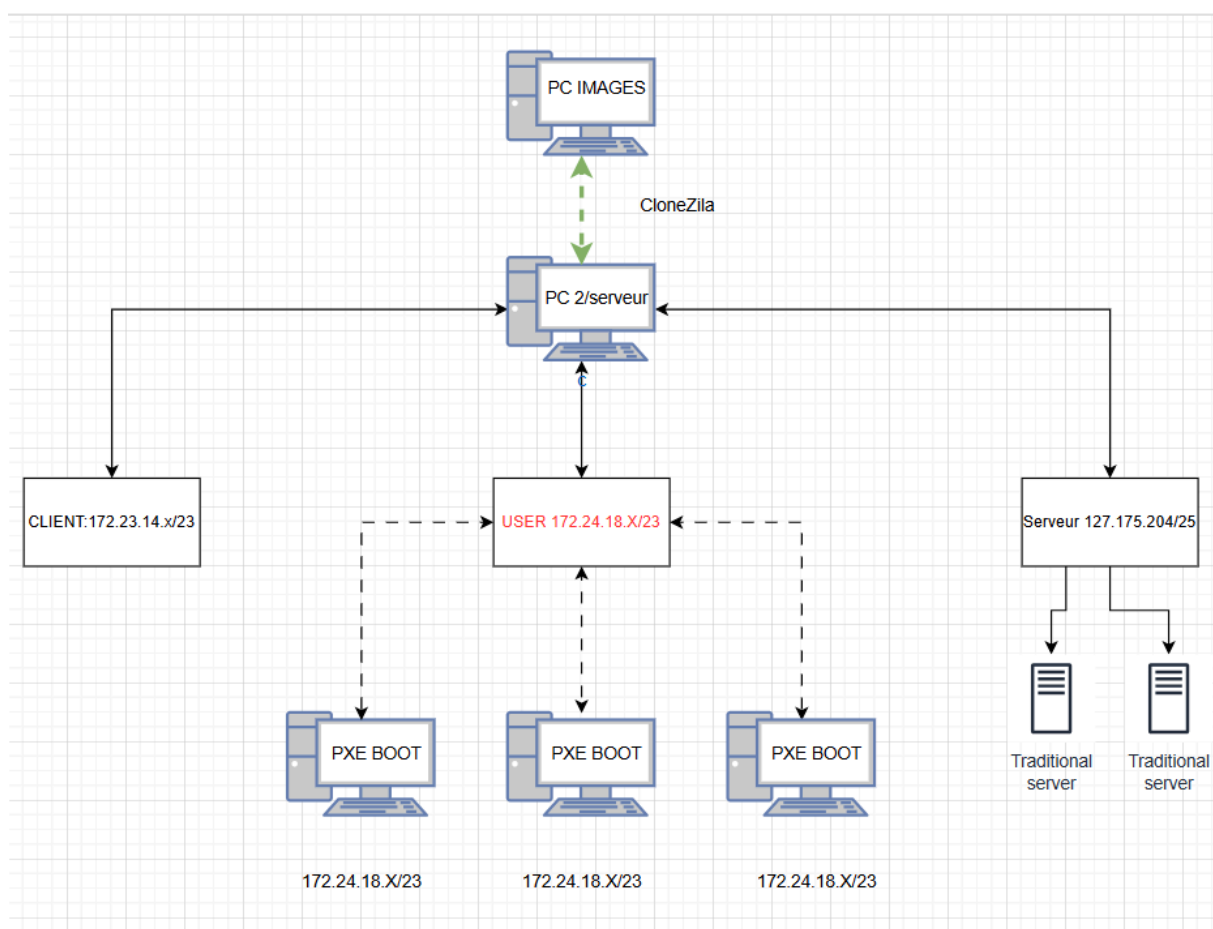


Documentation

! Mission : Déployer un snapshot d'un PC (système d'exploitation, logiciels,) sur des PC vides au sein d'un réseau entreprise.





Pré requis VirtualBox :

- 1 serveur
- 1 une image
- 1 pc vide

▼ 1.1 Configuration du serveur Debian 12 (VM1)

Tout d'abord sur virtual box dans les paramètres, veuillez activer dans le réseau l'adaptateur 1 en nat et le 2 en interne pour que toutes les machines communiquent entre elles





ensuite la mise à jour des paquets :

```
sudo apt update && sudo apt upgrade -y
```

et l'installation de tout les services nécessaires :

```
sudo apt install isc-dhcp-server tftpd-hpa nfs-kernel-server nginx syslinux pxelir
```

| Paquet | Rôle | Sert à... |
|--------------------------------|--|--|
| <code>isc-dhcp-server</code> |  Serveur DHCP | Fournit automatiquement une adresse IP aux machines clientes (comme VM2/VM4) et leur indique où booter en PXE (via options comme <code>next-server</code> et <code>filename</code>). |
| <code>tftpd-hpa</code> |  Serveur TFTP | Transfère les fichiers de boot PXE (comme <code>pxelinux.0</code> , <code>initrd</code> , <code>vmlinuz</code>) au client via le protocole TFTP, utilisé par le BIOS/UEFI. |
| <code>nfs-kernel-server</code> |  Serveur NFS | Permet aux clients PXE (comme Clonezilla Live) d' accéder à distance aux images de disque stockées sur le serveur (<code>/srv/nfs/clonezilla_images</code>). |

| | | |
|-------------------------|---|--|
| <code>nginx</code> |  Serveur web léger (facultatif mais utile) | Sert pour afficher les images disponibles via un navigateur ou partager des fichiers, logs, scripts, etc. via HTTP. |
| <code>syslinux</code> |  Outils de boot | Contient les fichiers nécessaires au boot PXE : <code>pxelinux.0</code> , <code>ldlinux.c32</code> , etc., à copier dans <code>/srv/tftp</code> . |
| <code>pxelinux</code> |  Chargeur PXE | Partie de <code>syslinux</code> dédiée au démarrage réseau ; permet d'afficher un menu PXE et charger Clonezilla automatiquement. |
| <code>clonezilla</code> |  Outil de clonage | Fournit l'outil de sauvegarde/restauration (<code>ocs-sr</code> , etc.) – pas forcément utile sur le serveur, mais pratique si tu veux faire des scripts manuels. |

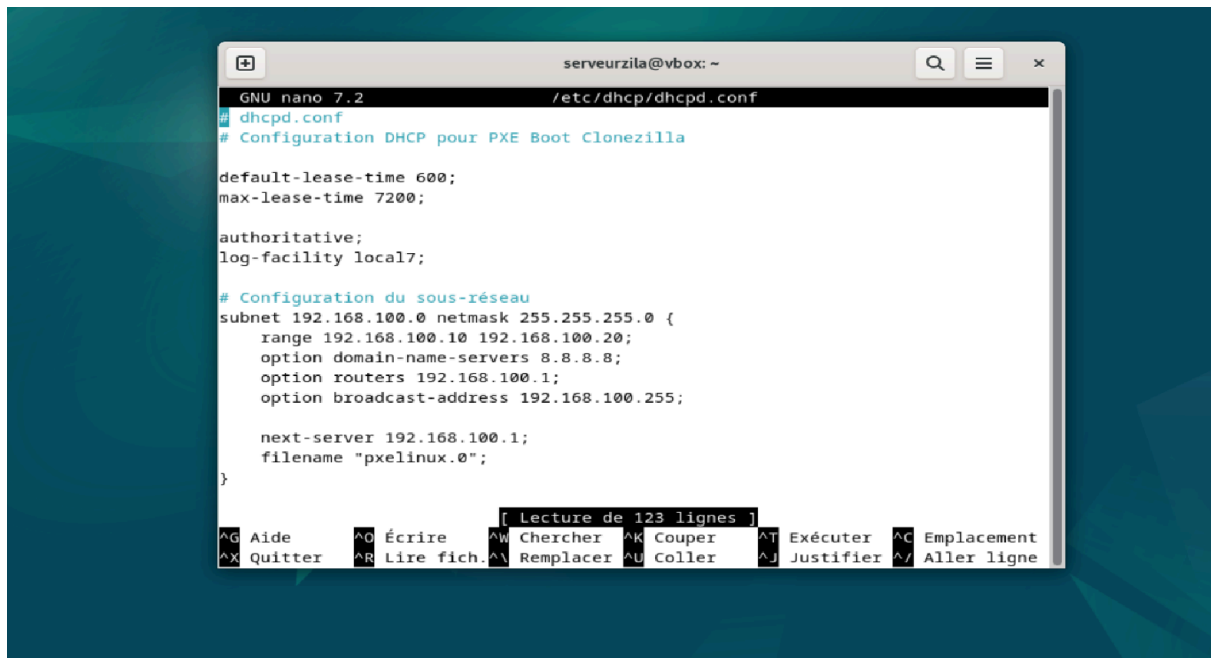
Les paquets si dessus servent pour faire :

- un serveur DHCP
- un serveur TFTP
- un serveur NFS
- les fichiers Clonezilla nécessaires

1.2 Configuration DHCP sur VM1

Tout d'abord il faut configurer le fichier `dhcpd.conf`

```
sudo nano /etc/dhcp/dhcpd.conf
#Recopier la configuration suivante.
```



```
GNU nano 7.2 /etc/dhcp/dhcpd.conf
# dhcpd.conf
# Configuration DHCP pour PXE Boot Clonezilla

default-lease-time 600;
max-lease-time 7200;

authoritative;
log-facility local7;

# Configuration du sous-réseau
subnet 192.168.100.0 netmask 255.255.255.0 {
    range 192.168.100.10 192.168.100.20;
    option domain-name-servers 8.8.8.8;
    option routers 192.168.100.1;
    option broadcast-address 192.168.100.255;

    next-server 192.168.100.1;
    filename "pxelinux.0";
}

[ Lecture de 123 lignes ]
^G Aide      ^O Écrire   ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^M Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne
```

Dans

`/etc/default/isc-dhcp-server`, spécifie l'interface :



Pour être sûr du nom de l'interface vous pouvez faire un :

```
ip a
```

Tu verras probablement :

- `enp0s3` → NAT (Internet)
- `enp0s8` → Interne (PXE, IP statique)

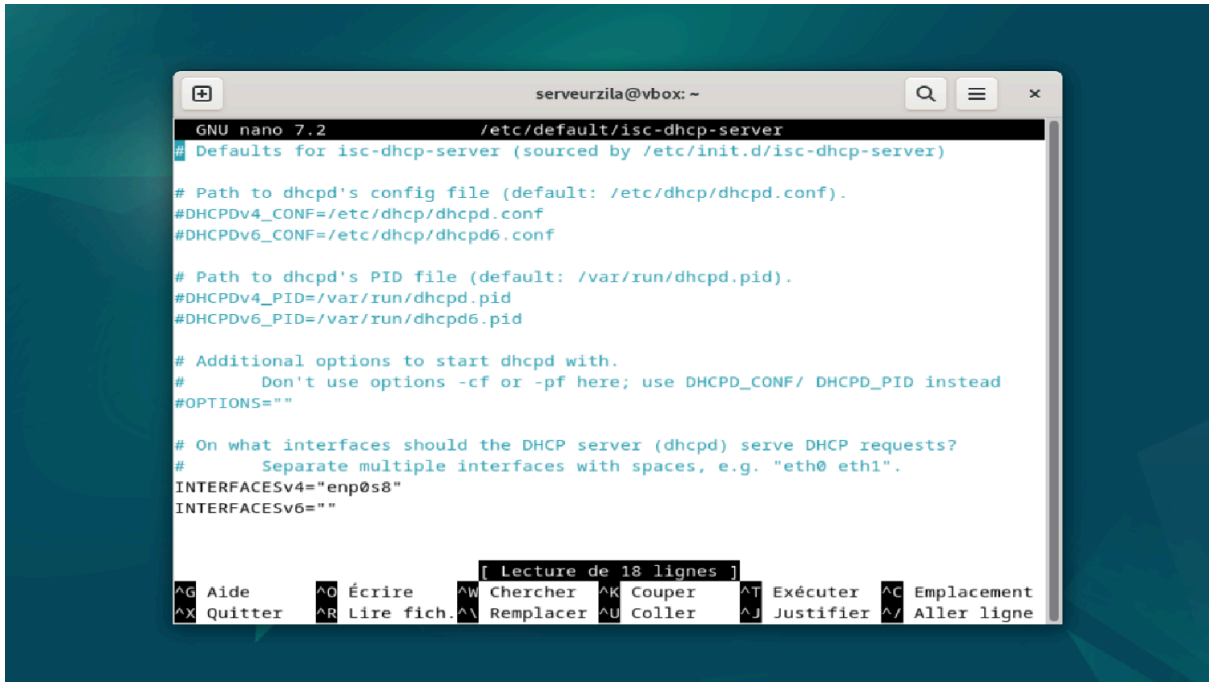
Fichier de config réseau statiq ENPOS8 : `/etc/network/interfaces`

```
sudo nano /etc/network/interfaces.d/pxe.cfg
auto enp0s8
iface enp0s8 inet static
```

```
address 192.168.100.1
netmask 255.255.255.0
sudo systemctl restart networking
```

 **Fichier de config réseau :** `/etc/network/interfaces` ou Netplan

Si tu utilises systemd (par défaut sous Debian 12), crée une config statique pour l'interface interne :



```
serveurzila@vbox: ~
GNU nano 7.2 /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDV4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDV6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDV4_PID=/var/run/dhcpd.pid
#DHCPDV6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s8"
INTERFACESv6=""

[ Lecture de 18 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper     ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich ^\ Remplacer ^U Coller    ^J Justifier ^_ Aller ligne
```

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

Redémarrer ensuite les services DHDHCP :

1.3 Configuration du serveur TFTP

TFTP utilise `pxelinux.0` , les fichiers doivent être dans `/srv/tftp/` .

Crée les dossiers nécessaires :

```
sudo mkdir -p /srv/tftp/pxelinux.cfg
sudo cp /usr/lib/PXELINUX/pxelinux.0 /srv/tftp/
```

```
sudo cp /usr/lib/syslinux/modules/bios/ldlinux.c32 /srv/tftp/
```

Fichier de configuration PXE par défaut :

[/srv/tftp/pxelinux.cfg/default](#)

```
DEFAULT clonezilla
LABEL clonezilla
  KERNEL clonezilla/live/vmlinuz
  APPEND initrd=clonezilla/live/initrd.img boot=live config noswap edd=on no
modeset ocs_live_run="ocs-live-general" ocs_live_batch="yes" ocs_live_mod
e="restore" ocs_srv=192.168.100.1 ocs_img="your_image_name" keyboard-lay
outs=fr locales=fr_FR.UTF-8 vga=788 fetch=tftp://192.168.100.1/clonezilla/live/
filesystem.squashfs
```

1.4 Préparer Clonezilla (image live + image à restaurer)

Clonezilla live ISO contient tout ce qu'il faut.



Télécharger l'iso

```
wget https://sourceforge.net/projects/clonezilla/files/clonezilla_live_stable
```

1.4.1 Monte Clonezilla live ISO :

```
mkdir /mnt/clonezilla
mount -o loop clonezilla-live.iso /mnt/clonezilla
#Si tu ne déplaces pas ton téléchargement dans le dossier tu auras une erreur
```

1.4.2 Copie des fichiers nécessaires :

```
mkdir -p /srv/tftp/clonezilla/live
cp /mnt/clonezilla/live/vmlinuz /srv/tftp/clonezilla/live/
cp /mnt/clonezilla/live/initrd.img /srv/tftp/clonezilla/live/
cp /mnt/clonezilla/live/filesystem.squashfs /srv/tftp/clonezilla/live/
umount /mnt/clonezilla
```

6. Serveur NFS pour partager l'image

Clonezilla aura besoin d'un partage NFS pour accéder à l'image à restaurer.

1.4.3 Crée un dossier pour stocker l'image :

```
sudo mkdir -p /srv/nfs/clonezilla_images  
sudo chown -R nobody:nogroup /srv/nfs
```

1.5 Déclare le partage NFS :

Ajoute à `/etc/exports` :

```
/srv/nfs/clonezilla_images 192.168.100.0/24(rw,sync,no_subtree_check)
```

Recharge NFS :

```
sudo exportfs -a  
sudo systemctl restart nfs-kernel-server
```



2. Configuration de la VM 2 (image source)

Installer un système d'exploitation (debian 12) par exemple.

Vous pouvez y ajouter quelques applications si nécessaires, il est possible de modifier l'image plus tard donc si vous ne le faites pas maintenant ce n'est pas très grave.

On se retrouve ensuite dans les paramètres de la VM pour y ajouter l'ISO clonezilla_live dans stockage.

Démarrer la VM.

. Démarrer VM2 avec Clonezilla Live (en ISO)

- Démarre sur **Clonezilla Live**
- Choisis l'option "**device-image**" → "**nfs_server**".

2. Monter le serveur NFS/

- Choix de Eth1 → Adressage IP static, ex : `192.168.100.2`
- Adresse du serveur NFS : `192.168.100.1`
- Dossier à monter : `/srv/nfs/clonezilla_images`
- Monté localement dans Clonezilla à : `/home/partimag`

3. Sauvegarder le disque complet de VM2

- Choisis `savedisk` ou `saveparts` en fonction de ce que tu veux
- Possibilité de choisir `zstd` pour la compression, pas besoin de couper les fichiers
- Donne un **nouveau nom d'image** (ex : `vm2_backup`)

4. L'image mise à jour est maintenant prête

Elle est stockée sur VM1 dans `/srv/nfs/clonezilla_images/vm3_updated` .



Si vous êtes amenés à faire une autre sauvegarde car vous avez changé des logiciels et que vous voulez mettre à jour il suffit de reproduire ses consignes mais donner un nom différent à la sauvegarde sinon cela écrasera la suivante.

Pour vérifier que tout a bien été copié vous pouvez aller sur le serveur et faire :

```
ls /srv/nfs/clonezilla_images
#Cela devrait rendre
Vm2_Backup
```




3. Configuration de la VM 3 (Copie) PXE BOOT

Création de la VM3

1. Va dans **Paramètres** de la VM3 :

- **Système > Carte mère > Ordre de démarrage**

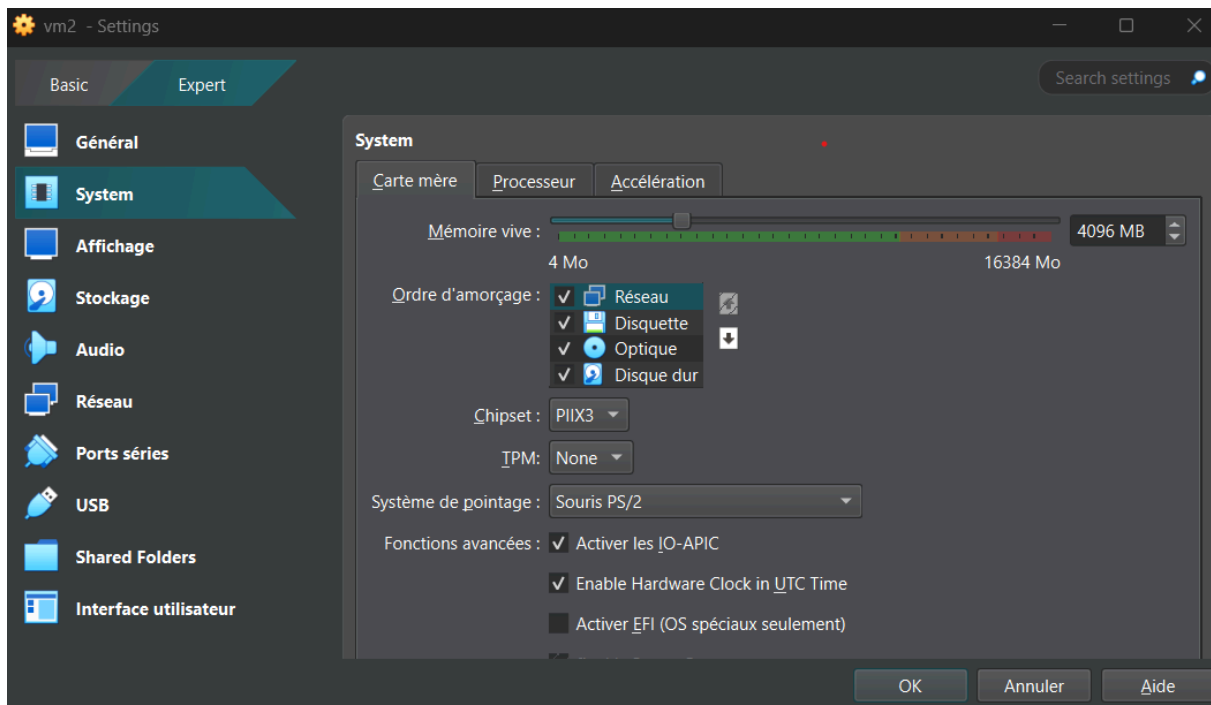
- ☒ Coche **Réseau**
-  Mets-le en **premier dans la liste**
- ☒ Décoche "Disquette" si elle est là

2. Ensuite, dans l'onglet **Réseau** :

- **Adaptateur 1** : Activé
- **Attacher à** : Réseau interne (même nom que VM1, ex : `clonezilla-net`)
- **Type** : Intel PRO/1000 MT Desktop (82540EM)
- ☒ Coche "Câble connecté" en bas

3. Stockage :

- Assure toi d'avoir assez d'espace de stockage pour accueillir la VM source.



Menu PXE

```

iPXE (PCI E2:00.0) starting execution...ok
iPXE initialising devices...ok

iPXE 1.21.1 -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS TFTP PXE PXEXT

net0: 08:00:27:85:4a:e8 using 82540em on 0000:00:03.0 (open)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Down (http://ipxe.org/38086101)]
Waiting for link-up on net0..... ok
Configuring (net0 08:00:27:85:4a:e8)..... ok
net0: 192.168.100.15/255.255.255.0 gw 192.168.100.1
Next server: 192.168.100.1
Filename: pxelinux.0
tftp://192.168.100.1/pxelinux.0... ok
pxelinux.0 : 42430 bytes [PXE-NBP]

PXELINUX 6.04 PXE 20200816 Copyright (C) 1994-2015 H. Peter Anvin et al
Loading clonezilla/live/vmlinuz...

```

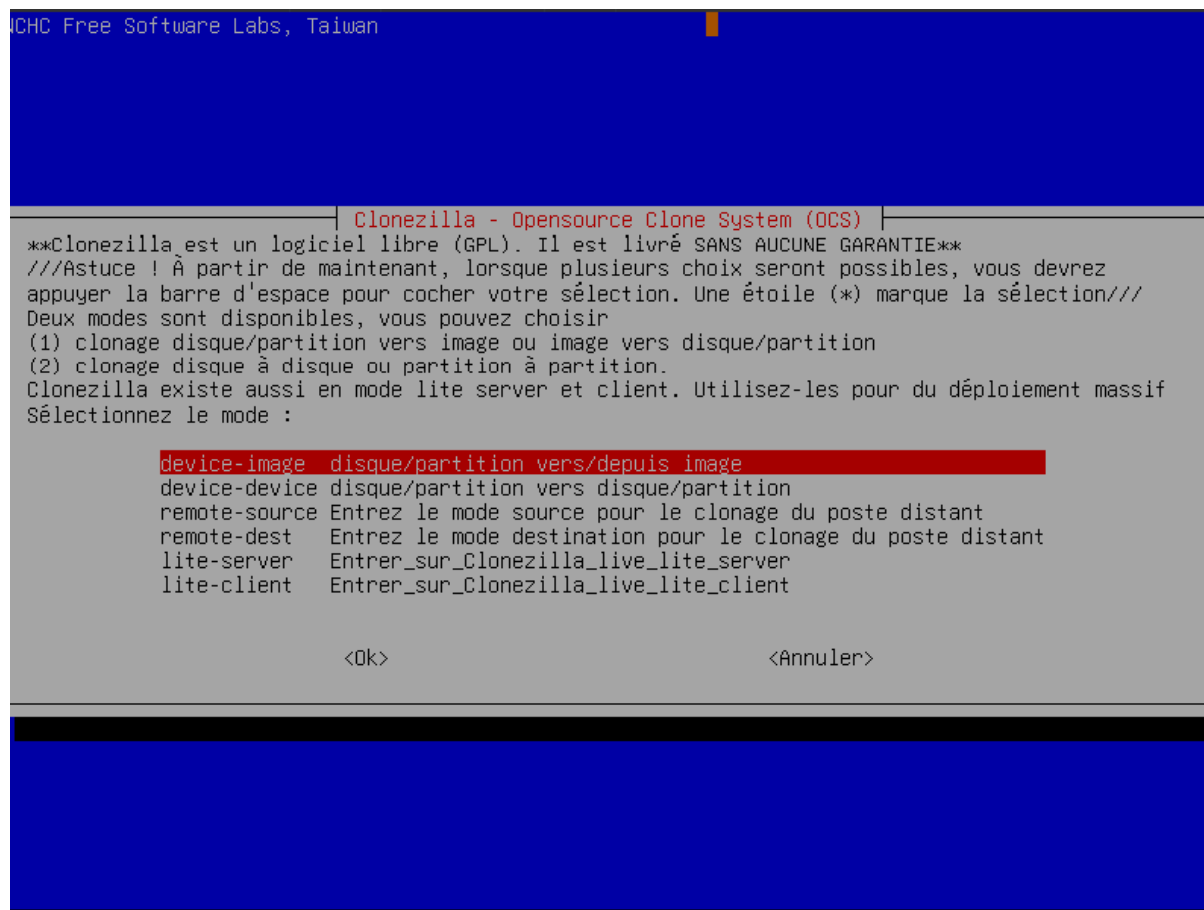
Chargement de l'iso Clonezilla via PXE

```

IP-Config: enp0s3 hardware address 08:00:27:dc:16:09 mtu 1500 DHCP RARP
IP-Config: enp0s3 complete (dhcp from 192.168.100.1):
  address: 192.168.100.13 broadcast: 192.168.100.255 netmask: 255.255.255.0
  gateway: 192.168.100.1 dns0 : 8.8.8.8 dns1 : 0.0.0.0
  rootserver: 192.168.100.1 rootpath:
  filename : pxelinux.0
Creating /etc/resolv.conf
Begin: Trying tftp -g -b 65464 -r /clonezilla/live/filesystem.squashfs -l /run/live/medium/live/filesystem.squashfs 192.168.100.1 ... /clonezilla/live/fil 0% | I 3835
  0:01:43 ETA
/cclonezilla/live/fil 2% | I 9.9M 0:01:15 ETA
/cclonezilla/live/fil 4% |* I 18.1M 0:01:00 ETA
/cclonezilla/live/fil 6% |** I 24.1M 0:00:59 ETA
/cclonezilla/live/fil 7% |*** I 28.0M 0:01:03 ETA
/cclonezilla/live/fil 8% |*** I 31.9M 0:01:06 ETA
/cclonezilla/live/fil 9% |*** I 36.0M 0:01:07 ETA
/cclonezilla/live/fil 10% |**** I 40.2M 0:01:08 ETA
/cclonezilla/live/fil 11% |**** I 44.0M 0:01:09 ETA
/cclonezilla/live/fil 12% |**** I 48.3M 0:01:09 ETA
/cclonezilla/live/fil 13% |**** I 51.4M 0:01:11 ETA
/cclonezilla/live/fil 14% |**** I 54.7M 0:01:12 ETA
/cclonezilla/live/fil 15% |**** I 58.8M 0:01:11 ETA
/cclonezilla/live/fil 16% |***** I 62.9M 0:01:11 ETA
/cclonezilla/live/fil 17% |***** I 68.2M 0:01:09 ETA
/cclonezilla/live/fil 18% |***** I 72.1M 0:01:09 ETA
/cclonezilla/live/fil 19% |***** I 76.1M 0:01:08 ETA
/cclonezilla/live/fil 20% |***** I 79.0M 0:01:09 ETA
/cclonezilla/live/fil 21% |***** I 83.3M 0:01:08 ETA
/cclonezilla/live/fil 22% |***** I 87.5M 0:01:07 ETA
/cclonezilla/live/fil 24% |***** I 92.7M 0:01:06 ETA
/cclonezilla/live/fil 24% |***** I 95.8M 0:01:06 ETA
/cclonezilla/live/fil 25% |***** I 99.3M 0:01:05 ETA
/cclonezilla/live/fil 26% |***** I 103M 0:01:05 ETA
/cclonezilla/live/fil 28% |***** I 107M 0:01:04 ETA
/cclonezilla/live/fil 29% |***** I 111M 0:01:03 ETA
/cclonezilla/live/fil 30% |***** I 115M 0:01:02 ETA

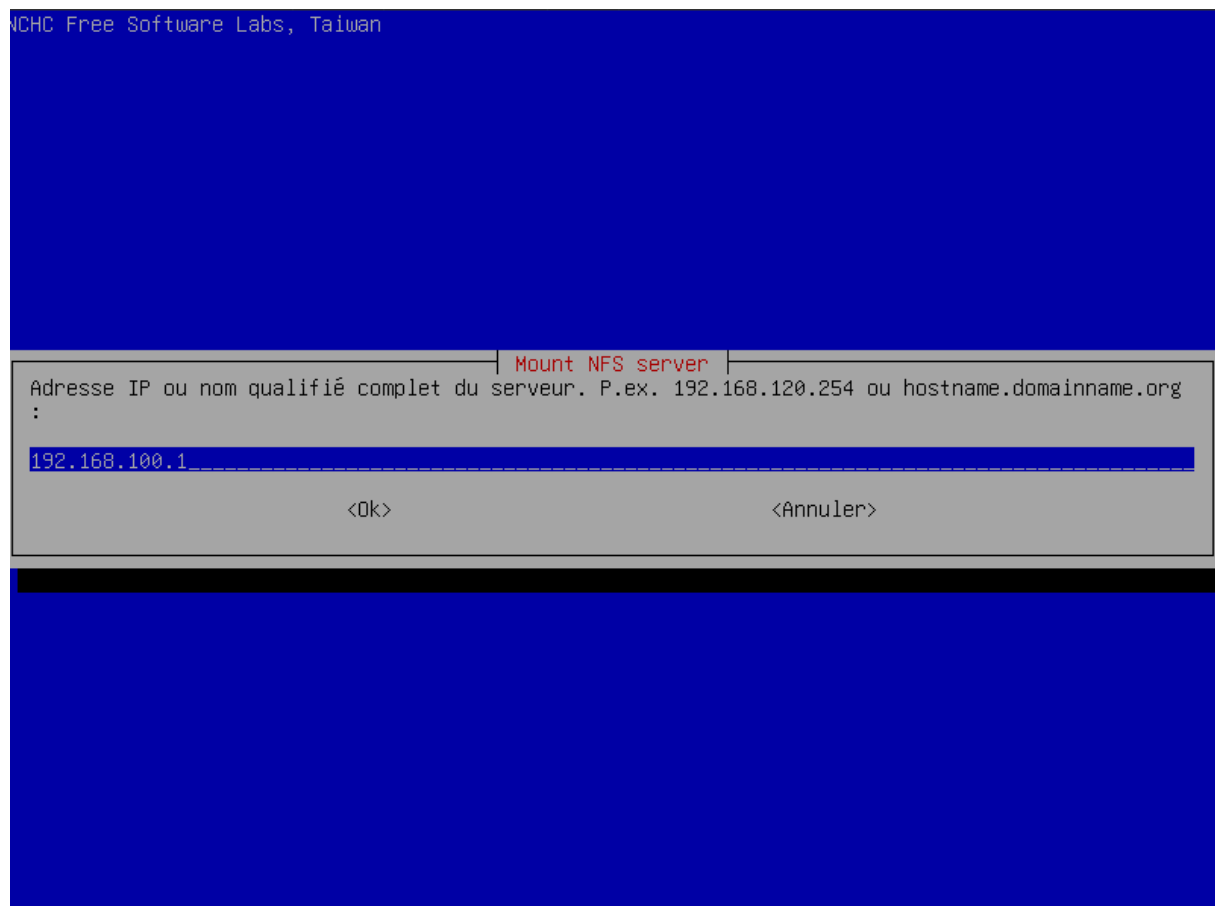
```

Ensuite on arrive sur le menu Clonezilla, il faut ici choisir Device/image/

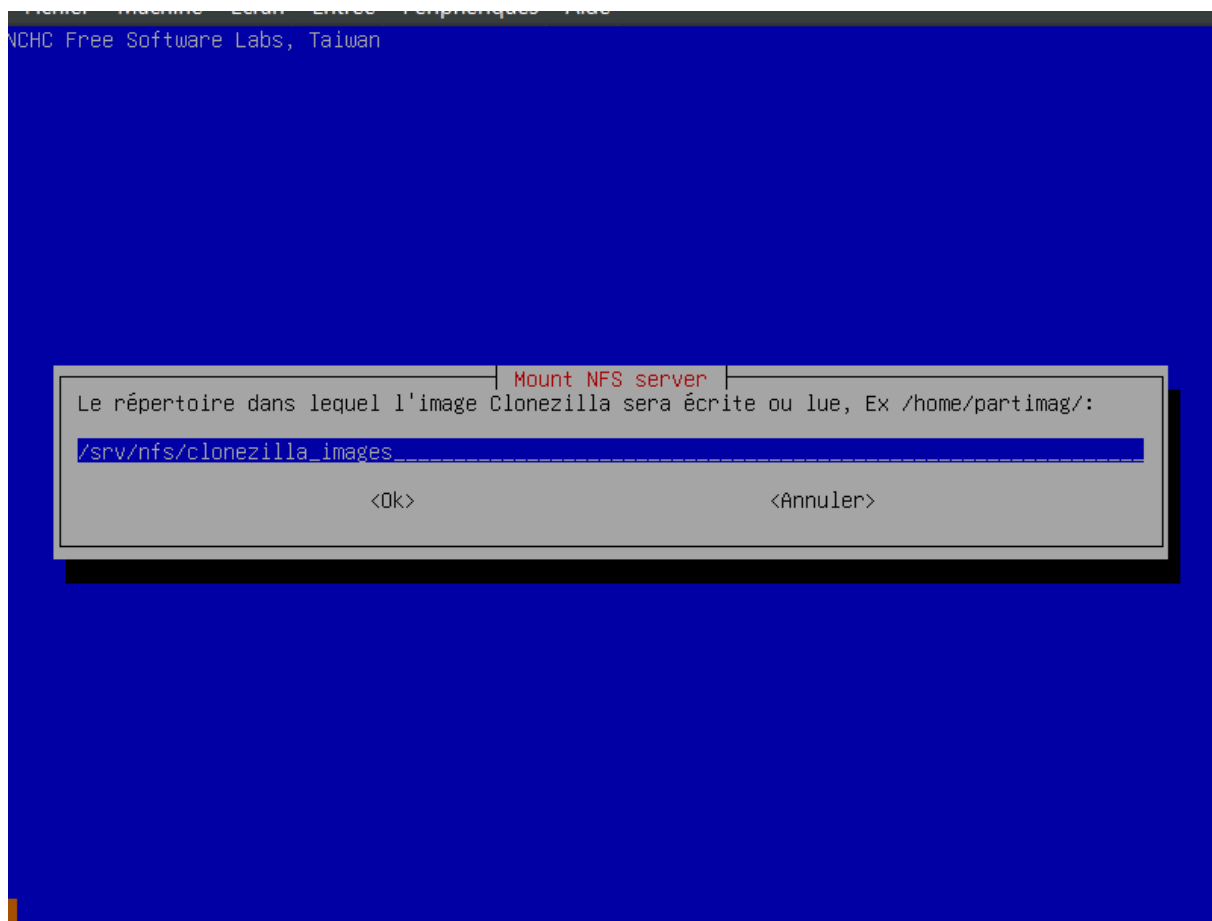


Ensuite on choisit le menu `nfs_serveur`. Clonezilla va skip l'étape des paramètres réseaux car notre serveur DHCP fait le job. Au prochain onglet il faut choisir le `NFS_V3` version la plus efficace.

Ici Clonezilla connaît déjà l'adresse IP du serveur automatiquement grace au DHCP



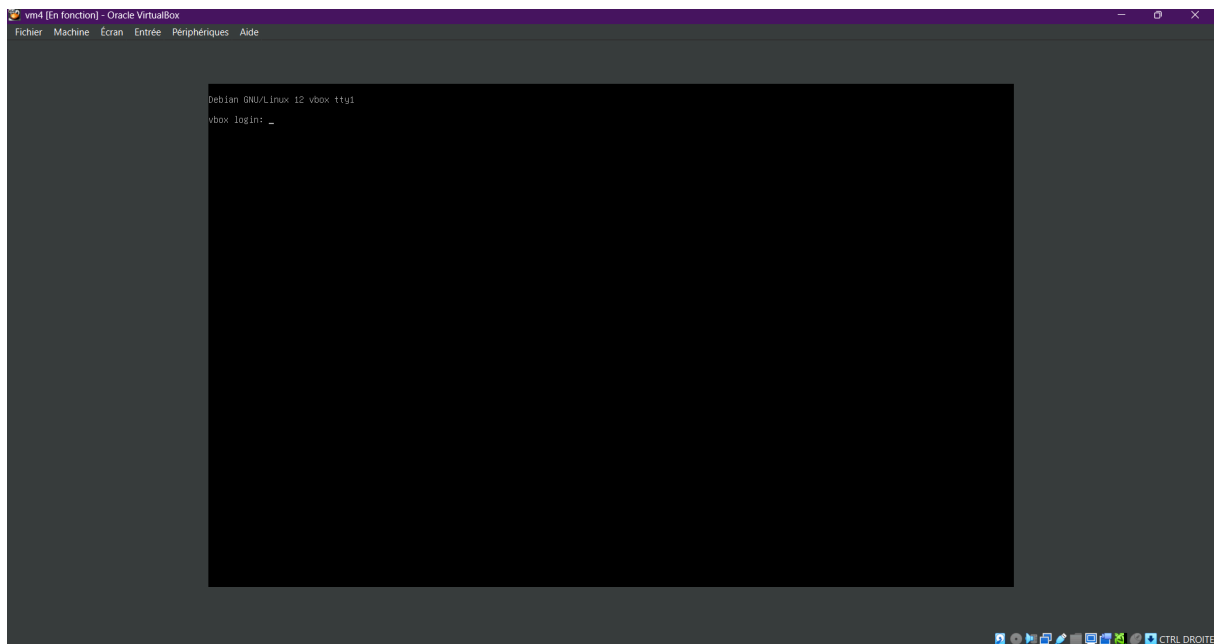
ATTENTION, BIEN REMPLACER LE CHEMIN DU RÉPERTOIRE



Voici l'interface qui devrait s'afficher, ensuite Clonezilla_live va s'afficher, veuillez sélectionner RecoveryDisk pour et non SaveDisk afin de récupérer VM2.

Ensuite les questions sont intuitifs et ne n'hésitent pas d'explications. Vous pouvez choisir la méthode zqsd en décompression car elle prend moins de place que les autres.

Voici notre VM restauré.



Pour aller plus loin j'ai donc fait plusieurs images dépendamment des logiciels ou des mises à jours que l'on veut installer sur notre nouvelle utilisateur, j'ai aussi créer un menu pxe dans lequel on peut choisir sur quoi booter.

```
DEFAULT vesamenu.c32
```

```
PROMPT 0
```

```
TIMEOUT 100
```

```
ONTIMEOUT clonezilla
```

```
MENU TITLE Boot PXE - ServeurZila
```

```
    LABEL clonezilla
```

```
MENU LABEL ^1. Installer Clonezilla
```

```
    KERNEL clonezilla/live/vmlinuz
```

```
    APPEND initrd=clonezilla/live/initrd.img boot=live config noswap edd=on nomoc
```

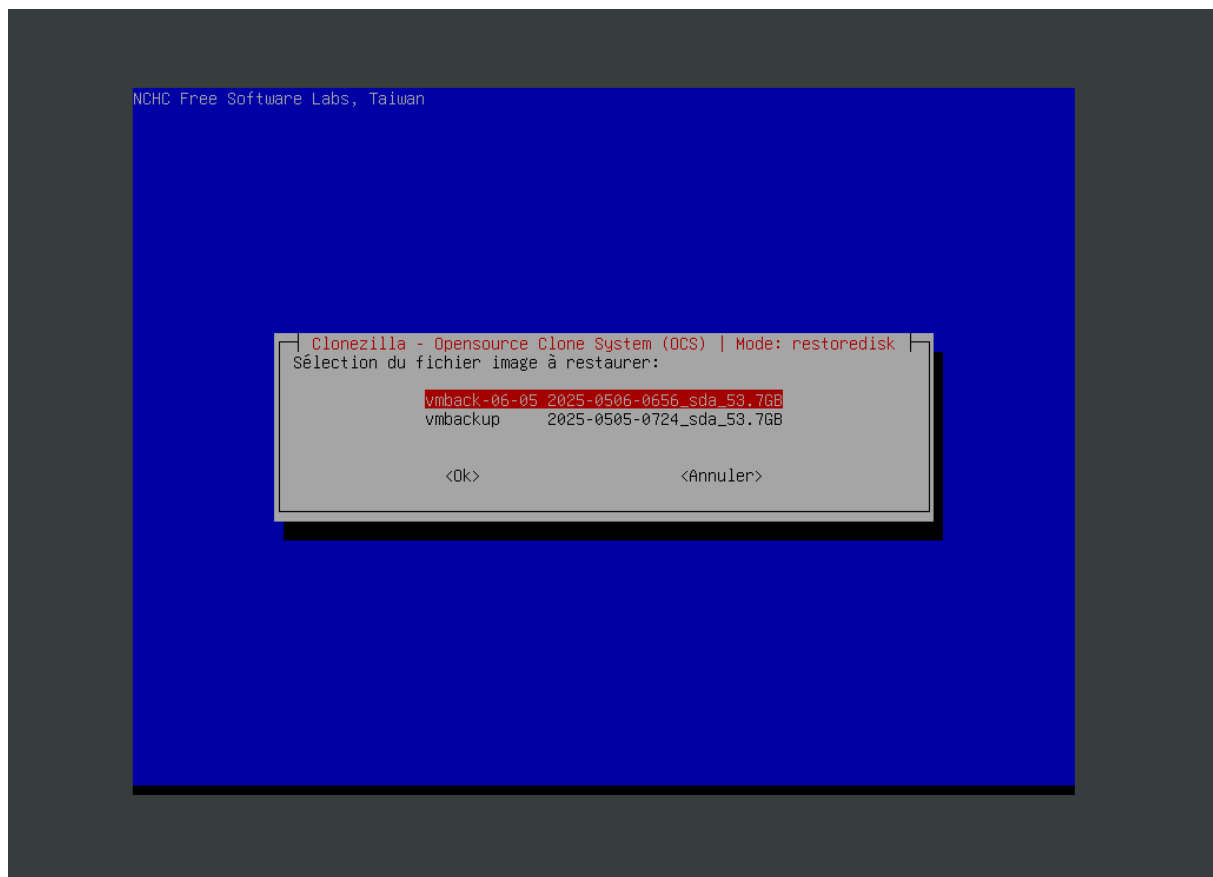
```
MENU LABEL ^2. Installer Debian depuis ISO (debianmenu.iso)
```

```
    KERNEL memdisk
```

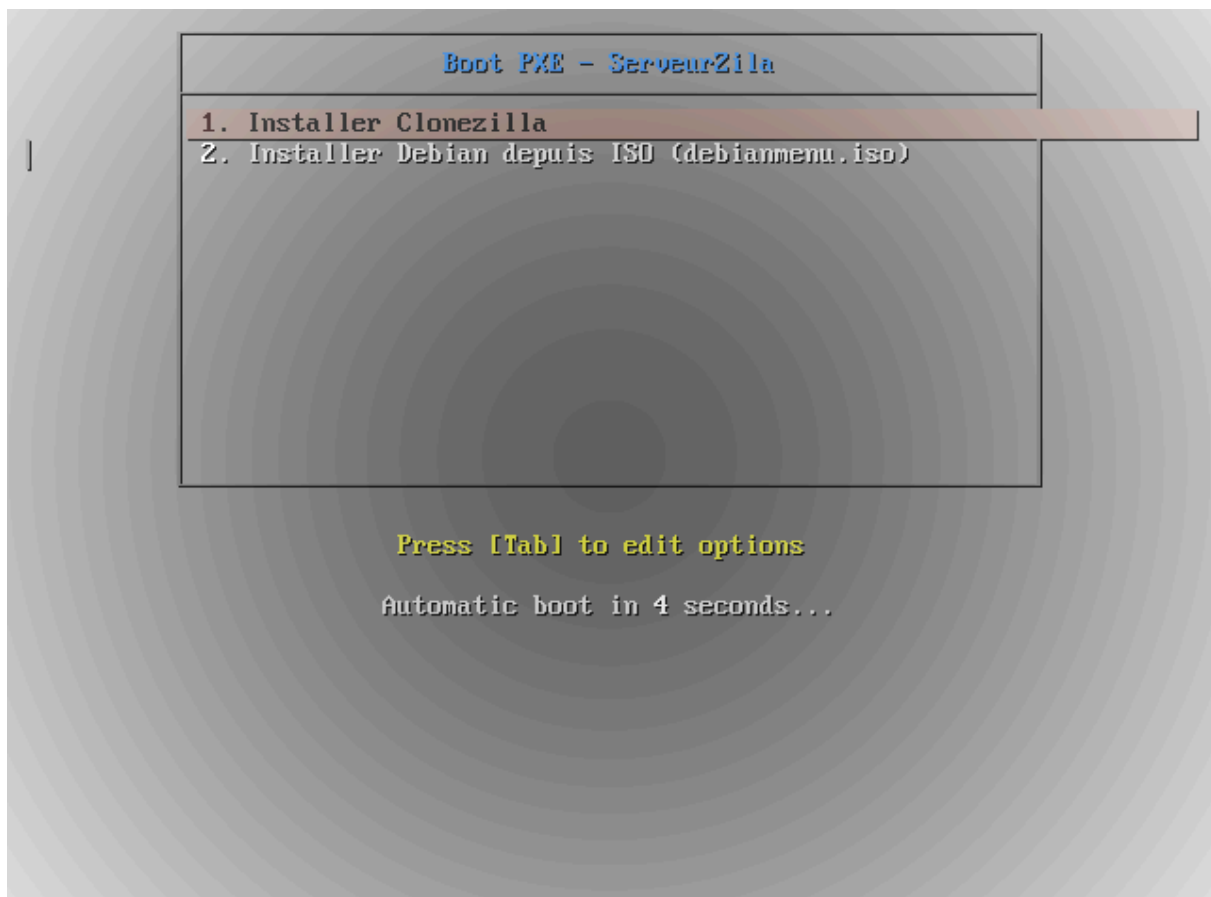
```
    INITRD debianmenu.iso
```

```
    APPEND iso raw
```

MENUS CLONEZILLA AVEC PLUSIEURS IMAGES :



MENU PXE AVEC PLUSIEURS OPTIONS :



On peut aussi éviter toutes ces étapes d'installation via ce script :

```
#!/bin/bash

# Monter le partage NFS contenant les images Clonezilla
mount -t nfs 192.168.X.X:/srv/nfs/clonezilla_images /home/partimag

# Vérifier si le montage a réussi
if [ $? -ne 0 ]; then
    echo "Erreur : échec du montage NFS."
    exit 1
fi
```


Lancer la restauration automatique

```
ocs-sr -e1 auto -e2 -g auto -r -j2 -k1 -scr -p poweroff restoredisk vmbackup sda
```



En arrivant sur Clonezilla, il faut faire **Ctrl + shift + F2** afin de passer en CLI et copier le script.



Détail et explication de chaque option :

| Option | Signification | Explication |
|--------------------------|---|--|
| <code>ocs-sr</code> | Script principal de Clonezilla | C'est le script de haut niveau utilisé pour lancer une sauvegarde ou une restauration de disque ou de partition. Il gère les différents outils (partclone, partimage, etc.). |
| <code>-e1 auto</code> | Paramètres avancés (1) | <code>auto</code> permet de laisser Clonezilla choisir les options par défaut pour les paramètres avancés de restauration (compression, etc.). |
| <code>-e2</code> | Fin des options <code>-e</code> | Marque la fin de la série des options <code>-e</code> . Obligatoire quand plusieurs <code>-e</code> sont utilisés. |
| <code>-g auto</code> | Nom de l'image auto | Cette option permet à Clonezilla de détecter automatiquement le nom de l'image si ce n'est pas précisé plus tard. Ici elle est redondante avec <code>vmbackup</code> mais ne gêne pas. |
| <code>-r</code> | Mode restauration | Indique que l'on veut restaurer une image (et non en créer une). |
| <code>-j2</code> | Pas de vérification du système de fichiers | Empêche Clonezilla de faire un check/reparation (fsck) du système de fichiers avant la restauration. Plus rapide, mais à utiliser si tu es sûr de ton image. |
| <code>-k1</code> | Ne pas recréer la table de partitions | Clonezilla ne touche pas à la table de partition existante sur le disque cible (<code>sda</code>). L'image écrasera simplement ce qui existe. |
| <code>-scr</code> | Pas de vérification de l'image | Clonezilla ne vérifie pas l'intégrité de l'image avant de restaurer. Gagne du temps mais à éviter si l'image peut être corrompue. |
| <code>-p poweroff</code> | Extinction automatique | Une fois la restauration terminée, le système s'éteint automatiquement . Idéal en mode PXE |

| | | |
|--------------------------|-------------------------|---|
| | | sans clavier. |
| <code>restoredisk</code> | Type d'opération | Indique qu'on veut restaurer un disque entier (et non une partition spécifique). |
| <code>vmbackup</code> | Nom de l'image | Nom du dossier contenant l'image dans <code>/home/partimag/</code> . Doit correspondre au nom réel de l'image. |
| <code>sda</code> | Disque cible | Le disque de destination sur lequel on veut restaurer l'image (<code>/dev/sda</code>). Toute son contenu sera écrasé . |

Souvent les entreprises utilisent aussi des scripts post Install avec des logiciels, cela rend les images moins lourdes pour les serveurs qui ne portent pas les logiciels et cela permet de gagner de la place.

Voici un exemple de script Post Install :

```
#!/bin/bash

# Vérification des privilèges root
if [[ $EUID -ne 0 ]]; then
    echo "Ce script doit être exécuté en tant que root. Utilisez sudo."
    exit 1
fi

echo "Mise à jour des dépôts APT..."
apt update && apt upgrade -y
```

```

# Liste des paquets
packages=(
  subversion
  git
  build-essential
  gdb
  autoconf
  libblas-dev
  cmake
  fftw2
  libeigen3-dev
  fftw-dev
  gfortran
  libgsl0-dev
  manpages-fr-dev
  xutils-dev
  libboost-all-dev
  libgmp-dev
  liblapack-dev
  libarpack2-dev
  libarpack++2-dev
  valgrind
  libxpm-dev
  f2c
  fort77
  clang
  python3-pip
  python3-venv
  ffmpeg
  libglew-dev
  freeglut3-dev
)

echo "Installation des paquets APT..."
apt install -y "${packages[@]}"

### INSTALLATION DE VS CODE ###
echo "Installation de VS Code..."
wget -qO- https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor >
echo "deb [arch=amd64] https://packages.microsoft.com/repos/code stable mai

```

```
apt update
apt install -y code

### INSTALLATION DE MINICONDA ###
echo "Installation de Miniconda (Conda)..."
CONDA_INSTALLER="Miniconda3-latest-Linux-x86_64.sh"
cd /tmp
wget https://repo.anaconda.com/miniconda/$CONDA_INSTALLER -O $CONDA_IN
bash $CONDA_INSTALLER -b -p /opt/miniconda
/opt/miniconda/bin/conda init

echo "Ajout de conda au PATH..."
export PATH="/opt/miniconda/bin:$PATH"

### INSTALLATION DE JUPYTER ###
echo "Installation de Jupyter Notebook via conda..."
/opt/miniconda/bin/conda install -y notebook

echo "Installation terminée avec succès."
```

Explication d'intégration avec SystemD :



Pourquoi créer un service **systemd** ?

systemd est le gestionnaire de services utilisé par Debian (et la plupart des distributions Linux récentes). Il permet de :

- Lancer automatiquement des programmes au démarrage.
- Contrôler l'ordre de démarrage des services.
- S'assurer que ton script s'exécute **une seule fois** si tu veux.

Étapes détaillées

1. Placer ton script

Place ton script dans un répertoire accessible. Par exemple :

```
bash
mv installer.sh /root/installer.sh
chmod +x /root/installer.sh
```

2. Créer le fichier de service **systemd**

Crée un fichier `/etc/systemd/system/installer.service` :

```
bash
nano /etc/systemd/system/installer.service
```

Et colle ceci :

```
ini
CopierModifier
[Unit]
Description=Installation automatique des outils
After=network.target

[Service]
ExecStart=/bin/bash /root/installer.sh
Type=oneshot
RemainAfterExit=true

[Install]
WantedBy=multi-user.target
```

Explication de chaque ligne :

- **[Unit]** définit quand et pourquoi ce service est lancé.

- `After=network.target` : attend que le réseau soit disponible (utile si ton script a besoin de `apt`).
- `[Service]` décrit **ce que fait le service**.
 - `ExecStart=...` : c'est le chemin vers ton script.
 - `Type=oneshot` : ton script s'exécute **une seule fois**, et le service se termine.
 - `RemainAfterExit=true` : `systemd` considère que le service est actif même après que le script est terminé.
- `[Install]` permet d'activer le service automatiquement.
 - `WantedBy=multi-user.target` : ton service se lancera **au démarrage normal du système**.

✓ 3. Activer le service

Une fois le fichier créé, tu dois **l'activer** :

```
bash
CopierModifier
systemctl daemon-reexec    # Recharge systemd
systemctl daemon-reload
systemctl enable installer.service
```

Tu peux aussi le tester immédiatement avec :

```
bash
CopierModifier
systemctl start installer.service
```

Et vérifier l'état avec :

```
bash
CopierModifier
systemctl status installer.service
```

🟡 (Facultatif) Supprimer le service après exécution

Si tu veux que le service **ne se relance pas à chaque démarrage**, tu peux :

- Ajouter une commande à la fin du script pour le désactiver :

```
bash
systemctl disable installer.service
```

- Ou même le supprimer :

```
bash
rm /etc/systemd/system/installer.service
```

```
SI CLONE [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

Debian GNU/Linux 12 vbox tty1

vbox login: siclone
Password:
Linux vbox 6.1.0-34-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.135-1 (2025-04-25) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon May  5 09:08:00 CEST 2025 on tty1
siclone@vbox:~$ systemctl status apache2
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-05-05 10:45:57 CEST; 13s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 590 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 637 (apache2)
     Tasks: 55 (limit: 9472)
    Memory: 23.9M
       CPU: 263ms
   CGroup: /system.slice/apache2.service
           └─637 /usr/sbin/apache2 -k start
             └─639 /usr/sbin/apache2 -k start
               └─640 /usr/sbin/apache2 -k start

Warning: some journal files were not opened due to insufficient permissions.
siclone@vbox:~$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda   8:0    0   50G  0 disk
├─sda1 8:1    0    9,6G  0 part /
├─sda2 8:2    0     1K  0 part 
├─sda5 8:5    0    3,5G  0 part /var
├─sda6 8:6    0   976M  0 part [SWAP]
├─sda7 8:7    0   685M  0 part /tmp
├─sda8 8:8    0   35,2G  0 part /home
sr0   11:0    1  1024M  0 rom
sr1   11:1    1  1024M  0 rom
siclone@vbox:~$ _
```

✓ Changez l'utilisateur de sa vm clone pour pas faire de doublon

Ouvre un terminal en tant que `siclone` et exécute :

```
sudo adduser admin_temp
sudo usermod -aG sudo admin_temp
```


Cela crée un utilisateur `admin_temp` et lui donne les droits sudo.

✓ 2. Déconnecte-toi de `siclone` et connecte-toi avec `admin_temp`

- Déconnecte ta session (`logout` , `exit` , ou depuis l'interface graphique).
- Connecte-toi avec `admin_temp` .

✓ 3. Termine la session `siclone` et renomme-le

Depuis le compte `admin_temp` , fais :

```
sudo loginctl terminate-user siclone
sudo usermod -l nouveau_nom siclone
sudo usermod -d /home/nouveau_nom -m nouveau_nom
```

🔄 Cette commande :

- Change le nom du compte `siclone` en `nouveau_nom` .
- Déplace le dossier `/home/siclone` vers `/home/nouveau_nom` .

Ajout d'un Pare-feu UFW :

```
sudo apt install ufw
```

Configurations des règles du pare-feu

```
# -A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT
```

Ici on commente la ligne dans le fichier de configuration pour empêcher les requêtes de ping

```
# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
#-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT
```