

Atelier OpenVZ

[Introduction](#)

[Etape 1: installation de Debian etch \(DEJA FAIT\)](#)

[Etape 2: ajout des sources d'installation CRIL et du package openvz-support \(DEJA FAIT\)](#)

[Etape 3: installation du package kernel OpenVZ \(DEJA FAIT\)](#)

[Etape 4: création de templates à l'aide du support](#)

[Etape 5: création d'une machine virtuelle à l'aide du support](#)

[Etape 6: manipulations](#)

Introduction

Le but de ce document est de permettre de faire l'atelier, pas à pas, en suivant les instructions, vous-même! En cas de problème, n'hésitez pas à me demander de l'aide. Les personnes avancées peuvent soit aider les autres participants, soit lire les scripts, et documentations et références.

Les sections indiquées par DEJA FAIT sont déjà effectuées sur votre poste de travail (afin de gagner du temps).

Etape 1: installation de Debian etch (DEJA FAIT)

A l'aide du CD d'installation réseau (netinstall), installer par réseau une Debian etch sans configuration particulière:

- Choose a language: C; choose a country: Europe/Switzerland; Keymap Swiss/French
- DHCP: Hostname: vzhost1; domain: alphanet.ch
- Partitionning: Guided, use entire disk; All files in a single partition; Finish partitionning; Yes.
- Root password: demo; User: demo, demo, password demo.
- Package survey: à ce moment ALT-F2, chroot /target; vi /etc/apt/sources.list, insérer avant ftp.ch.debian.org mirror local si utilisé; apt-get update; exit
- Software selection: si désiré, laisser Desktop environment (et Standard system)

Connexion et installation de quelques packages utiles

- `vi /etc/apt/sources.list`, commenter le CD
- `apt-get install tcpdump ltrace strace telnet ssh rsync`
- `apt-get update && apt-get -u dist-upgrade`
- `apt-get clean`

NB: on aurait pu utiliser FAI ou du preseeding, mais il y a peu de choses à faire. Dans notre cas on a simplement dupliqué sur les 5 autres machines avec dd ou tar et modifié juste le nom de la machine `/etc/hostname`, `/etc/hosts`, `/etc/mailname`

BUGS

- z25

Etape 2: ajout des sources d'installation CRIL et du package `openvz-support` (DEJA FAIT)

Sous root:

```
echo >> /etc/apt/sources.list deb http://packages.cril.ch/cril/debian/packages/ etch cril
wget -O - http://www.cril.ch/pub/CRIL/WebHome/3D79EDCA.asc | apt-key add -
apt-get update && apt-get -u dist-upgrade
apt-get install openvz-support
# copier la configuration attachée à ce topic dans /etc/default/openvz-support
/usr/lib/openvz-support/scripts/install-openvz --no-kernel-generate --complete
```

Cette dernière procédure préconfigure le réseau pour notre utilisation, ainsi que des redirections de ports de 2222 à 22231 (pour 192.168.100.100:22 à 192.168.100.109:22), utile par exemple pour accéder via l'adresse publique.

Etape 3: installation du package kernel OpenVZ (DEJA FAIT)

En alternative à utiliser des paquets kernel précompilés, on peut également utiliser la génération de kernel via `/usr/lib/openvz-support/scripts/make-kernel`. Mais c'est plus simple d'utiliser les kernels préparés comme ci-dessous:

Sous root:

```
cat >> /etc/apt/sources.list <<EOF

deb http://download.openvz.org/debian-systs etch openvz
deb-src http://download.openvz.org/debian-systs etch openvz
EOF

wget -q -O - http://download.openvz.org/debian-systs/dso_archiv_signing_key.asc | apt-key add -

apt-get update && apt-get -u dist-upgrade # mise à jour outils vz
apt-get install fza kernel-2.6.18-686
apt-get clean
reboot
```

(ne pas hésiter à faire: `apt-cache search fza` pour trouver un kernel plus adapté à vos besoins, p.ex. SMP ou 64 bits).

Etape 4: création de templates à l'aide du support

L'idée est de préparer une version de base d'un OS (ici Debian) à utiliser comme template lors de la création de nouvelles machines virtuelles. Cette génération se fait notamment grâce à l'outil `debootstrap`

et une archive Debian.

Sous root:

```
# Autoriser les machines virtuelles à sortir en SNAT
iptables -t nat -I POSTROUTING -o eth0 -s 192.168.100.0/24 -j MASQUERADE

apt-get install debootstrap
/usr/lib/opensvz-support/scripts/create-template
```

Attention: le numéro de VM 777 est réservé pour la création de templates. Cette création prend environ 10 minutes sur une machine Phoenix avec une bonne connexion réseau. Utilisez ce temps à lire les manuels systèmes concernant les commandes (voir étape 6 ci-dessous).

(voir aussi http://wiki.opensvz.org/Debian_template_creation)

En cas de nécessité (arrêt au milieu de l'installation), faire:

```
vzctl stop 777; rm -rf /etc/vz/conf/777.conf /var/lib/vz/private/777
```

et réessayer.

Etape 5: création d'une machine virtuelle à l'aide du support

Nous allons créer une machine virtuelle portant le numéro 100. Par convention, elle sera également accessible en 192.168.100.100 (p.ex. ses services: ssh, etc). Voir `/etc/default/opensvz-support` pour cette configuration et `/etc/rc.local`.

Nous allons définir que cette VM prendra la moitié des ressources de la machine, cela se fait à l'aide de l'outil `vzsplit`. Une configuration plus fine est certes possible, mais en général utiliser cet outil est plus fiable que de deviner des limites à la main.

```
/usr/lib/opensvz-support/scripts/vz-instanciate 100

vzlist -a

# Appliquer les limites
# (pour être propre, éditer ensuite le fichier et supprimer les doublons!)
vzsplit -n 2 >> /etc/vz/conf/100.conf
```

Si nécessaire, vous pouvez détruire la VM 100:

```
vzctl stop 100
vzctl destroy 100
```

Etape 6: manipulations

Essayez-vous aux diverses opérations suivantes:

- démarrer et stopper une VM: `vzctl start 100, vzctl stop 100`
- liste des processus depuis le système englobant: `ps auxw | more`
- liste des processus depuis la machine hébergée (VM): `vzctl exec 100 ps auxw | more`
- mémoire libre et espace disque dans les deux cas (`free, df`)
- n'hésitez pas à consulter les manpages: `vzctl(8), vz` p.ex. `man 8 vzctl, man vz`, ainsi que toutes les manpages associées (voir `man -k vz`), notamment `vzsplit(8)`, `vzlist` et exécutez ces diverses commandes *après* avoir lu le manuel!
- connectez-vous en ssh sur la VM 100
 - lancez le programme `perl -e 'while (1) { }'` et déterminez combien de CPU est utilisé sur la machine hébergée (VM) et la machine hôte. Modifiez éventuellement le CPU alloué à la VM à l'aide de la commande `vzctl set 100 --cpulimit 25%`. Est-ce que cela change en temps réel ? A quoi pourrait servir l'option `--save` dans le changement de paramètres ? Quels paramètres peuvent-être modifiés ? (voir `vzctl set`), Terminez le programme avec CTRL-C.
 - essayez d'augmenter la mémoire disponible et l'espace disque (cf http://wiki.openvz.org/UBC_parameters)
 - installez le package `apache2` et vérifiez qu'il fonctionne (port 80 de l'adresse IP de 100)
 - si apache ne démarre pas (`ps auxw | grep apache`, dans la VM!), faites `tail /var/log/apache2/error.log`
 - modifiez `/etc/apache/apache2.conf`
 - diminuez le nombre de thread dans `ifModule ? mpm_worker_module`
- nous allons crée une copie de la VM 100 dans la VM 101:
 - dupliquez la VM 100 dans la VM 101 comme suit:


```
vzctl create 101 --config 100
vzctl stop 100 # pas on-line
rsync -aH --delete /var/lib/vz/private/100/. /var/lib/vz/private/101/.
vzctl start 100
```
 - changez la configuration IP et p.ex. de nom de la machine via `vzctl set` (indications utiliser `--save` et `--ipdel 192.168.100.100` et `--ipadd 192.168.100.101`)
 - démarrez cette VM et testez qu'Apache répond également.
 - faites `date > /var/www/bla` dans cette VM et contrôlez qu'Apache sur cette VM liste ce fichier, mais pas sur l'autre VM 100 (==w3m <http://192.168.100.101/bla>==)
 - faudrait-il changer la clé SSH privée de machine de cette nouvelle VM ? y-a-t-il d'autres choses qui peuvent être à faire dans le cas d'une telle copie ?
 - imaginez ce qu'il faudrait faire pour intégrer une machine réelle comme VM
- documentez-vous sur la commande `vzmigrate`. Qu'est-ce que la migration *on-line* ?
 - connectez-vous en SSH sur la VM 100 depuis le host: `ssh 192.168.100.100 -l root`
 - suspendez la VM: `vzctl 100 checkpnt`
 - faites `vzlist -a`

- confirmez que la session SSH est coincée
 - n'attendez pas trop (timeout règles firewall p.ex.) et faites: `vzctl 100 restore`
 - la session SSH continue-t-elle ?
 - où est stocké l'état de la VM pendant "suspend" (deux endroits) ?
 - comment feriez-vous pour une migration transparente entre deux machines physiques d'une VM ?
- sur le modèle de `/etc/rc.local`, ajoutez une redirection de port sur l'adresse publique de votre système de manière à ce que l'Apache de chaque machine virtuelle soit accessible sous `10080 + 100 * (adresse - 100)`
 - consultez `/proc/user_beancounters` et déterminez si des dépassements de ressource ont eu lieu; si oui que proposez-vous ?
 - on aimerait pouvoir utiliser sshfs, encfs ou d'autres systèmes de fichiers basés sur **fuse**. Pour ce faire il faut charger le module kernel `modprobe fuse` (dans l'hôte), et autoriser ce périphérique dans la VM (cf `--devices` dans `vzctl(8)`).
 - si vous voulez depuis une machine qui n'a PAS de sous-réseau 192.168.100 local accéder au sous-réseau 100 de la machine 192.168.1.193, faire:

```
route add -net 192.168.100.0/24 gw 192.168.1.193
```

-- [MarcSCHAEFER](#) - 30 Dec 2008

This topic: Main > [TWikiUsers](#) > [MarcSCHAEFER](#) > [MarcSCHAEFERLeftBar](#) > [MarcSCHAEFERHomePage](#) > [MarcSCHAEFERConferencesLinuxNE](#) > [OpenVZ](#) > [AtelierOpenVZ](#)

History: r1 - 30 Dec 2008 - 18:20:35 - [MarcSCHAEFER](#)

Copyright © by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? [Send feedback](#)

