Déploiement avec FOG sous Virt-Manager

Cette page n'est plus d'actualité → Nouvelle page plus récente avec FOG 1.6 : https://documentation.primtux.fr/books/manuel-de-ladministrateur/page/deploiement-avec-fog-16-sous-virt-manager

AVERTISSEMENT : Virt-Manager ne fonctionne que sur des ordinateurs sous GNU/Linux !!!

Il existe une page décrivant l'utilisation de FOG avec VirtualBox ici : Déploiement avec FOG sous VirtualBox.

Principe de fonctionnement et intérêts de FOG

Si l'on veut déployer Primtux sur plusieurs ordinateurs de même modèle (cas typique dans une école possédant généralement plusieurs exemplaires d'une même série), il peut être intéressant de le faire à l'aide d'un serveur FOG. On a tout intérêt à installer ce serveur sur un ordinateur portable afin de pouvoir le déplacer facilement sur site.

Site officiel : https://fogproject.org/

Documentation en cours de construction : https://docs.fogproject.org/en/latest/

La solution FOG permet via un réseau Intranet à la fois de récupérer une image d'une machine "modèle" et ensuite de la déployer sur plusieurs machines en même temps, ce qui représente un gros gain de temps par rapport à une installation via clé USB avec CloneZilla.

Matériel

- Un ordinateur avec le serveur FOG installé dessus.
- Un swith Gigabit d'au moins 5 ports RJ45 qui permettra le déploiement de 4 machines en même temps.
- 5 câbles Ethernet pour relier les ordinateurs ainsi que le serveur FOG au switch.

Préparation de la machine virtuelle sous Debian 13

Il n'est pas nécessaire de réserver un ordinateur à cette seule tâche, ce qui serait quand même dommage car on ne déploie pas tous les jours non plus ;-). Ceci est rendu possible grâce à la virtualisation. Pour la virtualisation, on utilise ici Virt-Manager (VMM) qui fonctionne mieux et est plus léger que VirtualBox.
Pour une utilisation confortable, il vaudra mieux disposer d'un ordinateur assez récent avec 4 Go de Ram au minimum.

Sur cette page, nous décrirons l'**installation d'un serveur FOG sur un ordinateur tournant déjà sous une distribution GNU/Linux, ici en l'occurrence une Linux Mint 22.3** (qui repose elle-même sur Ubuntu 24.04).

Remarque : On peut reprendre les indications ci-dessous pour installer FOG directement sur une Debian 13 sans avoir à virtualiser.

A- Installation de Virt-Manager

Page du site officiel de Virt-Manager : https://virt-manager.org/

On retrouve Virt-Manager dans la plupart des dépôts des distributions GNU/Linux. Du coup, il suffit d'installer le paquet avec la commande suivante pour les distributions dérivées de Debian comme Ubuntu ou Linux Mint :

```
sudo apt install virt-manager
```

Il faudra **redémarrer son ordinateur** afin que Virt-Manager soit fonctionnel (sinon on risque d'avoir des messages d'erreurs concernant "libvirt").

Le reste des explications s'appuieront sur une utilisation sous Linux Mint mais cela n'est pas très différents pour d'autres systèmes.

B- Virtualisation de Debian 13

On récupère l'image iso du système d'exploitation Debian 13.3 (netinstall) sur le site https://www.debian.org/index.fr.html. Puis on crée une machine virtuelle dans Virt-Manager en installant Debian à partir de l'image récupérée précédemment (voir ce tutoriel explicitant comment créer une image virtuelle et les réglages utiles). Lors de lancement de l'iso, choisir "Install" simple, garder les cases cochées par défaut et cocher les cases pour installer un Web serveur (Apache) et éventuellement l'accès SSH si utile.

C- Création d'un connexion réseau de type bridge avec nmcli

*Précaution : il est fortement recommandé de créer un **instantané** de son système avec **TimeShift** afin de pouvoir revenir à un état antérieur avant les manipulations ci-dessous...*

La connexion réseau entre la machine hôte (host) sous Linux Mint 22.3 et la machine invitée (guest) sous Debian 13 doit être de type "Bridge" (pont) et non pas NAT qui est le réglage par défaut !!! Le mode bridge permet aux autres machines sur le réseau de pouvoir récupérer à travers le pc hôte, le serveur FOG de la machine virtuelle : le serveur FOG sera visible directement dans le LAN.
Schéma :

PC physique bootant en PXE --- Switch ---> PC hôte sous Linux Mint 22.3 --- Bridge br0 ---> PC invité sous Debian 13 avec Serveur FOG (DHCP / TFTP / HTTP)

Pour cela, on va utiliser la ligne de commande de Network-Manager, nmcli qui permet une configuration fine.

1. On va chercher le nom de notre carte réseau Ethernet sur la machine hôte avec la commande :

   ```
   ip a
   ```

   Sur ma machine, je trouve la carte "**enp1s0**".

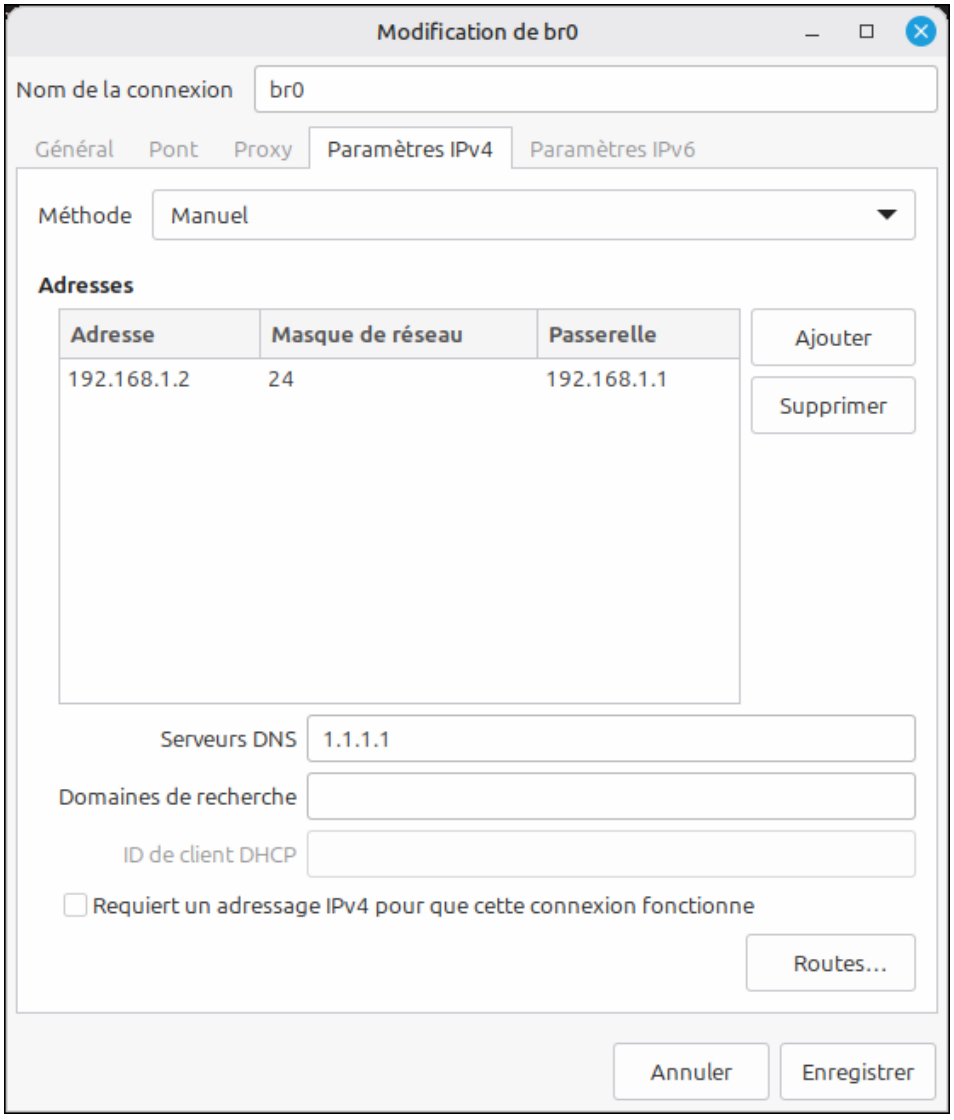
2. Il faut ensuite créer une connexion de type bridge dans la machine hôte avec les commandes :

   ```
   nmcli connection add type bridge ifname br0 con-name br0
   ```

3. On va ensuite l'IP fixer de br0 en 192.168.1.2/24 avec la commande suivante :

```
nmcli connection modify br0 ipv4.method manual ipv4.addresses 192.168.1.2/24
ipv4.gateway 192.168.1.1 ipv4.dns 1.1.1.1
```

   Cette commande correspond aux réglages que l'on ferait via l'outil "Configuration Réseau Avancée" :



4. Il ne faut pas que l'interface enp1s0 ait d'IP avec cette commande :

```
nmcli connection modify bridge-slave-enp1s0 ipv4.method disabled
```

5. Ensuite, il faut associer la carte réseau physique **enp1s0** au bridge avec la commande :

```
nmcli connection add type ethernet ifname enp1s0 master br0
```

6. On active la connexion br0 avec la commande :

```
nmcli connection up br0
```

7. Avec le câble Ethernet branché sur l'ordinateur hôte, on vérifie si le bridge br0 a bien une adresse IP du réseau LAN avec :

```
ip a show br0
```

ou

```
nmcli connection show br0
```

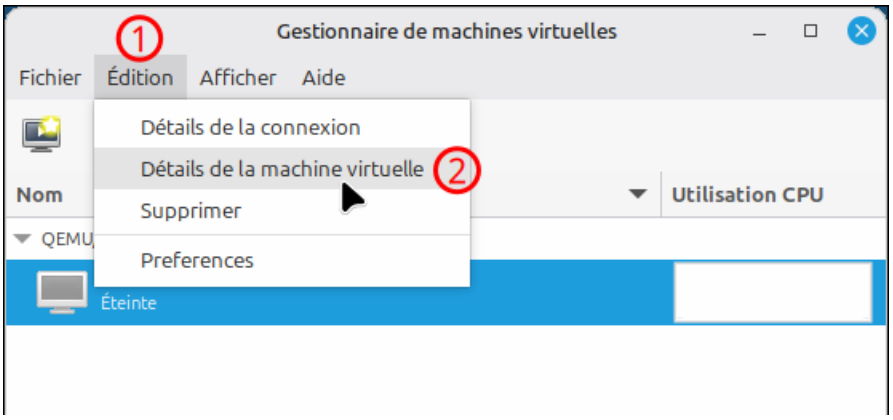
Info utiles :
La commande nmcli est très puissante avec beaucoup de fonctionnalités dont un prompt. Voir cette page pour plus d'info : https://debian-facile.org/doc:reseau:network-manager

- Pour supprimer l'interface réseau br0 (ou d'autres) créée (en cas d'erreur) :

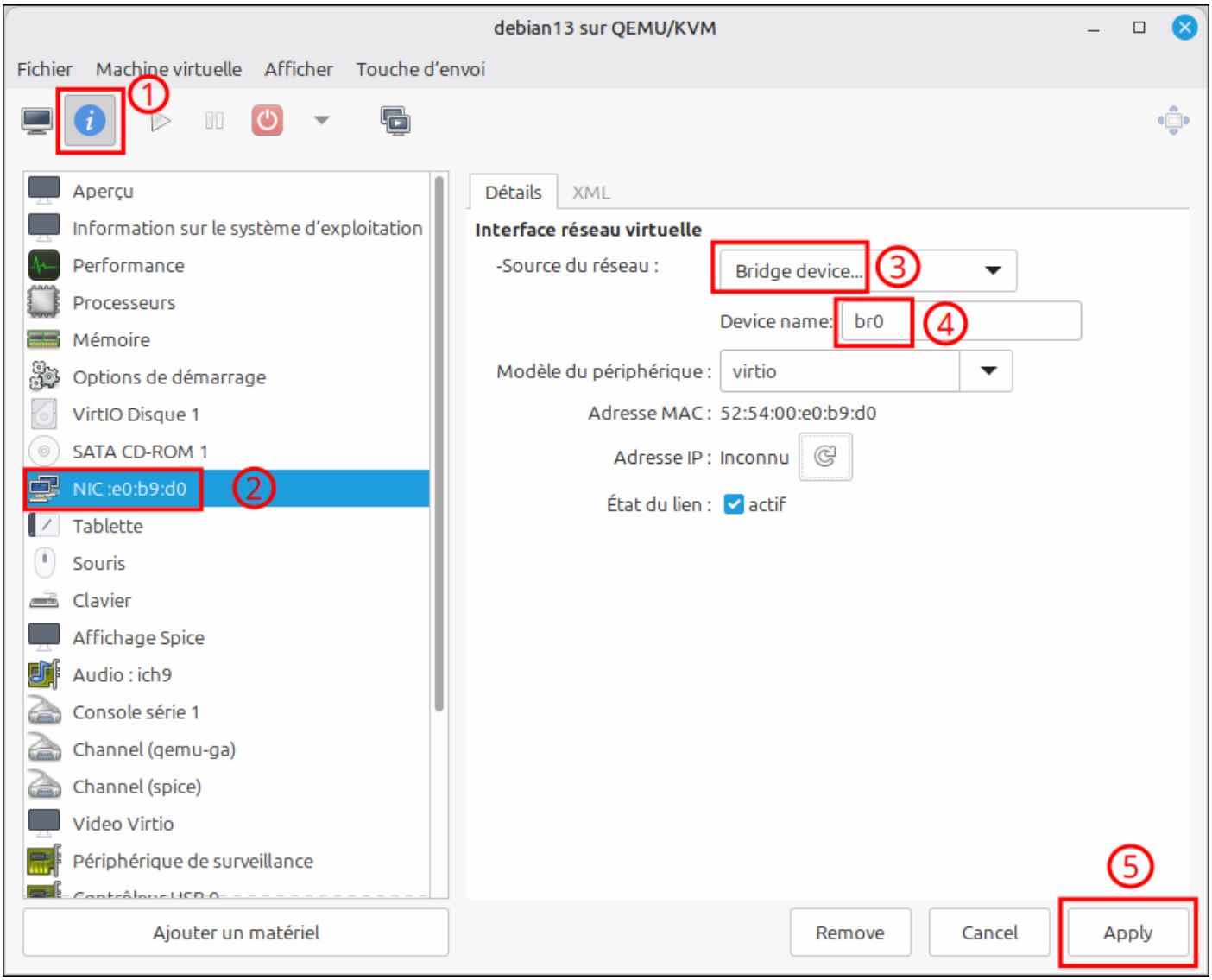
```
sudo nmcli -f bridge con delete br0
```

D- Paramétrage de la connexion réseau de la machine virtuelle

Il faut modifier le type de connexion réseau de la machine virtuelle. Pour cela, dans Virt-Manager, on sélectionne sa machine virtuelle et on va dans le menu "**Édition**" ① puis "**Détails de la machine virtuelle**" ② :



Dans la fenêtre de la machine virtuelle, on clique sur l'icône d'information bleue "i" ①, puis sur la carte réseau (NIC) ② et là comme source du réseau, on choisit "Bridge device..." ③ (à la place du NAT de départ) et en "Device name", on saisit "br0" ④. On n'oublie pas de valider en appuyant sur le bouton "Apply" ⑤ :



On peut maintenant lancer la machine virtuelle et installer le serveur FOG.

Installation de DHCP dans la machine virtuelle sous Debian 13

Remarque : Lorsqu'on installe FOG, normalement le serveur DHCP s'installe également. Mais comme j'ai eu des souci avec ce serveur, j'ai préféré reprendre une installation de zéro. Ça permet également de mieux comprendre les réglages à effectuer.

Comme nous utiliserons FOG sans connexion Internet et sans routeur, il faut que son serveur face office également de serveur DHCP pour pouvoir attribuer des adresses IP aux autres machines du réseau.

Pour cela, il faut taper la commande :

```
sudo apt install isc-dhcp-server
```

Ensuite, il faut éditer le fichier de configuration avec la commande :

```
sudo nano /etc/default/isc-dhcp-server
```

Et renseigner l'interface réseau du bridge (enp1s0 dans mon cas) à la ligne :

```
INTERFACESv4="enp1s0"
```

Ensuite on édite le fichier de configuration avec la commande suivante :

```
sudo nano /etc/dhcp/dhcpd.conf
```

Et on colle la configuration suivante :

```
# DHCP minimal pour FOG PXE BIOS + UEFI
default-lease-time 600;
max-lease-time 7200;
authoritative;

# Détection architecture client
option arch code 93 = unsigned integer 16;

subnet 192.168.1.0 netmask 255.255.255.0 {

    # Plage IP distribuée aux clients PXE
    range 192.168.1.50 192.168.1.200;

    # Passerelle
    option routers 192.168.1.100;

    # Masque réseau
    option subnet-mask 255.255.255.0;

    # DNS (paramètre obligatoire même si inutile pour le PXE)
    option domain-name-servers 1.1.1.1;

    # Serveur TFTP / PXE = FOG VM
    next-server 192.168.1.100;

    # Fichier de boot PXE
    if option arch = 00:07 {
```

```
        filename "ipxe.efi";         # UEFI 64 bits
    } else {
        filename "undionly.kpxe";    # BIOS legacy
    }
}
```

On teste si cette configuration est correcte avec la commande :

```
sudo dhcpd -t
```

Si c'est correct, on peut lancer le serveur DHCP avec les 2 commandes ci-dessous :

```
sudo systemctl enable isc-dhcp-server
sudo systemctl restart isc-dhcp-server
```

Problème potentiel de connexion Internet

Ce serveur DHCP pourrait rentrer en conflit avec un autre serveur DHCP sur le même réseau. Si c'était le cas, voici comment le désactiver temporairement ou de façon "permanente".
Pour le désactiver temporairement (jusqu'au prochain boot où il se réactivera), taper la commande :

```
sudo systemctl stop isc-dhcp-server
```

Pour le désactiver de façon permanente, taper la commande :

```
sudo systemctl disable isc-dhcp-server
```

Installation de FOG dans la machine virtuelle sous Debian 13

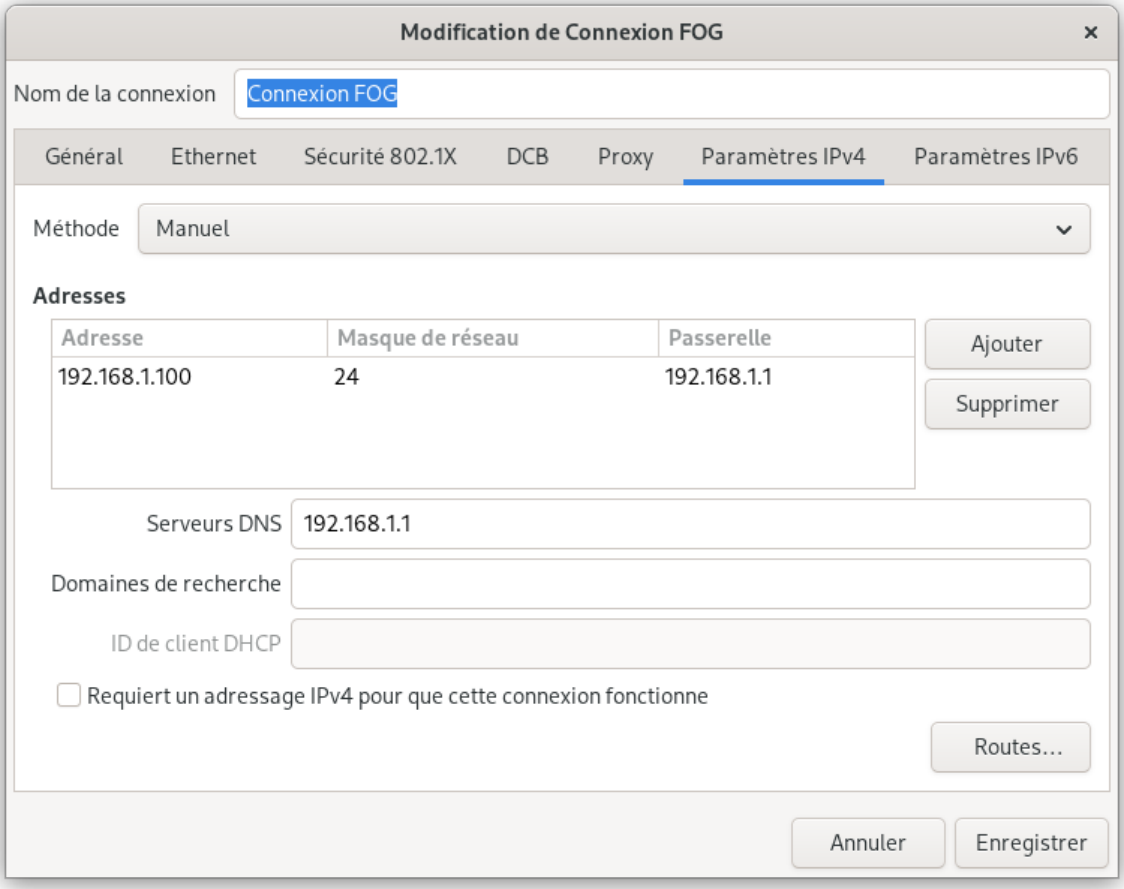
Remarque : pendant toute la préparation de la machine virtuelle, dans Virt-Manager, il faudra paramétrer le réseau en NAT afin d'avoir accès à Internet pour télécharger les programmes notamment. Par contre, il faudra modifier plusieurs fois dans la machine Debian, les paramètres réseaux.

Nous utiliserons FOG sans l'intégrer dans un réseau existant, en utilisation isolée pour faire les installations de Primtux. Pour cela, on peut aller voir cette page (non mise à jour) :

https://wiki.fogproject.org/wiki/index.php/FOG_on_an_Isolated_Network.

Pour faciliter l'installation et permettre l'installation de FOG qui a besoin d'avoir accès à Internet pour récupérer des fichiers, **il faut paramétrer 2 connexions Ethernet dans Debian**. La

première est celle déjà existante et qui permet d'accéder à Internet (on ne la modifie pas), c'est la "Connexion filaire1" (ou "Wired connection 1") qui fonctionnera avec la connexion réseau de type NAT dans Virt-Manager. Par contre, on va se créer une 2e connexion Ethernet spécifique pour l'utilisation de FOG, avec l'application "**Connexion Réseau Avancée**" que j'ai baptisée "Connexion FOG" :



On fixe les réglages suivantes dans l'onglet IPv4 en configuration manuelle avec une IP fixe :
adresse IP : *192.168.1.100* (ou une autre)
masque de sous-réseau : *255.255.255.0* (correspond à 24 en notation CDIR)
passerelle : *192.168.1.1*
Remarque : il serait aussi possible de modifier l'adresse IP du serveur a posteriori, en suivant les indications ici : https://docs.fogproject.org/en/latest/kb/how-tos/change-fog-server-ip-address/

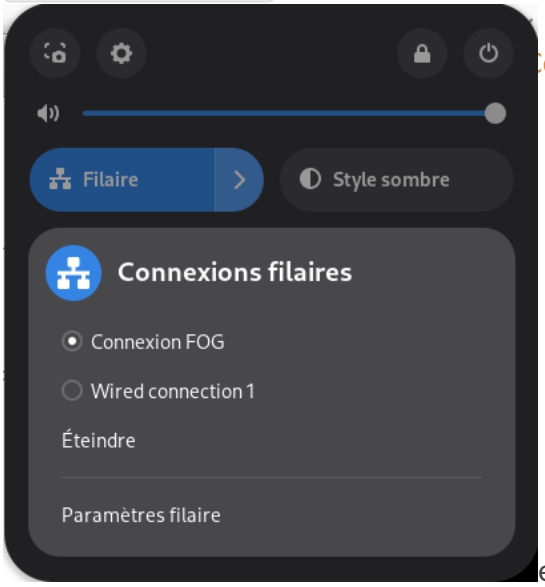
Dans notre machine virtuelle Debian, avec la connexion pour accéder à Internet, on va récupérer l'archive FOG dans Git comme cela est recommandé dans la documentation (https://docs.fogproject.org/en/latest/installation/server/install-fog-server/). Pour cela, on installe git avec les commandes suivantes :

```
sudo -i
apt install git
```

Puis on récupère FOG avec les commandes (toujours en étant root via la commande "sudo -i") :

```
cd /opt
git clone https://github.com/FOGProject/fogproject.git
```

Ensuite on revient dans le terminal et on lance l'installeur avec :
`cd fogproject/bin`



onnexion FOG" :

et on revient dans le terminal pour taper la commande :

`./installfog.sh`

Voici les différentes options choisies lors de l'installation (les choix par défaut sont indiqués entre crochets [ ] et avec une majuscule) :

- Choise: [2] -> "**2**" ou valider directement
- What type of installation would you like to do? [N/s] -> "**n**" pour Normal ou valider directement
- Would you like to change the default network interface from enp0s1? [y/N] -> "**n**" ou valider directement (il nous faut l'adresse 192.168.1.100/24 correspondant à la connexion FOG)
- Would you like to setup a router address for the DHCP server? [Y/n] -> "**y**"
- What is the IP address to be used for the router on the DHCP server? [192.168.1.1]
- Would you like DHCP to handle DNS? [Y/n] -> "**y**"
- What DNS address should DHCP allow? [192.168.1.1]
- Would you like to use the FOG server for DHCP service? [y/N] -> "**y**"
- you like to install the additional language packs ? [y/N] -> "**y**" si on veut l'interface en français
- Would you like to enable secure HTTPS on your FOG server? [y/N] -> "**n**" ou valider directement
- Would you like to change it (hostname debian-FOG)? [y/N] -> "**n**" ou valider directement
- Are you ok with sending this information? [Y/n] -> "**n**" inutile car on ne sera pas connecté au réseau lorsqu'on utilisera FOG.
- On remodifie la connexion Ethernet pour avoir accès à Internet en se mettant sur "Connexion filaire1" (ou "Wired connection 1").
- Are you sure you wish to continue (Y/N) -> "y" si on n'a pas fait d'erreur sinon "n" pour annuler l'installation et relancer le process du début.
  L'installateur se lance alors et va récupérer les fichiers nécessaires sur Internet.

- Lorsqu'on arrive à la création de la database, il faut de nouveau modifier l'adresse IP "Connexion FOG" pour pouvoir accéder à la page : http://192.168.1.100/fog/management et créer la base de données.
- On laisse se finir l'installation.
- Une fois l'installation terminée, on pourra alors se connecter au serveur FOG à cette adresse http://192.168.1.100/fog/management avec les identifiants suivants :
  Username: **fog**
  Password: **password** (avec un clavier AZERTY fonctionnant en QWERTY, il faudra saisir **pqsszord**)
  Remarque : il peut être utile de changer le mot de passe afin qu'on puisse le saisir quelque soit le type de clavier des machines clientes (souvent en QWERTY lorsqu'on démarre), à la façon de "tuxprof" qui fonctionne quel que soit le clavier.
- On peut aller maintenant paramétrer le serveur FOG dans notre langue, en allant sur la configuration de FOG (icône en forme de clé à molette), puis dans l'onglet à gauche "**FOG settings**" ou "**Paramètres de FOG**" (accès direct : http://192.168.1.100/fog/management/index.php?node=about&sub=settings), puis dans la partie "**General Settings**", dans le champ **KEYMAP**, on saisit "**fr**" pour que les claviers des clients correspondent bien au clavier français, et dans le champ "**DEFAULT LOCALE**", on choisit "**Français**" pour avoir l'interface en français. Ne pas oublier de valider en cliquant sur le bouton "Update" au bas de la section.
  On peut aussi en profiter pour aller dans la section "**FOG Boot Settings**" et allonger le temps de boot de FOG qui par défaut est de 3 secondes, ce qui est un peu juste. Ne pas oublier de valider en cliquant sur le bouton "Update" au bas de la section.

REMARQUE : Si on se trompe dans une étape et que l'on veut reprendre de zéro l'installation de FOG, le mieux est de le désinstaller complètement (sans supprimer le dossier git) et de relancer la commande `./installfog.sh`. Plus d'info ici : https://wiki.fogproject.org/wiki/index.php/Uninstall_FOG

Utilisation de FOG

Source utile : https://forums.fogproject.org/topic/14906/configuring-a-standalone-fog-vm-under-ubuntu-20-04

A- Test de FOG

On va tester le serveur FOG pour voir s'il est fonctionnel. Il va falloir vérifier que nous sommes bien avec les réglages suivants :

- Sur la machine virtuelle sous Debian 13 avec le serveur FOG :
  - connexion réseau filaire = "**Connexion FOG**",
  - dans son navigateur Internet, se mettre sur la page "http://192.168.1.100/fog/management" pour vérifier que le serveur FOG est bien lancé.

- Sur l'ordinateur hôte sous Linux Mint :
  - dans Virt-Manager, vérifier que la carte réseau NIC de la VM Debian 13 est en mode bridge avec br0 (pas en NAT),
  - le câble Ethernet est bien branché sur le switch,
  - il peut y avoir une connexion wifi active : pratique pour aller chercher des informations sur Internet, comme cette page ;-) . Attention toutefois à ne pas être sur le même segment de réseau en 192.168.1.xxx.
- Sur l'ordinateur de test (le client) :
  - le câble Ethernet est bien branché sur le switch,
  - booter en PXE

Si on voit le menu FOG qui apparaît sur le client, cela signifie que le serveur est fonctionnel !

B- Récupération de l'image Primtux à déployer

FOG marche dans les 2 sens, c'est-à-dire qu'il peut déployer une image mais aussi "l'aspirer" (comme CloneZilla). On peut donc préparer un ordinateur sous Primtux avec une seule partition (de cette façon, il sera possible de déployer l'image sur un disque de capacité plus petite à condition que les données ne dépassent pas sa capacité) et ensuite récupérer son image qui servira de modèle (master). Pour Primtux 9, j'en profite pour y installer DWService et le gestionnaire de presse-papier CopyQ (plus complet que Clipman), ... (voir cette page :

https://lofurol.fr/joomla/enseignement/391-primtux-installation-de-primtux-9).

Source : https://docs.fogproject.org/en/latest/kb/how-tos/capture-an-image/

- On démarre l'ordinateur servant de modèle Primtux en boot PXE (en général, touche F12 au démarrage) et quand le menu FOG apparaît on a le choix entre 2 options :
  - soit on fait une "**Quick registration and inventory**" afin d'enregistrer directement cette machine dans FOG.
  - soit on fait un enregistrement plus complet (qui permet notamment de donner un nom plus identifiable à la machine) via l'option "**Perform Full Host Registration and Inventory**".
- Une fois l'enregistrement fait, l'ordinateur redémarre sous Primtux et on l'éteint.
- **Dans FOG, on va dans le menu "Images"** (icône en forme d'image justement), et on crée une nouvelle image. On lui donne un nom comme Primtux9 et sans rien modifier d'autre, on l'enregistre. Remarque : on choisit une seule partition pour l'image car de cette façon, il sera possible de déployer l'image sur un disque de capacité plus petite à condition que les données ne dépassent pas sa capacité.
- Dans FOG, on revient sur la machine enregistrée auparavant (elle a pour nom son adresse MAC) et on la modifie en cliquant sur son nom. Dans le champ "Image machine" on lui associe l'image Primtux créée précédemment. On n'oublie pas de sauvegarder la modification.
- Toujours dans la modification de la machine, on clique en haut sur l'**onglet "Tâches basiques"** puis sur l'**icône "Capture"**. On laisse les paramétrages par défaut et on crée

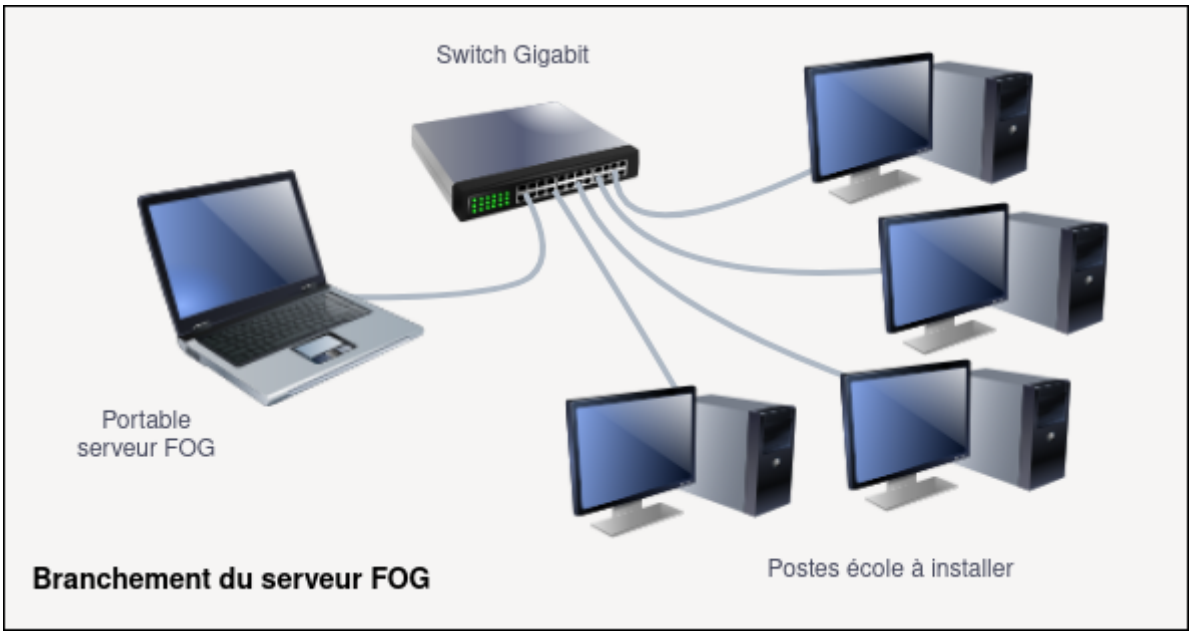
la tâche **en cliquant sur le bouton bleu "Tâche"** -> Un message vert indique que la tâche a été créée avec succès. On peut vérifier l'existence de cette tâche en cliquant sur l'icône "Tâches" représentant un empilement horizontal de dossiers.
- On redémarre alors l'ordinateur modèle Primtux en boot PXE. Cette fois-ci, le menu FOG n'apparaîtra pas mais un terminal indiquant des opérations en cours avec une dernière fenêtre bleue à la Clonezilla avec 2 barres de progression rouges. Sur le serveur FOG, on voit la tâche s'effectuer avec petite roue en mouvement. Au bout d'un certain temps (d'où l'intérêt d'avoir un switch réseau 1 Gigabit), l'image est obtenue (autour de 25 Go).

REMARQUE : il est possible de déployer une image universelle indépendante du matériel sur lequel on veut l'installer. On appelle ce type d'image, une **golden image**. Piste à creuser :

https://forums.fogproject.org/topic/7391/deploying-a-single-golden-image-to-different-hardware-with-fog

C- Déploiement de l'image Primtux

Source : https://docs.fogproject.org/en/latest/kb/how-tos/deploy-an-image/



Tout se fait uniquement à partir de l'ordinateur client (pas besoin d'agir sur le serveur FOG), ce qui est très rapide.
Sur les ordinateurs à installer :

- on démarre en boot PXE et on procède à leur "**Perform Full Host Registration and Inventory**" qui permet de leur assigner un nom (nom dans FOG mais qu'on peut aussi donner à la machine via un script, voir ci-dessous) et surtout une image ID (dans notre cas celle créée auparavant de Primtux et qui est la numéro 1).
- On laisse tous les autres champs vides (inutile de renseigner "Primary user" ou les "Tags").
- À la question "Deploy image now ?", choisir Y puis renseigner les mots de passe FOG (utilisateur = fog ; mot de passe =password).

- Lorsque l'ordinateur redémarre, il faut le faire en boot PXE (touche F12) et alors l'image Primtux s'installera automatiquement.

Scripts de post-installation

Il est possible de lancer des scripts de post-installation pour par exemple renommer les machines à partir de leur nom enregistré dans Fog, redimensionner une partition, ...
Ces scripts doivent être placés dans le répertoire du serveur FOG /images/postdownloadscripts et se lanceront une fois que l'image aura été déployée.

A- Script pour renommer la machine sous Primtux :

Source : https://memo-linux.com/fog-1-2-script-post-installation-pour-modifier-le-nom-dune-machine-linux-clonee/

Pour cela, il faut modifier les fichiers /etc/hostname et /etc/hosts. On va pouvoir le faire en créant un script dans le dossier /images/postdownloadscripts. Voici les commandes à saisir dans le terminal du serveur FOG :
`cd /images/postdownloadscripts`

Ensuite on crée et on édite un fichier "renommage.sh" avec la commande :
`sudo nano renommage.sh`

Et on colle les lignes de code suivantes :

```
#! /bin/bash
## Script de post-installation "renommage.sh" pour FOG
## Permettant de renommer une machine après l'installation de Primtux
## créé par Thierry Munoz le 30/01/2024 et mis à jour le 18/03/2026 pour Primtux 9

##création d'un répertoire temporaire /partprimtux
mkdir /partprimtux
##montage temporaire de la partition système du Linux
## ATTENTION : il faut indiquer la partition sur laquelle est installé Primtux !!!
## ici il s'agit de la partition sda3 (sinon ça ne marchera pas)
mount /dev/sda3 /partprimtux

##changement du nom host de la machine définie dans l'interface web
echo $hostname > /partprimtux/etc/hostname
##changement du nom "primtux" dans /etc/hosts qui se trouve sur la 2e ligne
sed -i "s/primtux/$hostname/" /partprimtux/etc/hosts
```

```
##Démontage de la partition système
umount /partprimtux
##Suppression du répertoire temporaire
rmdir /partprimtux
```

On sauvegarde (CTRL + O) et on ferme le fichier (CTRL + X) puis on le rend exécutable avec la commande suivante :
`sudo chmod +x renommage.sh`

Ensuite, on édite et modifie le fichier fog.postdownload qui indique à FOG quel(s) script(s) lancer après la copie de l'image Primtux sur le client :
`sudo nano fog.postdownload`

Et on colle cette ligne en veillant bien à laisser un espace derrière le point en début de ligne :

```
. ${postdownpath}renommage.sh
```

On sauvegarde (CTRL + O) et on ferme (CTRL+X) l'éditeur Nano.

Maintenant le script modifiera automatiquement les fichiers /etc/hostname et /etc/hosts avec le nom de la machine.

B- Script pour redimensionner la partition Primtux :

Normalement cette partie est inutile car FOG redimensionne automatiquement les partitions. Pour cela il faut choisir l'option "**Single Disk - Resizable**" lors de la capture.

PARTIE CI-DESSOUS INUTILE MAIS LAISSÉE POUR GARDER UNE TRACE

Le redimensionnement automatique de la partition finale ne fonctionne que pour Windows (en fait FOG a été développé initialement pour Windows) et du coup après avoir copié l'image Primtux, la totalité de la partition du disque dur n'est pas utilisée. On peut la modifier avec GParted mais l'idéal serait de l'agrandir automatiquement avec un script de post-installation.

Autre piste : dans le dossier /images/nom_image, il y a pour les images "resizables", des fichiers d1.minimum.partitions, d1.fixed_size_partitions et d1.partitions.

Voici un script à tester (pour des disques HDD, disques SSD "non sda" ?) :

```
#! /bin/bash#
```

```
#################################################################
# Script de redimensionnement de partition PrimTux pour FOG #
# philippe.dpt35@yahoo.fr                                       #
# La parition de PrimTux doit être la dernière partition    #
# du disque ; elle peut être dans une partition étendue     #
# La variable periphdisk est à modifier si le disque        #
# cible n'est pas sur /dev/sda                              #
#################################################################


periphdisk="sda"

# Récupère le N° de la partition étendue du disque s'il y en une
extendpart=$(LC_ALL=C fdisk -l |  awk -v disk="/dev/$periphdisk" '$0 ~ disk && /Extended/
{sub("/dev/sd[a-z]","");sub("\n"," ");print $1}')

# Récupère le N° de la dernière partition du disque
lastpart=$(lsblk -o "name" -n |  awk -v disk=$periphdisk '/└─/ && $0 ~ disk {sub("└─sd[a-
z]","");print $1}')

if [ -n "$extendpart" ]; then
  parted "/dev/$periphdisk" -- resizepart $extendpart 100%
fi
if [ -n "$lastpart" ]; then
  parted "/dev/$periphdisk" -- resizepart $lastpart 100%
  devpart=$(echo "/dev/$periphdisk$lastpart")
  sleep 1
  resize2fs "$devpart"
fi
exit 0
```

parted /dev/sda resizepart 2 100%

resize2fs /dev/sda5

Ressources utiles pour les scripts

- Pour vérifier les scripts shell : https://www.shellcheck.net/
- https://forums.fogproject.org/topic/7740/the-magical-mystical-fog-post-download-script
- https://forums.fogproject.org/topic/4510/scripts-customization-of-ubuntu-fog-service-for-ubuntu

---

Sources utiles

- Site avec un autre tuto détaillé : https://chrtophe.developpez.com/tutoriels/deploiement-fogproject/
- Page dont est issue cette documentation : https://lofurol.fr/joomla/logiciels-libres/339-installation-dun-serveur-fog-dans-virtualbox
- Pour mettre à jour FOG : https://forums.fogproject.org/post/151675

Revision #68
Created 16 February 2026 13:49:10 by Thierry
Updated 5 November 2026 16:41:23 by Thierry