Installation d’un serveur FOG :

apache2 bc build-essential cpp curl g++ gawk gcc genisoimage git gzip htmldoc isolinux lftp libapache2-mod-php libc6 libcurl4 liblzma-dev m4 mariadb-client mariadb-server net-tools nfs-kernel-server openssh-server php php-bcmath php-cli php-curl php-fpm php-gd php-gettext php-json php-ldap php-mbstring php-mysql php-mysqlnd sysv-rc-conf tar tftpd-hpa tftp-hpa unzip vsftpd wget xinetd zlib1g

Serveur FOG : ip > 192.168.0.15

Machine Test : Callée sur le DHCP du réseau

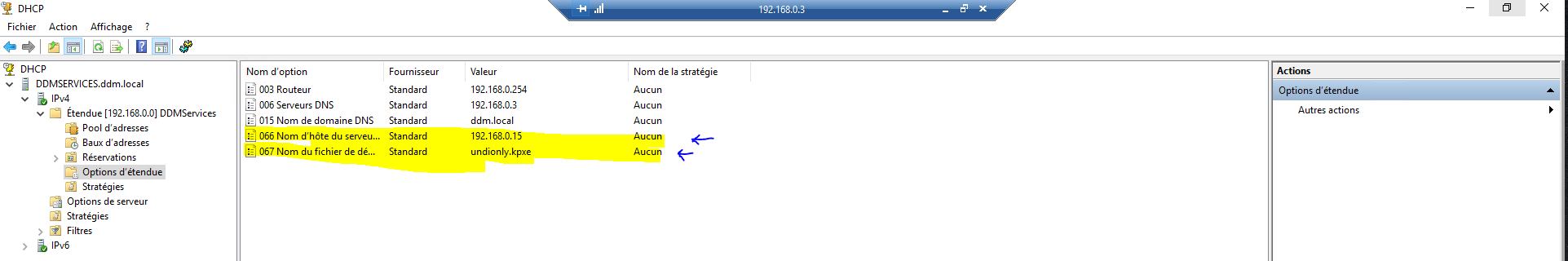
🡾 Attention : le mot de passe saisie lors de l’installation d’Ubuntu est en qwerty.

Le serveur Fog s’installe sur une Debian. On télécharge le « projet FOG » ( fogproject.org)

* Décompression
* Installation

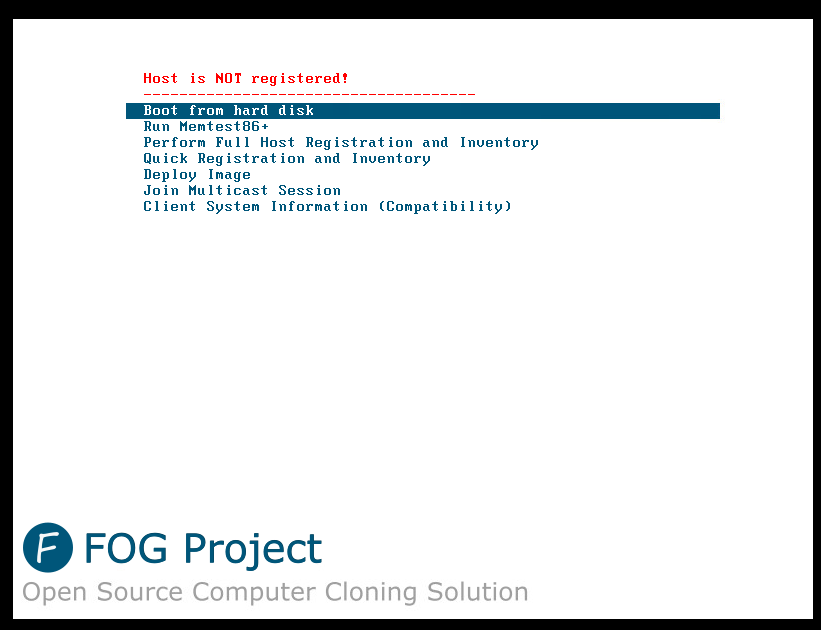
Sur le serveur DHCP :

Ajout sur l’étendue des options 066 (correspondant au serveur fog) et 067 (avec comme paramètre undionly.kpxe)



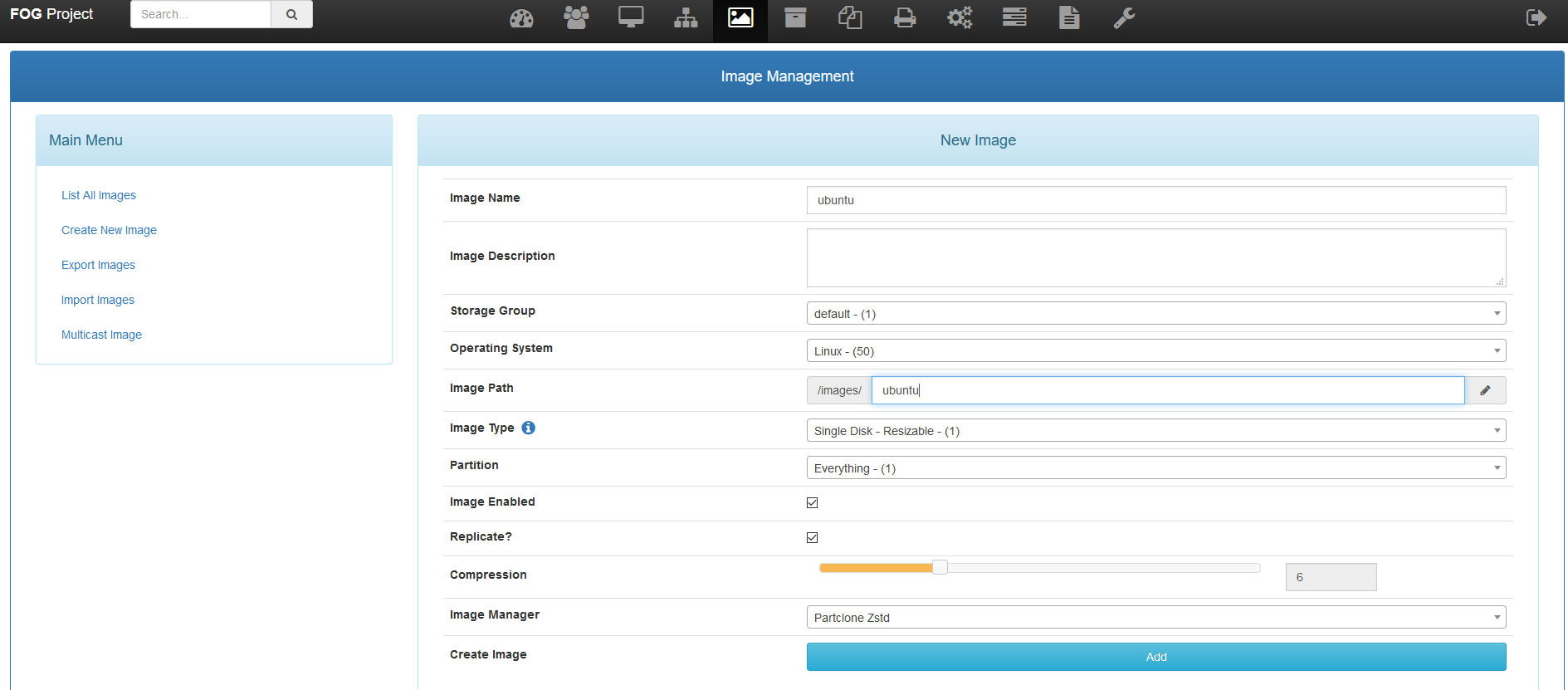
1ere étape :

Après installation de la machine Ubuntu-Test, on la démarre en pxe (F12 au démarrage), puis on arrive ici :

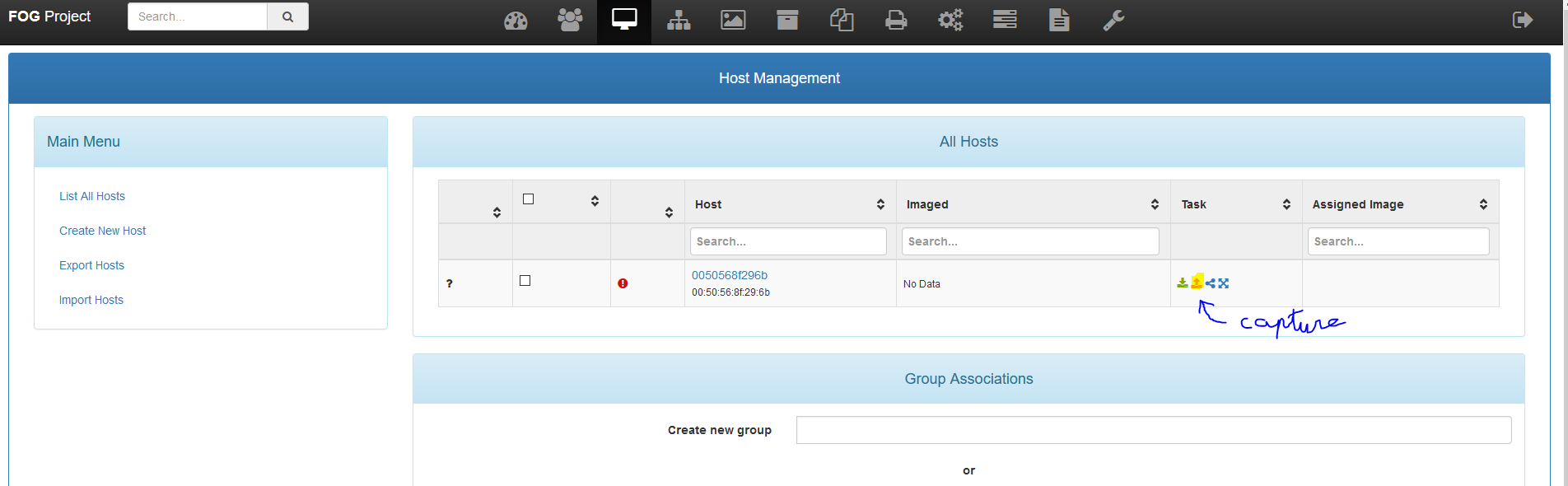


*Choisir Quick Registration and inventory va permettre de repertorier la machine /le fog*

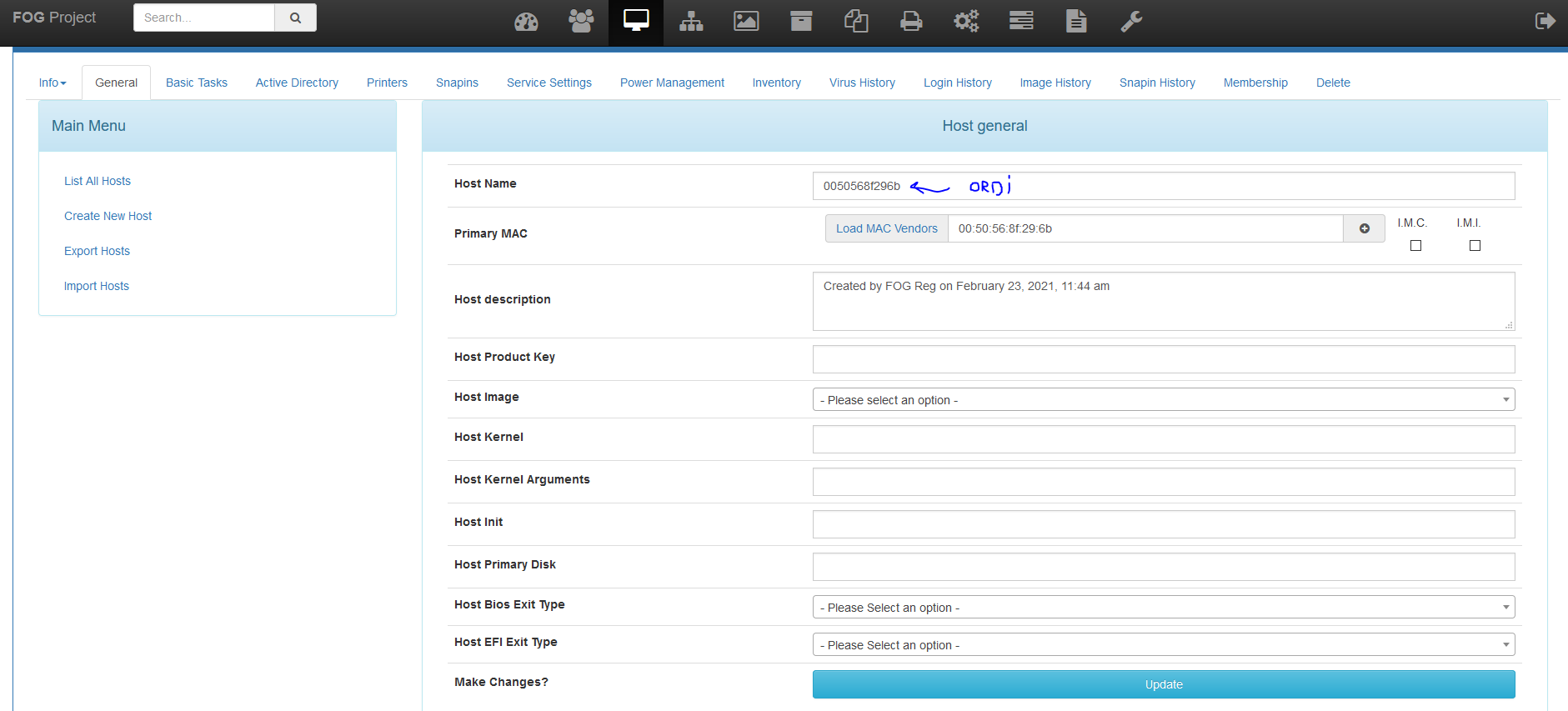
Dans Image Management :



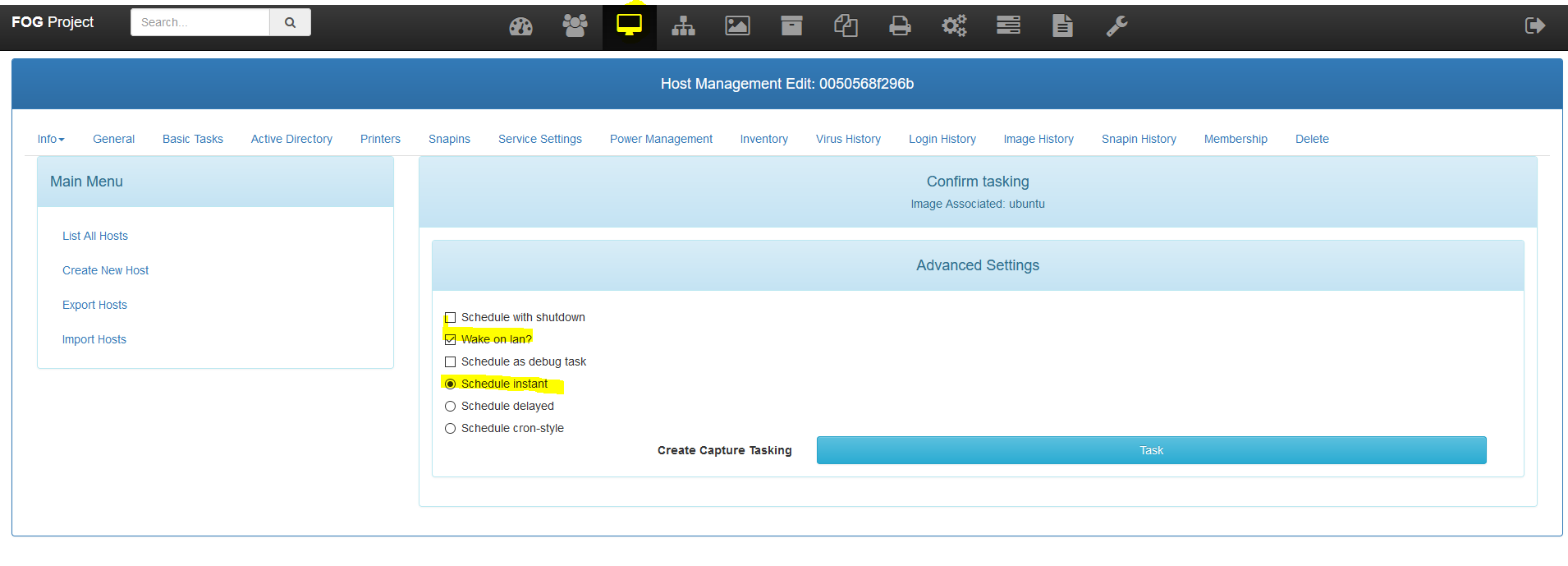
*On créer le container qui contiendra l’image – ici l’image portera le nom d’Ubuntu*



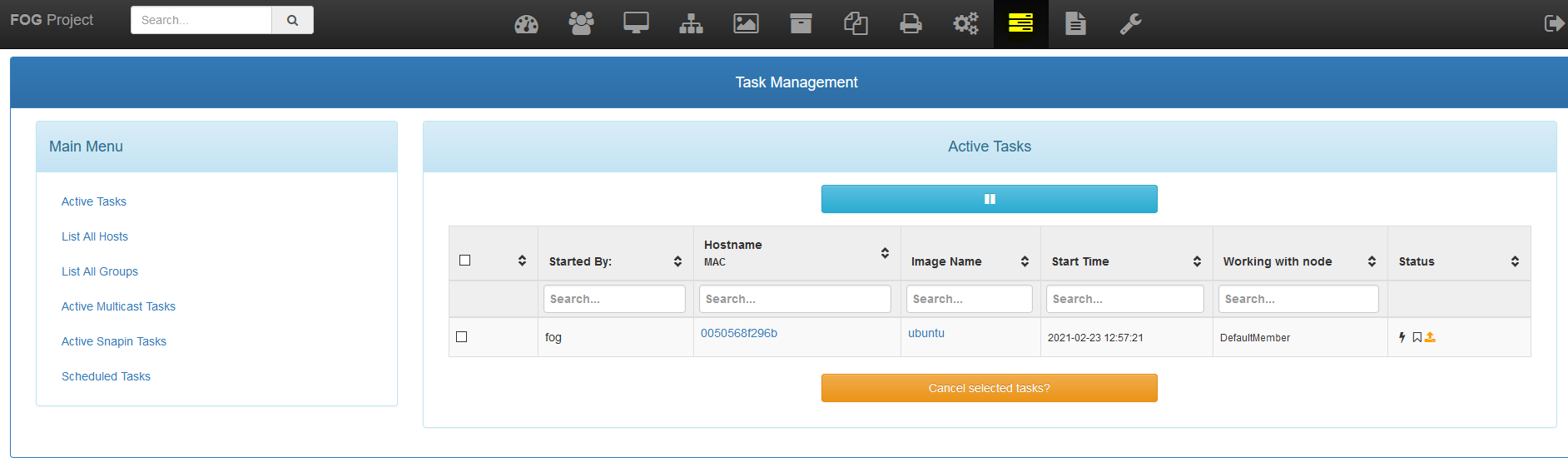
*Ici on voit bien que fog a bien enregistré les caractéristiques de notre Ubuntu – On va donc capturer l’image de notre Ubuntu.*



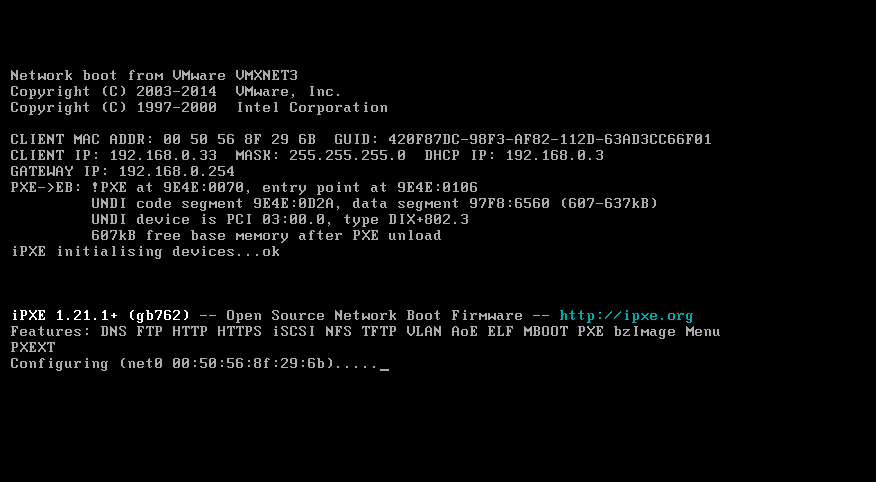
*Ici on indique l’ordinateur depuis lequel nous voulons aspirer l’image. Dans Host Image, on le cale sur la coquille crée précédemment, Ubuntu.*



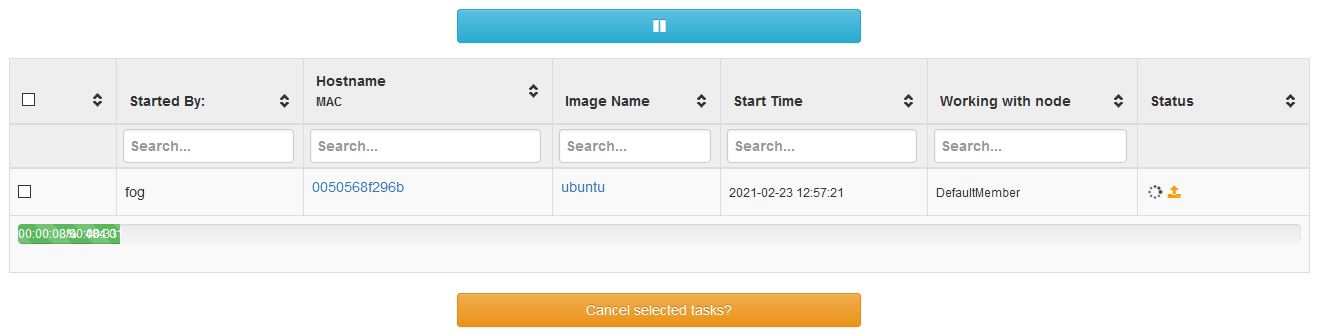
*On prévoit la tache immédiatement avec un wake on lan pour booter l’ordi s’il n’est pas allumé.*



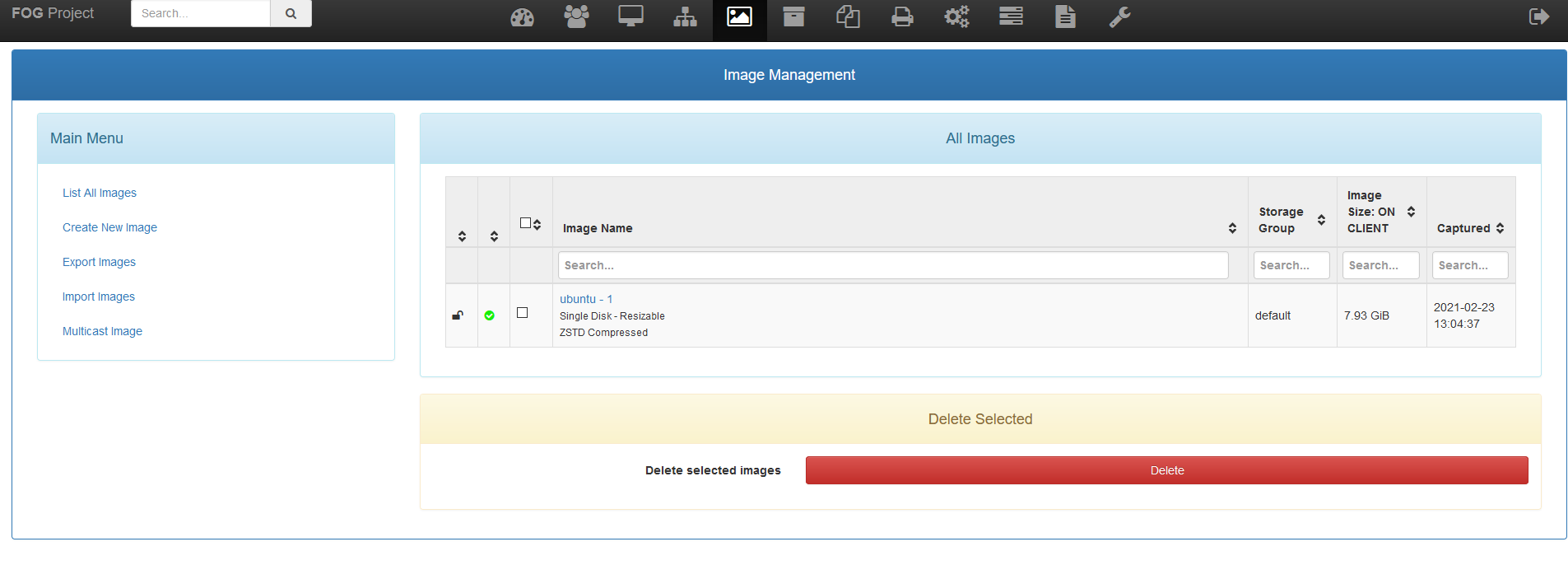
*On voit dans les taches que notre tache est prete à etre effectuée.*



*C’est parti, notre ubuntu boot après un F12 au démarrage.*



*Sur notre interface web, on voit que notre le status de notre tache à changé.*



*Terminé, notre Ubuntu a été aspiré par notre fog, l’image est capturée.*