurer le serveur rend une machine fonctionnelle, configurée, connaissant tous ses hôtes — **mais dont le**

> **répertoire des images est vide**. Une étape de recapture est nécessaire avant que le service soit réellement rendu. Sans cette mention, la restauration paraît réussie alors qu'aucun déploiement n'est possible.

4. Le flux d'amorçage réseau

C'est le mécanisme central du projet, et le seul dont le fonctionnement ne va pas de soi : deux serveurs DHCP répondent à la même requête sans se concurrencer, et le poste interroge deux fois le réseau avant de charger quoi que ce soit.

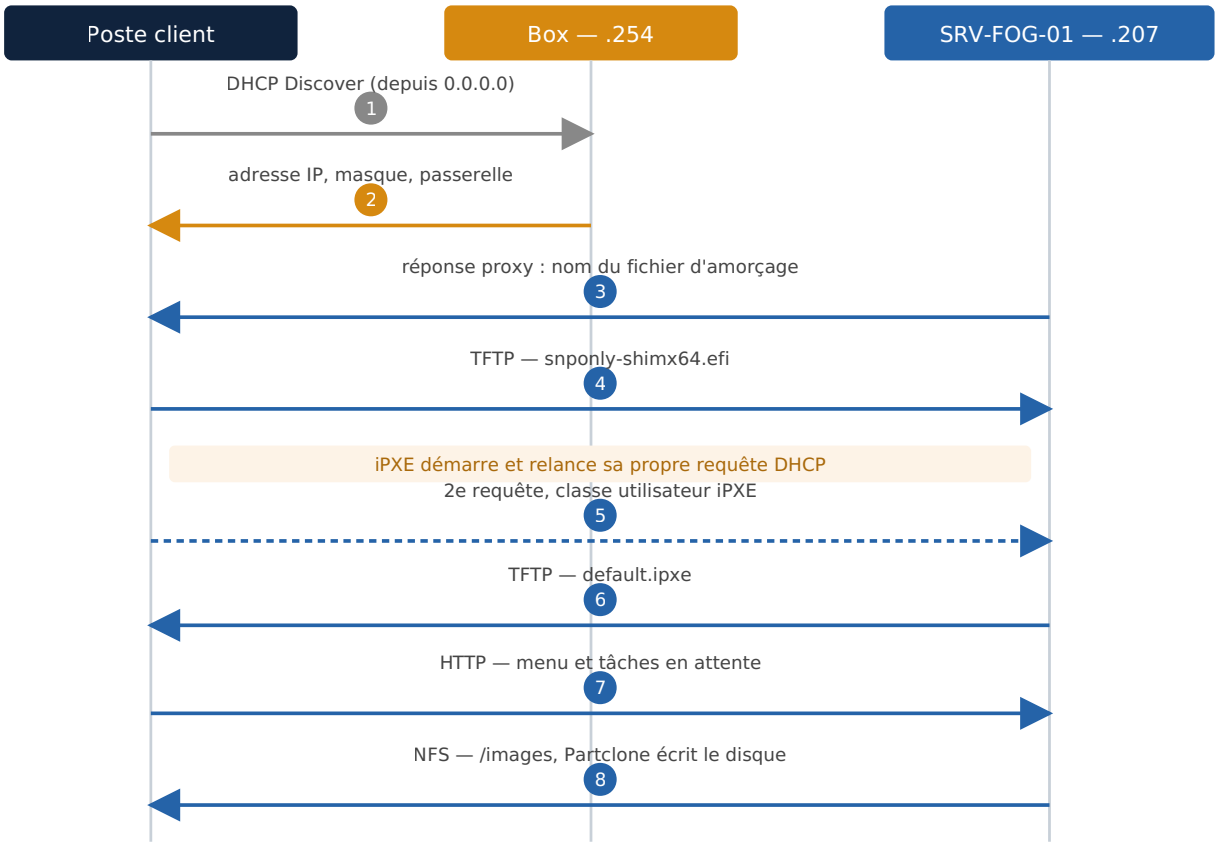

Figure 3 — Séquence d'amorçage réseau, du démarrage du poste à l'écriture du disque

4.1 Lecture de la séquence

Étape	Ce qui se passe
1	Le poste, qui n'a pas encore d'adresse, diffuse une requête DHCP depuis `0.0.0.0` en déclarant son architecture
2	La box attribue l'adresse, le masque et la passerelle. Elle ne connaît aucun fichier d'amorçage.
3	Le serveur FOG répond en mode proxy : pas d'adresse, mais le nom du serveur d'amorçage et du fichier à charger
4	Le poste télécharge le chargeur en TFTP
5	Le chargeur démarre iPXE, qui relance sa propre requête DHCP en se signalant par sa classe utilisateur
6	Le serveur reconnaît iPXE et lui sert cette fois le script `default.ipxe`

Étape	Ce qui se passe
7	Le script interroge l'interface web en HTTP et affiche le menu, ou déclenche directement la tâche en attente
8	Le poste charge le système temporaire, monte `/images` en NFS et lance Partclone

> **Pourquoi le poste interroge deux fois le réseau**
>
> La première requête vient du micrologiciel UEFI, qui ne sait faire qu'une chose : charger 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. Cette seconde requête se distingue par sa classe utilisateur, ce qui permet de lui répondre autre chose qu'au premier tour. Sans cette distinction, iPXE se verrait servir iPXE indéfiniment.

4.2 Le mode ProxyDHCP

Dans un déploiement classique, le serveur DHCP transmet lui-même l'adresse du serveur d'amorçage et le nom du fichier, par les options 66 et 67. Cela suppose de pouvoir configurer le DHCP — condition que ce réseau ne remplit pas.

Le mode proxy contourne l'obstacle. dnsmasq écoute les requêtes d'amorçage et y répond **sans distribuer d'adresse** : le poste reçoit deux réponses complémentaires, l'une de la box pour son adressage, l'autre du serveur FOG pour son amorçage. Aucune modification du réseau existant n'est nécessaire.

> **Une réponse qui dépasse le cadre de ce lab**
>
> Ce point n'est pas un contournement de circonstance. En clientèle, il est fréquent d'intervenir sur un site dont on ne maîtrise pas l'infrastructure réseau — DHCP tenu par un prestataire, équipement mutualisé, ou simple refus de toucher à un service de production. Le mode ProxyDHCP permet de déployer sans demander cette modification, donc sans dépendre d'un tiers ni prendre la responsabilité d'une coupure.

4.3 Secure Boot et protocole

Le fichier servi aux postes UEFI 64 bits est `snponly-shimx64.efi`, un chargeur signé par Microsoft. C'est le seul moyen d'amorcer un poste dont le Secure Boot est actif. Il est servi à tous les clients UEFI, y compris ceux dont le Secure Boot ne l'est pas : **un serveur DHCP ne peut pas connaître l'état du Secure Boot d'un poste**, cette information ne figure dans aucune option de la requête.

> **La conséquence : le serveur est en HTTP**
>
> Le chargeur signé ne valide pas les chaînes de certificats auto-signées. Un serveur FOG en HTTPS avec un certificat de lab deviendrait inaccessible aux postes qui passent par lui : **Secure Boot et HTTPS sont mutuellement exclusifs dans cette configuration**. Le serveur fonctionne donc en HTTP, sur un réseau privé. Ce n'est pas une négligence mais l'autre terme d'une alternative dont il fallait choisir un côté. En production, un certificat émis par une autorité reconnue du parc lèverait l'exclusion et permettrait les deux.

> **Cette contrainte vaut pour FOG 1.5.10 et la configuration retenue ici.** Elle doit être réévaluée lors d'un passage à FOG 1.6, dont la documentation distingue le transport web, le transport d'amorçage réseau et la préparation Secure Boot.

4.4 Ce que Secure Boot autorise, et où il s'arrête

La compatibilité du chargeur ne suffit pas à rendre la chaîne complète compatible. Un essai mené avec Secure Boot réellement activé sur un poste de test le montre : la chaîne fonctionne jusqu'à iPXE, puis s'arrête.

Étape	Sous Secure Boot actif
Chargement de `snponly-shimx64.efi`	Réussi — le chargeur est signé par Microsoft
Démarrage d'iPXE, ProxyDHCP, script `default.ipxe`	Réussi
Téléchargement du noyau et du système de fichiers initial	Réussi — les fichiers arrivent
Exécution du noyau FOS	**Refusée** — violation de la politique de sécurité

> **Le menu s'affiche, le déploiement ne démarre pas**
>
> C'est la distinction qu'il faut retenir. Voir apparaître le menu FOG prouve seulement que le chargeur signé a été accepté. Le noyau FOS chargé ensuite n'est pas signé par une clé que reconnaît le chargeur : avec Secure Boot actif, son exécution est refusée alors même que le fichier a été correctement téléchargé.
>
> **Les déploiements décrits dans ce dossier ont donc été menés Secure Boot inactif**, sur des machines créées sans clés pré-enregistrées. C'est une limite mesurée, pas une supposition.

> **La levée existe, mais elle ne passe pas à l'échelle**
>
> Le menu FOG propose une entrée d'enrôlement de clé, qui inscrit la clé du projet dans la liste des clés acceptées par le poste. La chaîne complète fonctionne ensuite sous Secure Boot. Mais l'opération exige une **confirmation manuelle sur chaque poste**, devant l'écran, au premier passage — ce qui contredit l'objectif d'un déploiement sans intervention. Sur un parc, la réponse serait plutôt de faire signer le noyau par une autorité déjà reconnue des postes, ou d'accepter de déployer Secure Boot inactif puis de le réactiver après installation.

5. Choix structurants

Sept décisions déterminent le comportement de l'ensemble. Elles sont rassemblées ici avec ce qu'elles coûtent, car un choix dont on ignore le prix finit par être remis en cause au mauvais moment.

5.1 Récapitulatif

Décision	Motif	Contrepartie
ProxyDHCP plutôt que migration du DHCP	Ne pas toucher au réseau existant	Un service supplémentaire à maintenir sur le serveur
HTTP plutôt que HTTPS	Compatibilité Secure Boot	Trafic d'administration non chiffré sur le réseau local
Debian 12 plutôt que Debian 13	Contrainte de l'installeur FOG 1.5.10	Une version de retard ; à revérifier avant toute montée de version
Logiciels en snapins plutôt que dans l'image	Mise à jour sans recapture, modularité par groupe	Le poste n'est complet qu'après exécution des snapins
Image « Single Disk - Resizable »	Déployable sur des disques de capacités différentes	Un seul disque traité par image
Enrôlement rapide plutôt que complet	Quelques secondes par poste devant la machine	Configuration à faire ensuite depuis l'interface
Disque des images exclu des sauvegardes	Données volumineuses et reconstructibles	Étape de recapture à prévoir au plan de reprise

5.2 La répartition des rôles après clonage

Un choix mérite d'être détaillé, car il conditionne la possibilité même d'utiliser une image unique sur tout le parc : la répartition entre ce qui est figé dans l'image et ce qui est appliqué poste par poste.

Élément	Porté par	Pourquoi là
Système, réglages, régionalisation	L'image	Identique sur tous les postes
Écrans de configuration initiale	`unattend.xml`	Réponses connues d'avance, identiques partout
Nom de la machine	Client FOG	Diffère par poste : ne peut pas figurer dans une image commune
Jonction au domaine	Client FOG	Diffère par poste ; les identifiants restent chiffrés côté serveur
Logiciels	Snapins	Modulable par poste ou par groupe, remplaçable sans recapture

Élément	Porté par	Pourquoi là
Réseau, accès distant, horloge	Script de post-déploiement	Dépend du matériel réellement présent sur le poste
Fond d'écran, lecteurs réseau	Stratégies de groupe	Relève de l'annuaire, pas du déploiement

Pourquoi la jonction au domaine n'est pas dans le script

Deux raisons. La première tient à la sécurité : joindre un domaine exige des identifiants, et les inscrire dans un script les exposerait en clair dans un fichier stocké sur le serveur puis copié sur chaque poste. Confiée au client FOG, l'opération laisse les identifiants chiffrés dans la base du serveur. La seconde tient à la cohérence : la jonction est une fonction native du client, configurée par hôte ou par groupe. La dupliquer dans un script créerait deux sources de vérité pour un même paramètre — la garantie d'une divergence à terme.

6. Résultats et limites

6.1 Ce qui a été validé

Fonction	Validation
Amorçage réseau	Chaîne complète observée : ProxyDHCP, chargeur signé, script iPXE, menu FOG, puis chargement de FOS — Secure Boot inactif
Capture d'image	Master Windows 11 généralisé et capturé — 16 Go pour un disque de 60 Go
Déploiement unitaire	Poste déployé, renommé, joint au domaine dans la bonne unité d'organisation
Déploiement de flotte	Trois postes déployés simultanément en multicast, flux mutualisé vérifié
Distribution logicielle	Quatre snapins exécutés, tous en code de retour zéro
Configuration post-déploiement	Réseau, accès distant, horloge et alimentation appliqués par script

6.2 L'apport du multicast, mesuré

La différence entre les deux modes de déploiement est le point le plus parlant du projet.

	Unicast	Multicast
Transport	NFS, un flux par poste	UDPCAST, un flux unique diffusé
Flux émis par le serveur	Un flux par poste	**Un seul flux**, quel que soit le nombre de postes
Débit observé, un poste	environ 10 Go/min	—
Débit observé, trois postes	le débit se divise entre eux	environ 4,4 Go/min reçus par chaque poste
Débit réseau du serveur, trois postes	la somme des flux	environ 4,4 Go/min — un seul flux émis
Comportement à la montée en charge	Se dégrade proportionnellement	Stable sur les trois postes mesurés
Usage	Remplacement d'un poste	Renouvellement de parc, ouverture de salle

Comment lire le chiffre de 13 Go/min

Les trois postes reçoivent environ 4,4 Go/min chacun, ce qui fait 13 Go/min si l'on additionne. **Ce total n'est pas le débit réseau du serveur.** FOG diffuse en multicast au moyen d'un émetteur unique : le serveur émet un seul flux d'environ 4,4 Go/min, que les trois postes écoutent simultanément. Les 13 Go/min mesurent le travail cumulé traité par les clients, pas ce qui sort de la carte réseau du serveur.

La distinction n'est pas cosmétique : c'est précisément parce que le serveur n'émet qu'un flux que l'ajout de postes ne le sature pas. Un total présenté comme un débit serveur laisserait croire l'inverse.

Conditions de la mesure

Ces chiffres ont été relevés sur **trois machines virtuelles hébergées par le même hôte Proxmox et raccordées au même pont virtuel**. Le trafic ne traverse aucun commutateur ni aucun lien physique : il transite en mémoire entre machines du même hyperviseur. Les valeurs absolues ne sont donc pas transposables à un réseau physique d'entreprise, où le débit dépendrait du câblage, des commutateurs et de leur gestion du multicast. **Ce qui reste transposable, c'est le rapport entre les deux modes**, qui découle du nombre de flux émis et non du support.

Seule la copie de l'image est mutualisée

Le gain du multicast porte exclusivement sur le transfert. Tout ce qui suit — configuration initiale, renommage, jonction au domaine, installation des logiciels — reste propre à chaque poste et s'exécute en parallèle sans mutualisation possible, puisque ces opérations diffèrent d'une machine à l'autre. Sur trois postes, le gain est réel mais modeste. Le seuil à partir duquel il devient déterminant — de l'ordre d'une dizaine de machines, quand l'unicast saturerait le lien du serveur — est une **estimation par extrapolation, non un résultat mesuré** : la validation n'a porté que sur trois postes.

6.3 Limites de la solution

Limite	Portée	Réponse possible
Pas d'injection de pilotes	Structurelle — parc hétérogène	Une image par modèle, ou injection en post-déploiement
Serveur en HTTP	Liée au Secure Boot	Certificat émis par une autorité reconnue du parc
Déploiement impossible sous Secure Boot actif	Le noyau FOS est refusé après chargement du menu	Enrôlement d'une clé poste par poste, ou noyau signé par une autorité reconnue du parc
Nœud de stockage unique	Pas de tolérance de panne sur le serveur	Nœud secondaire, réplication déjà prévue par FOG
Compte de jonction privilégié	Écart de sécurité identifié	Compte de service dédié, délégué sur la seule OU visée
Mot de passe local commun à tous les postes	Compromission d'un poste vaut compromission du parc	Windows LAPS : mot de passe unique par machine, stocké dans l'annuaire
Le serveur n'est pas supervisé	Occupation du volume des images non surveillée	Agent Zabbix — port déjà ouvert en prévision

Pourquoi ces limites figurent dans un document de présentation

> Un projet dont on ne connaît que les réussites ne peut pas être repris. Les six lignes ci-dessus sont les points sur lesquels un lecteur qui transposerait cette solution rencontrerait un problème — et pour chacun, la réponse est connue et documentée. Les inscrire ici est ce qui distingue une solution évaluée d'une solution simplement mise en service.

6.4 Évolutions

Évolution	Apport
Intégration à la supervision	Alerte sur la saturation de `/images`, dont la réserve système a été volontairement supprimée
Compte de service dédié à la jonction	Suppression d'un privilège élevé exposé sur chaque poste déployé
Windows LAPS	Fin du mot de passe local partagé
Déploiements planifiés avec réveil par le réseau	Renouvellement de parc hors heures de travail, sans intervention
Nœud de stockage secondaire	Répartition de charge et tolérance de panne

6.5 Bilan

Les six objectifs du chapitre 1 sont atteints. Le temps d'installation par poste passe d'environ une heure à quelques minutes de transfert suivies d'une phase automatisée ; l'uniformité est garantie par construction ; les postes rejoignent le domaine dans la bonne unité d'organisation sans saisie ; le déploiement simultané fonctionne sans dégradation du débit par poste ; le réseau existant n'a pas été modifié ; et les snapins permettent d'intervenir sur un parc déjà en production.

Ce qui reste ouvert relève de la sécurisation et de l'exploitation courante — compte de service, LAPS, supervision — et non du fonctionnement de la chaîne de déploiement, qui est complète et validée de bout en bout.

Références

Sources consultées pour l'ensemble du dossier (documents A à E). Les documentations en ligne évoluent : la date de consultation est celle de la rédaction, août 2026.

Projet FOG

Source	Usage
Documentation officielle — docs.fogproject.org	Installation du serveur, branches du dépôt, exigences réseau et pare-feu, transport et PKI
Dépôt du projet — github.com/FOGProject/fogproject	Sources de l'installeur, organisation des branches, format de version
Suivi des anomalies du projet	Compatibilité de l'installeur avec Debian 13 (anomalie n° 797)
Forum du projet — forums.fogproject.org	Comportement du client, diagnostic d'amorçage

Microsoft

Source	Usage
Microsoft Learn — Sysprep (généralisation)	Effets et limites de la généralisation, nombre d'exécutions autorisées
Microsoft Learn — Automatisation de l'OOBE	Fichier de réponses, écrans masqués, comptes locaux
Microsoft Learn — Confirm-SecureBootUEFI	Contrôle de l'état du démarrage sécurisé sur un poste
Microsoft Learn — Test-ComputerSecureChannel	Contrôle du canal sécurisé avec le contrôleur de domaine
Microsoft Learn — Recommandations IPv6	Pourquoi ne pas désactiver la pile IPv6

Infrastructure

Source	Usage
Documentation Proxmox VE	Configuration des machines virtuelles, OVMF, ordre d'amorçage
Documentation Debian 12	Installation, partitionnement, gestion des services
Pages de manuel — dnsmasq, ufw, ext4, findmnt	Directives de configuration et options de commande

> **Pourquoi cette liste figure dans le document A**

> Les cinq volumes s'appuient sur les mêmes sources. Les rassembler ici évite de les répéter, et signale au lecteur que les éléments repris de ces documentations ont une origine identifiable — ce qui compte particulièrement pour les points où la documentation du projet fait autorité sur ce que l'on observe, comme l'organisation des branches ou les exigences réseau.