PROJET FOG — DOCUMENT E

Snapins : installation logicielle et script de post-déploiement

Mode opératoire — Lab TSSR / Infrastructure Proxmox

Auteur	Julien BARBET
Version	1.0
Date	28 août 2026
Serveur	SRV-FOG-01 M192.168.1.207 MFOG 1.5.10.2254
Script	Post-Deploiement-v3.ps1
Domaine	lab.julien.local
Contexte	Lab personnel de formation

1. Objet du document

Un snapin est un paquet exécuté par le client FOG sur un poste, après le déploiement de l'image ou à la demande. Ce document décrit les quatre snapins mis en place sur cette infrastructure et détaille le script de post-déploiement qui constitue le plus important d'entre eux.

1.1 Pourquoi ne pas tout mettre dans l'image

Il serait techniquement possible d'installer les logiciels sur le master avant la capture. Ce choix a été écarté pour trois raisons.

Critère	Logiciel dans l'image	Logiciel en snapin
Mise à jour	Impose de reconstruire le master, de le généraliser et de recapturer l'image entière	Il suffit de remplacer le fichier d'installation sur le serveur
Modularité	Tous les postes reçoivent les mêmes logiciels	Chaque poste ou groupe reçoit ce qui le concerne
Taille de l'image	Croît avec chaque logiciel ajouté	Reste minimale
Parc existant	Impose un redéploiement complet du poste	Le snapin peut être envoyé seul, sans réimager

> **Le snapin est aussi un outil de télédistribution**
>
> FOG permet d'envoyer un snapin à un poste déjà en production, sans le redéployer. Un snapin n'est donc pas seulement un composant de la chaîne de déploiement : c'est un moyen de distribuer un logiciel ou d'exécuter un script sur tout ou partie du parc. Cette capacité justifie à elle seule de sortir les logiciels de l'image.

1.2 Les quatre snapins

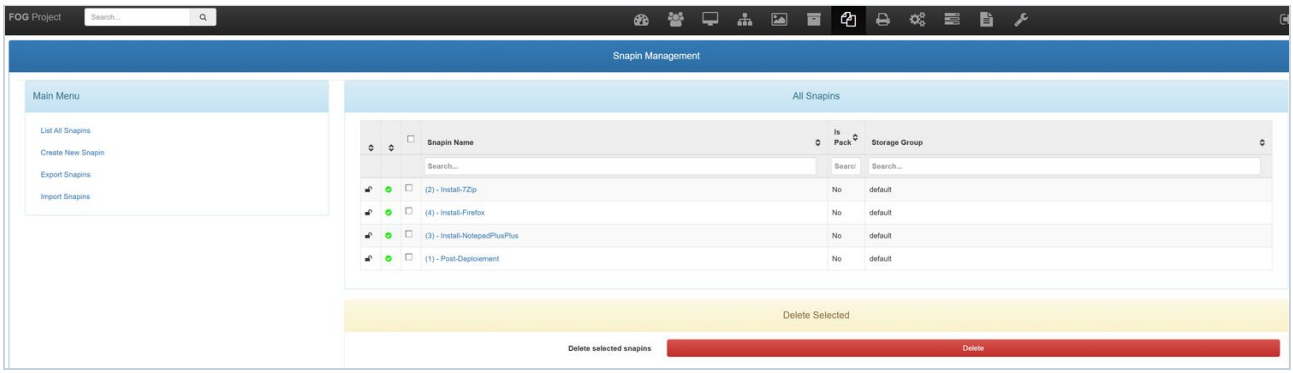

Figure 1 — Snapin Management — les quatre snapins déclarés sur le serveur

Snapin	Type	Rôle
Install-7Zip	MSI	Archiveur

Install-Firefox	MSI	Navigateur
Install-NotepadPlusPlus	MSI	Éditeur de texte
Post-Deploiement	PowerShell	Configuration réseau, accès distant, horloge, alimentation

La colonne **Is Pack** indique *No* pour les quatre : ce sont des snapins simples, exécutant un fichier unique. FOG gère également des paquets multi-fichiers, non utilisés ici.

2. Snapins d'installation logicielle

Les trois snapins logiciels reposent sur le même principe : un paquet MSI installé en mode silencieux.

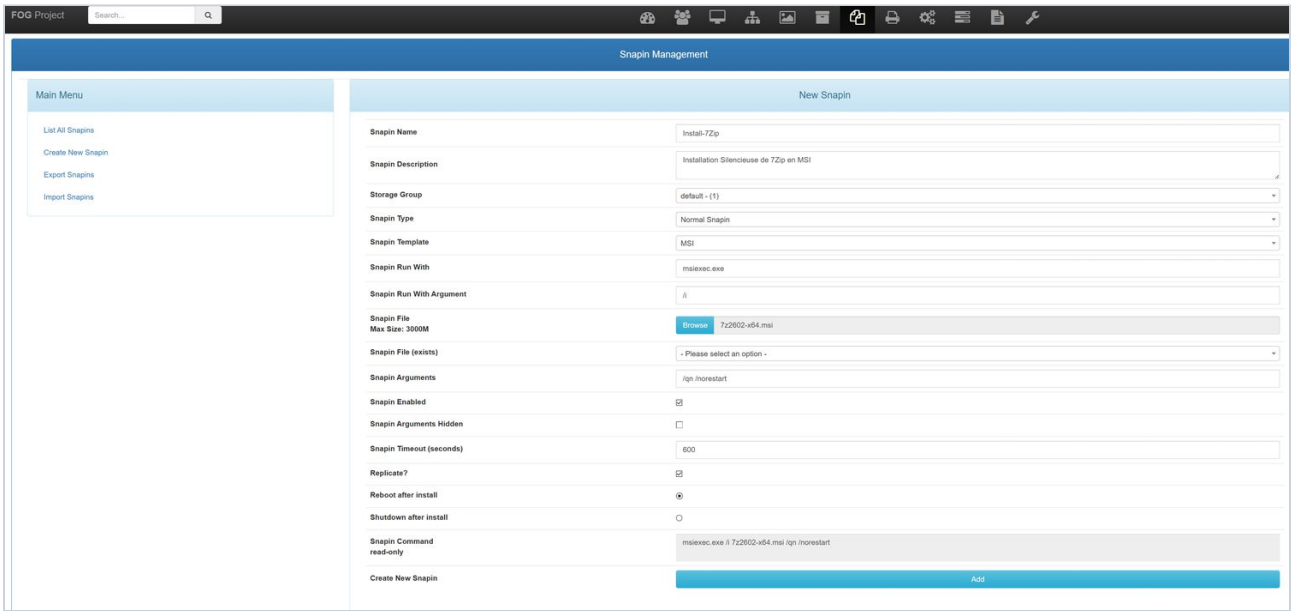

Figure 2 — Fiche du snapin Install-7Zip

2.1 Champs de configuration

Champ	Valeur	Rôle
Snapin Template	MSI	Modèle qui pré-remplit les champs d'exécution
Snapin Run With	`msiexec.exe`	Programme qui traite le fichier
Snapin Run With Argument	`/i`	Argument placé avant le nom du fichier : mode installation
Snapin File	Le paquet MSI téléversé	Fichier stocké sur le serveur et transmis au poste
Snapin Arguments	`/qn /norestart`	Arguments placés après le nom du fichier
Snapin Timeout	600 secondes	Délai au-delà duquel FOG considère l'exécution en échec
Replicate	Coché	Réplication du fichier vers les nœuds de stockage secondaires

La ligne **Snapin Command**, en lecture seule, montre la commande finalement construite :

```
msiexec.exe /i 7z2602-x64.msi /qn /norestart
```

> **Le rôle des deux arguments silencieux**
>
> `/qn` supprime toute interface : ni fenêtre, ni barre de progression, ni bouton. Sans lui, l'installeur attendrait un clic que personne ne donnera, puisque le snapin s'exécute sous le compte Système et qu'aucune session n'est nécessairement ouverte. Le délai d'expiration finirait par considérer l'opération en échec.
>
> `/norestart` empêche l'installeur de redémarrer le poste de sa propre initiative. Le redémarrage doit rester piloté par FOG, qui enchaîne plusieurs snapins : un redémarrage intempestif au milieu de la séquence interromprait les suivants.

2.2 Les deux autres snapins

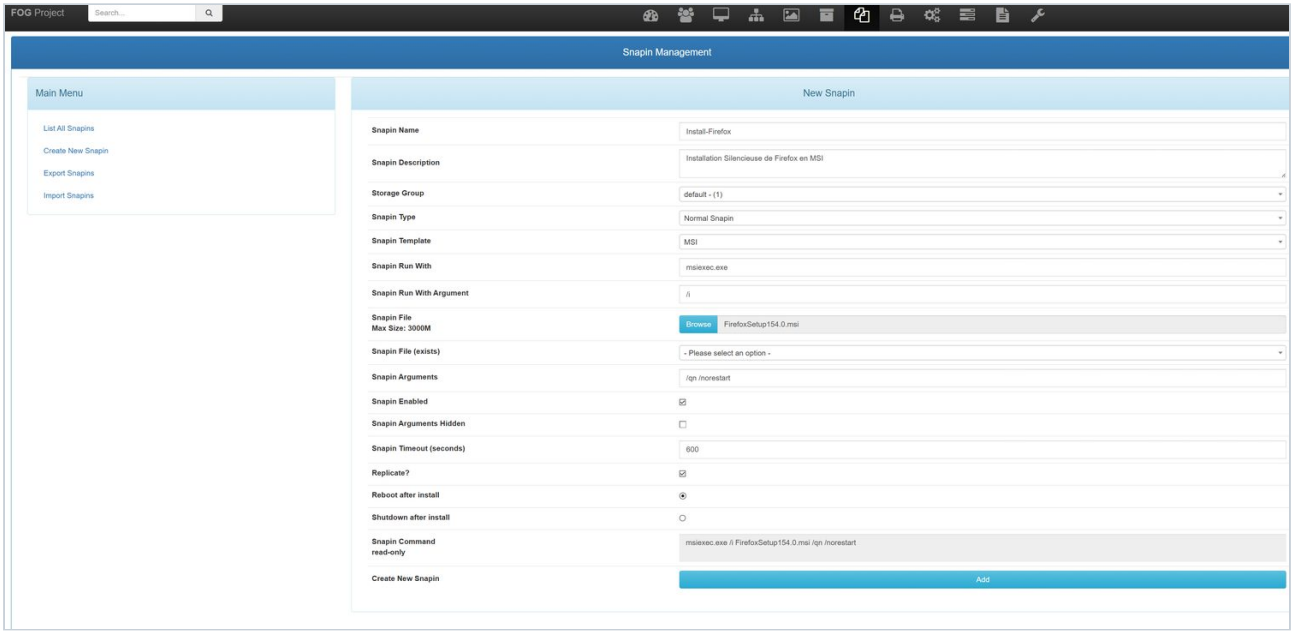

Figure 3 — Fiche du snapin Install-Firefox

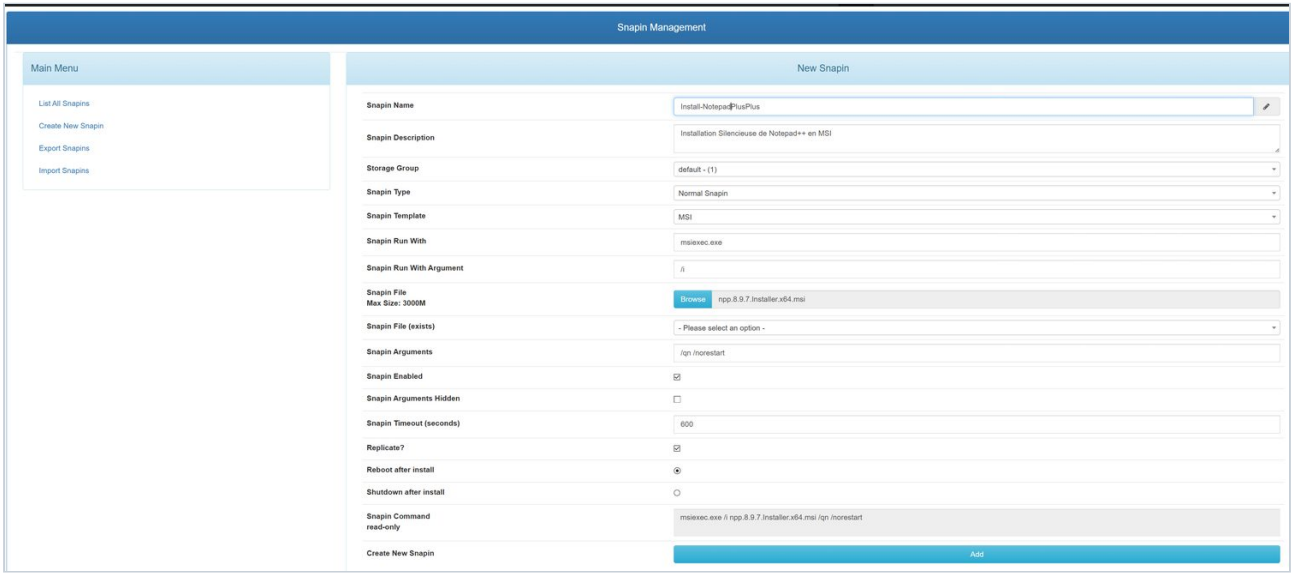

Figure 4 — Fiche du snapin Install-NotepadPlusPlus

La structure est identique. Seuls le nom du snapin, sa description et le fichier MSI changent. Cette uniformité est volontaire : elle rend la maintenance prévisible et permet de créer un nouveau snapin logiciel en quelques minutes.

> **Le format MSI n'est pas anodin**
>
> Un installeur au format `.exe` n'accepte pas nécessairement d'arguments silencieux, et ceux qu'il accepte varient d'un éditeur à l'autre. Le format MSI, lui, est traité par un moteur unique de Windows dont les arguments sont normalisés. Chercher la version MSI d'un logiciel avant de l'intégrer au déploiement évite les comportements imprévisibles — c'est le critère de sélection retenu pour les trois logiciels.

3. Le snapin de post-déploiement

Ce quatrième snapin ne distribue pas un logiciel : il exécute un script PowerShell qui configure le poste. Sa fiche diffère donc des précédentes.

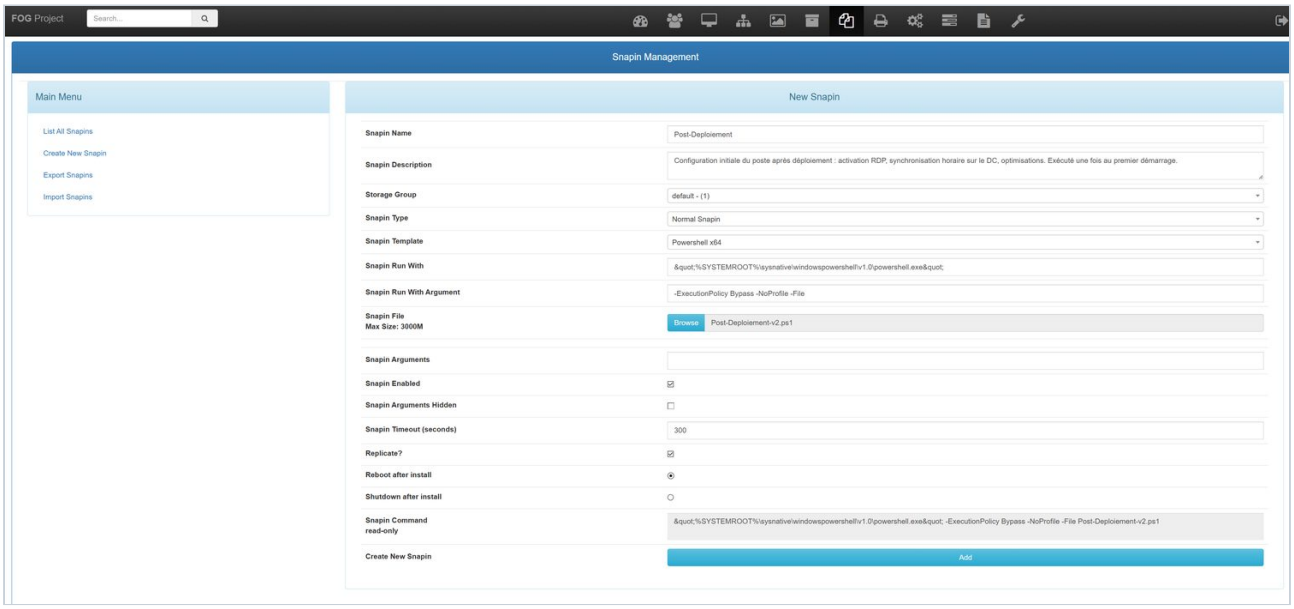

Figure 5 — Fiche du snapin Post-Deploiement

Champ	Valeur
Snapin Template	Powershell x64
Snapin Run With	`"%SYSTEMROOT%\sysnative\windowspowershell\v1.0\powershell.exe"`
Snapin Run With Argument	`-ExecutionPolicy Bypass -NoProfile -File`
Snapin File	Post-Deploiement-v3.ps1
Snapin Timeout	300 secondes
Reboot after install	Activé

> **Pourquoi le chemin sysnative**
>
> Le client FOG est un processus 32 bits. Sur un Windows 64 bits, tout accès d'un processus 32 bits à `C:\Windows\System32` est silencieusement redirigé vers `SysWOW64`, qui contient les versions 32 bits des programmes. Le script s'exécuterait alors dans un PowerShell 32 bits, dont plusieurs applets réseau ne voient pas l'intégralité du système. `sysnative` est un alias virtuel qui contourne cette redirection et donne accès au véritable `System32`. Sans lui, les commandes de configuration réseau du script échoueraient sans raison apparente.

Pourquoi contourner la stratégie d'exécution

Windows refuse par défaut d'exécuter un script PowerShell non signé. `-ExecutionPolicy Bypass` lève cette restriction **pour cette exécution uniquement** : la stratégie globale de la machine n'est pas modifiée. C'est préférable à un assouplissement permanent, qui autoriserait ensuite n'importe quel script sur le poste. `-NoProfile` évite le chargement des profils utilisateur, à la fois pour la vitesse et pour garantir que le script s'exécute dans un environnement identique sur tous les postes.

Un délai d'expiration plus court que pour les MSI

300 secondes contre 600 pour les installations logicielles. Un script de configuration ne doit pas dépasser quelques dizaines de secondes ; au-delà, c'est qu'une commande est bloquée en attente. Un délai court fait remonter l'anomalie rapidement dans l'historique des snapins, plutôt que d'immobiliser la séquence pendant dix minutes.

4. Contenu du script

Le script s'exécute sous le compte Système, sans interface. Il est structuré en sept sections, chacune tolérante aux erreurs : l'échec d'une étape n'interrompt pas les suivantes, sauf pour la détection réseau dont tout le reste dépend.

4.1 Répartition des rôles

L'en-tête du script énonce explicitement ce qu'il ne fait pas :

Opération	Prise en charge par
Renommage du poste	Client FOG, module HostnameChanger
Jonction au domaine	Client FOG, onglet Active Directory
Installation des logiciels	Snapins dédiés
Fond d'écran, lecteurs réseau	Stratégies de groupe

> **Pourquoi la jonction au domaine n'est pas dans le script**
>
> Deux raisons. La première est de sécurité : joindre un domaine nécessite des identifiants. Les inscrire dans le script les exposerait en clair dans un fichier stocké sur le serveur et copié sur chaque poste. En laissant FOG s'en charger, les identifiants restent dans la base du serveur, chiffrés. La seconde est de cohérence : la jonction est une fonction native du client FOG, configurée par hôte ou par groupe dans l'interface. La dupliquer dans un script créerait deux sources de vérité pour un même paramètre.

4.2 Journalisation

Le script s'exécutant sans interface, le journal est sa seule trace exploitable. Chaque action est horodatée et qualifiée par un niveau :

```
function Ecrire-Journal {
    param(
        [Parameter(Mandatory = $true)][string]$Message,
        [ValidateSet('INFO','OK','AVERTISSEMENT','ERREUR')][string]$Niveau = 'INFO'
    )
    $Horodatage = Get-Date -Format 'yyyy-MM-dd HH:mm:ss'
    $Ligne = "$Horodatage  [$Niveau]  $Message"
    Add-Content -Path $FichierLog -Value $Ligne -Encoding UTF8
    Write-Output $Ligne
}
```

Le journal est écrit dans `C:\Windows\Temp\post-deploiement.log`. Il constitue le premier point de diagnostic en cas de comportement anormal sur un poste déployé.

4.3 Détection de l'interface réseau

```
$RouteDefaut = Get-NetRoute -DestinationPrefix '0.0.0.0/0' |
               Sort-Object -Property RouteMetric |
               Select-Object -First 1

$Interface = Get-NetAdapter -InterfaceIndex $RouteDefaut.InterfaceIndex
```

Le nom d'une carte réseau varie selon le matériel : *Ethernet*, *Ethernet 2*, *Wi-Fi*. Le coder en dur rendrait le script inutilisable dès le premier poste au matériel différent.

La méthode retenue identifie l'interface qui porte la **route par défaut IPv4 de plus faible métrique**, c'est-à-dire celle réellement utilisée pour sortir du réseau local. C'est la seule étape dont l'échec interrompt le script : toute la configuration suivante s'applique à cette interface.

4.4 Désactivation de DHCPv6

```
Set-NetIPInterface -InterfaceIndex $Interface.InterfaceIndex `
                   -AddressFamily IPv6 -Dhcp Disabled
```

> **Un diagnostic propre à ce laboratoire**
>
> Windows privilégie IPv6 sur IPv4 pour la résolution de noms. Tant que la pile IPv6 reçoit un serveur DNS de la box par DHCPv6, c'est celui-ci qui répond en premier — et la correction appliquée côté IPv4 reste sans effet. La jonction au domaine échoue alors avec le code 1355, domaine introuvable, alors même que le DNS IPv4 est correctement configuré.
>
> **La solution n'est pas de désactiver IPv6.** Couper IPv6 au niveau de la carte casse le service NLA (Network Location Awareness), qui ne reconnaît alors plus le profil réseau comme « Domaine » : les règles de pare-feu du profil Domaine ne s'appliquent plus jamais. Le script coupe donc uniquement **DHCPv6**, sur cette interface, en laissant la pile IPv6 active — désactiver IPv6 est déconseillé.

4.5 Configuration DNS et vérification

```
Set-DnsClientServerAddress -InterfaceIndex $Interface.InterfaceIndex `
                           -ServerAddresses $DnsServeur
Clear-DnsClientCache
```

Le poste conserve l'adresse IP distribuée par le DHCP du réseau ; seule la résolution de noms est redirigée vers le DC ; contrôle par Get-DnsClientServerAddress.

Le script vérifie ensuite que le domaine est effectivement joignable, avant que le client FOG ne tente la jonction :

```
Resolve-DnsName -Name "_ldap._tcp.dc._msdcs.$NomDomaine" `
                -Type SRV -Server $DnsServeur
```

> **Pourquoi interroger un enregistrement SRV**
>
> Un poste ne trouve pas un contrôleur de domaine par son nom, mais par des enregistrements de service publiés dans le DNS. `_ldap._tcp.dc._msdcs.lab.julien.local` est celui qui liste les contrôleurs offrant le service LDAP. Si cette requête aboutit, la jonction réussira ; si elle échoue, elle échouera avec le code 1355. Le script anticipe donc le résultat et l'inscrit au journal, ce qui transforme un échec silencieux en information exploitable.

4.6 Accès distant

```
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal Server' `
                 -Name 'fDenyTSConnections' -Value 0

Set-ItemProperty -Path 'HKLM:\...\WinStations\RDP-Tcp' `
                 -Name 'UserAuthentication' -Value 1

Enable-NetFirewallRule -Group '@FirewallAPI.dll,-28752'
```

Le service est activé, l'authentification au niveau réseau est exigée, et le pare-feu est ouvert. L'accès reste contrôlé par le groupe de sécurité `GG-RDP-Lab`, appliqué par stratégie de groupe : activer le service n'autorise personne à s'y connecter.

> **Pourquoi cibler la règle par son identifiant de ressource**
>
> Le groupe de règles de pare-feu porte un nom traduit : *Bureau à distance* en français, *Remote Desktop* en anglais. Un script qui cible ce nom cesse de fonctionner dès qu'il rencontre un système dans une autre langue. `@FirewallAPI.dll,-28752` est l'identifiant de la chaîne dans la bibliothèque système : il est identique quelle que soit la langue installée. C'est le réflexe à avoir pour tout script destiné à un parc hétérogène.

4.7 Synchronisation horaire

```
$InfosSysteme = Get-CimInstance -ClassName Win32_ComputerSystem

if ($InfosSysteme.PartOfDomain) {
    w32tm /config /syncfromflags:domhier /update
    Restart-Service w32time
    w32tm /resync
}
```

Kerberos rejette toute authentification dont l'horodatage dérive de plus de cinq minutes par rapport au contrôleur de domaine. Un poste à l'heure fausse ne peut plus ouvrir de session sur le domaine, sans message explicite.

La condition est importante : la synchronisation sur la hiérarchie du domaine n'a de sens que si le poste en est membre. Au premier passage du script, la jonction n'a pas encore eu lieu — le script le constate et l'inscrit au journal plutôt que d'échouer.

4.8 Alimentation

```
powercfg /change standby-timeout-ac 0
powercfg /change hibernate-timeout-ac 0
powercfg /change monitor-timeout-ac 15
```

Un poste en veille devient injoignable : ni prise en main à distance, ni supervision, ni réception de nouveaux snapins. La veille et la mise en veille prolongée sont désactivées sur secteur ; l'extinction de l'écran est conservée à quinze minutes, sans incidence sur la disponibilité de la machine.

4.9 Code de sortie

```
exit 0
```

Le code de sortie est la valeur que FOG enregistre dans l'historique des snapins. Le script retourne toujours zéro : ce zéro atteste que le script s'est exécuté, non que chaque étape a réussi. Le journal du poste reste la seule preuve ; retourner 1 sur échec critique reste à faire.

5. Association et exécution

5.1 Association à un hôte

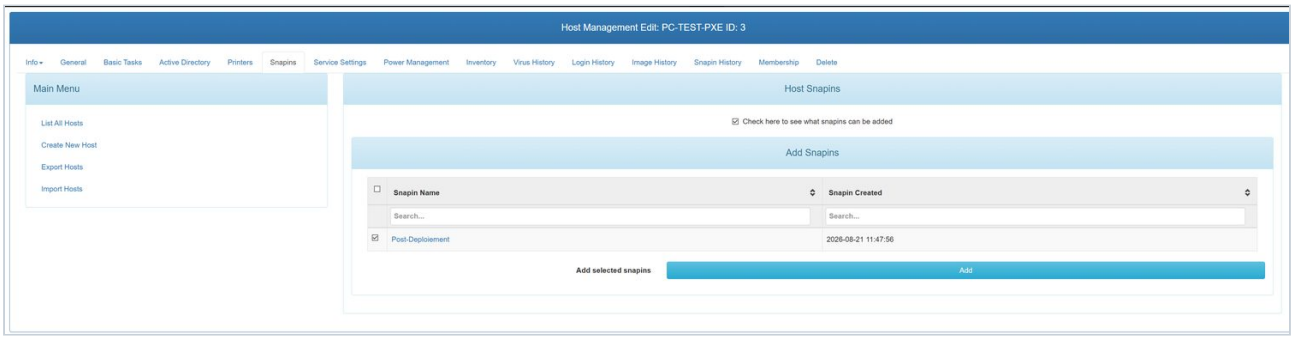

Figure 6 — Onglet Snapins d'un hôte — ajout d'un snapin à la sélection

La case **Check here to see what snapins can be added** affiche les snapins disponibles mais non encore associés. Ceux qui le sont déjà apparaissent dans la liste **Remove snapins**.

L'association peut également se faire au niveau d'un groupe, auquel cas elle se propage à tous ses membres. Ce mécanisme est décrit dans le document D.

5.2 Ordre d'exécution

Le client FOG exécute les snapins associés à un hôte les uns après les autres, sans que l'ordre soit configurable directement. Sur cette infrastructure, la séquence observée place les installations logicielles avant le script de post-déploiement.

> **Ce que cet ordre implique**
>
> Le script de post-déploiement contient une étape conditionnée à l'appartenance au domaine. En s'exécutant en dernier, il a le plus de chances de trouver la jonction déjà effectuée et de synchroniser l'horloge dans la foulée. Si ce n'était pas le cas, le script le consigne au journal et la synchronisation devra être relancée : le snapin ne repasse pas de lui-même. Un script de configuration doit être conçu pour ne pas dépendre d'un ordre qu'il ne maîtrise pas.

5.3 Exécution sur le poste

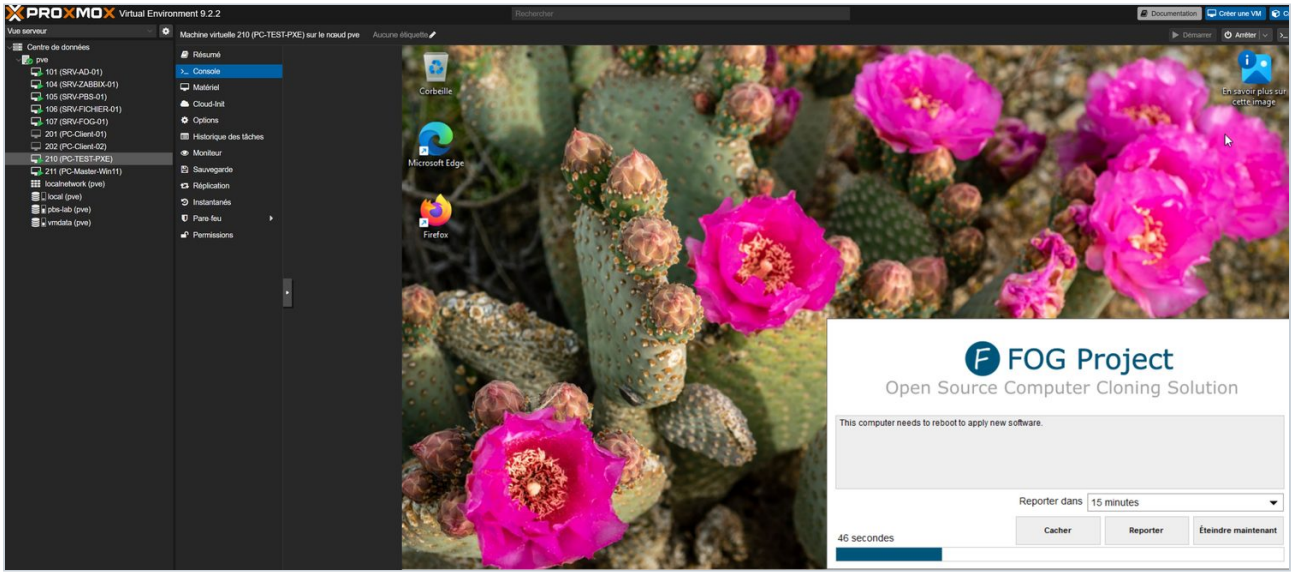

Figure 7 — Le client FOG signale un redémarrage nécessaire après installation ; Firefox est déjà présent sur le bureau

Le client informe l'utilisateur qu'un redémarrage est requis et propose de le différer. Ce comportement découle de l'option **Reboot after install** activée sur le snapin de post-déploiement.

> **Le client diffère ses actions quand une session est ouverte**
>
> Le module HostnameChanger du client FOG ne renomme pas une machine tant qu'une session utilisateur est ouverte : le renommage impose un redémarrage, que le client évite d'infliger à un utilisateur en cours de travail. Il réessaie automatiquement toutes les 90 secondes. Sur un poste dont la session s'ouvre automatiquement, ce comportement peut donner l'impression d'un blocage : fermer la session débloque immédiatement la séquence. C'est la raison pour laquelle l'ouverture de session automatique a été retirée du fichier de réponses du master (document C, section 4.2).

5.4 Envoi d'un snapin sans redéploiement

Depuis la fiche d'un hôte, **Basic Tasks → Advanced** donne accès à l'envoi de snapins seuls, sans réimager le poste. Deux options sont disponibles :

Option	Effet
Single Snapin	Envoie un seul snapin choisi dans la liste
All Snapins	Envoie tous les snapins associés à l'hôte

C'est la fonction de télédistribution mentionnée au chapitre 1 : elle permet de corriger une configuration ou de déployer un logiciel sur un parc en production, sans toucher aux données des utilisateurs.

6. Validation

6.1 Historique côté serveur

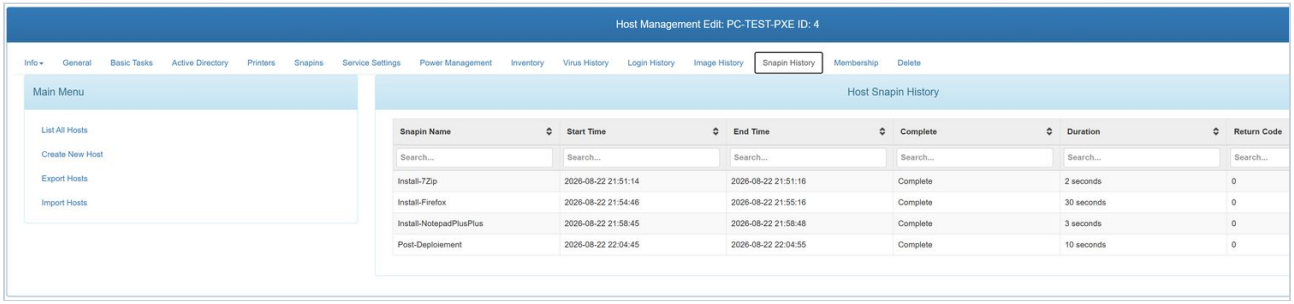

Figure 8 — Snapin History de l'hôte — les quatre snapins avec leur code de retour

Colonne	Interprétation
Snapin Name	Snapin exécuté
Start Time / End Time	Bornes de l'exécution
Complete	État de fin de traitement
Duration	Durée réelle. À comparer au délai d'expiration configuré.
Return Code	**Le point décisif.** Zéro signale une réussite ; toute autre valeur indique une anomalie à investiguer.

Les durées relevées sont cohérentes avec la nature des paquets : quelques secondes pour les petits MSI, une trentaine de secondes pour le navigateur, une dizaine pour le script de configuration. Une durée anormalement longue, proche du délai d'expiration, est le premier signe d'une commande bloquée.

6.2 Journal côté poste

Le serveur indique qu'un snapin s'est exécuté, mais pas ce qu'il a fait. Le détail se trouve dans le journal du script :

```
Get-Content C:\Windows\Temp\post-deploiement.log
```

Les lignes à vérifier en priorité :

Ligne recherchée	Ce qu'elle confirme
Interface retenue	La détection automatique a fonctionné sur ce matériel
DHCPv6 desactive	La priorité IPv6 est neutralisée
Serveurs DNS declares	Le poste interroge bien le contrôleur de domaine

Controleur(s) de domaine detecte(s)	La ligne décisive : la jonction pourra aboutir
Bureau a distance active	Le service et le pare-feu sont ouverts
Horloge synchronisee	Kerberos ne rejettera pas l'authentification

> **Deux niveaux de vérification complémentaires**
>
> L'historique des snapins prouve que FOG a bien transmis et lancé le paquet ; c'est une information de chaîne de déploiement. Le journal du script prouve que les actions ont produit l'effet attendu sur le poste ; c'est une information de configuration. Un snapin peut parfaitement retourner le code zéro alors qu'une de ses étapes internes a échoué en étant traitée en avertissement. Les deux sources sont donc à consulter.

6.3 Liste de contrôle

Point	Contrôle
Snapins associés	Les quatre apparaissent dans l'onglet Snapins de l'hôte
Exécution	Les quatre sont présents dans Snapin History
Codes de retour	Tous à zéro
Durées	Cohérentes, loin du délai d'expiration
Logiciels	Présents dans la liste des applications installées
Journal du script	Aucune ligne de niveau ERREUR
Résolution du domaine	Ligne « Controleur(s) de domaine detecte(s) » présente
Accès distant	Connexion effective depuis un autre poste, avec un compte du domaine