OpenVZ

- Objet : OpenVZ
- Niveau requis :
  avisé
- Commentaires : *Technologie de paravirtualisation*
- Suivi :
  à-tester
  - Création par smolski le 23/11/2009
  - Testé par .... le ....
- Commentaires sur le forum : C'est ici[1)]

Introduction

Wikipédia :

OpenVZ est une technologie de paravirtualisation de niveau système d'exploitation basée sur le noyau Linux.
OpenVZ permet à un serveur physique d'exécuter de multiples instances de systèmes d'exploitation isolés, connus sous le nom de serveurs privés virtuels (VPS) ou environnements virtuels (VE).

En comparaison aux machines virtuelles telles que VMware et aux technologies de paravirtualisation telles que Xen, OpenVZ offre moins de flexibilité dans le choix du système d'exploitation : le système d'exploitation invité et hôte doivent être de type Linux (bien que les distributions de Linux peuvent être différentes dans des VEs différents).

Cependant, la virtualisation au niveau OS d'OpenVZ offre :

- une meilleure performance,
- une meilleure scalabilité (i.e. évolution),
- une meilleure densité,
- une meilleure gestion de ressource dynamique,
- et une meilleure facilité d'administration que ses alternatives.

Selon le site Web d'OpenVZ, cette méthode de virtualisation introduirait une très faible pénalité sur les performances : 1 à 3% de pertes seulement par rapport à un ordinateur physique.

OpenVZ est la base de Virtuozzo, un produit propriétaire fourni par SWsoft, Inc. OpenVZ est distribué sous la Licence publique générale GNU version 2.

OpenVZ comprend le noyau Linux et un jeu de commandes utilisateurs.

Ajout du dépôt

En théorie, cette étape est inutile.

Je la mets sans l'avoir validée et juste à titre de mémo.

Dans le fichier **sources.list** on ajoute le dépôt :

```
http://download.openvz.org/debian-systs lenny openvz
```

On télécharge la clef et on fait la mise à jour :

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

Installation du bon kernel et des softs de contrôle des machines virtuelles (VE)

Kernel OpenVZ

Un kernel OpenVZ, ce n'est ni plus ni moins qu'un kernel classique qui a été patché afin d'améliorer le principe de chroot.

Utilité de vzctl et vzquota

vzctl

```
Un utilitaire pour contrôler les VE (création, destruction, démarrage, arrêt, etc.)
```

vzquota

```
Un programme pour gérer les quota des VE.
Très souvent utilisé indirectement et implicitement (grâce à vzctl).
```

Installation des paquets utiles

```
apt-get install iproute libatm1 linux-image-2.6.26-2-openvz-686 linux-image-openvz-686 rsync vzctl vzquota
```

Si le paquet linux-image-openvz-686 ou linux-image-2.6.26-2-openvz-686 n'existe pas, alors un simple :

```
apt-cache search openvz | grep 686 | grep image
```

suffira pour déterminer le bon kernel à installer.

Avant de rebooter, vérifiez votre grub.
Dans grub :

```
/boot/grub/menu.lst
```

Dans grub2 :

```
 En attente...
```

ce dernier doit pointer sur le kernel OpenVZ. Après vérification, on reboot.

Principe de OpenVZ

Très simplement, OpenVZ c'est du chroot très fortement optimisé et amélioré.

Après le reboot

- On vérifie que l'on est sur le bon kernel :

```
uname -a
```

Une réponse de ce type doit alors apparaître :

retour de la commande

```
2.6.26-2-openvz-686
```

- On vérifie également que le daemon OpenVZ est actif :

```
ps ax | grep vz
```

On doit obtenir quelque chose de ce genre :

retour de la commande

```
2349 ?        S      0:00 [vzmond]
```

- On configure correctement le sysctl (pour que les VE puissent accéder au NET entre autre) :

```
nano /etc/sysctl.conf
```

- Et on ajoute ces lignes à la fin :

```
net.ipv4.conf.default.forwarding=1
net.ipv4.conf.default.proxy_arp = 0
```

```
## Ligne à rajouter (oubliée dans les tutoriels que l'on trouve sur le net).

net.ipv4.ip_forward = 1
```

- On recharge ensuite les options :

```
sysctl -p
```

Activer NAT

On va maintenant créer un script qui va activer la NAT dans iptables pour que les VE puissent accéder a internet (Exemple avec la carte Eth0 sur le serveur) :

```
nano $HOME/nat_VE.sh
```

- Que l'on rend exécutable (seul le root à besoin d'un accès) :

```
chmod 700 $HOME/nat_VE.sh
```

- Et que l'on remplit comme ci-dessous :

```
#!/bin/bash
# Active les Nattage au reboot

# Toutes les VE accède au NET
/sbin/iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to IP_DU_SERVEUR
```

- On automatise ce script en l'ajoutant à la crontab :

```
crontab -e
```

- Et on ajoute cette ligne :

```
 @reboot /root/nat_VE.sh
```

Egalement, pour des raisons de simplification sur d'éventuels tutoriels trouvés sur Internet :

```
ln -s /var/lib/vz /vz
```

Création des machines virtuelles (VE)

On attaque enfin la partie intéressante 

Template

Bon, tout d'abord la notion de template.

Notion de template

Très simplement, un template est une archive tar.gz (généralement) qui va contenir la base du système d'exploitaition de votre machine virtuelle. Elles se trouvent dans :

```
/vz/template/cache
```

Vous pouvez en obtenir ici :
http://download.openvz.org/contrib/template/precreated/

J'en ai testée pas mal, et à part celle de ubuntu, debian, et gentoo, le reste c'est un peu galère...
Je mettrai plus tard à disposition des VE toutes prêtes qui intègrent des services comme un :

```
VE-serveur-ftp,
une VE-serveur-nagios,
etc…
```

Bon, j'ai téléchargé mes templates, qu'est-ce que je fais maintenant ?

Création de la machine virtuelle

Assez papoté 😀

On créé notre machine virtuelle :

```
 vzctl create $numero_de_la_machine --ostemplate
$votre_template_sans_extensions
```

→ $numero_de_la_machine sera la numéro de votre machine virtuelle. La valeure minimale est 100
→ $votre_template_sans_extension sera le nom de l'archive que vous avez placée dans /vz/template/cache mais sans l'extension (ne marche pas avec tar.bz2 je crois).

Scripts de création de VE

Un script perso de création de machines virtuelles.

Des questions vont vous être posées comme l'allocation de RAM, d'espace disque ou de CPU.


Je n'offre aucune garantie sur le fonctionnement de ce script.

```
nano $HOME/creation_VE.sh
```

- On le rend exécutable :

```
chmod 700 $HOME/creation_VE.sh
```

- On le remplit :

creation_VE.sh

```
#!/bin/bash

clear

###################################
#### Affichage des VE actives ####
###################################

echo "    --------Les differents VE en cours d'utilisation-------"
vzlist

###############################
#### Selection du templates ####
###############################

PS3="Selection -> "
echo "Selectionner l'OS de votre template"
select template in "Ubuntu-8.04-x86" "Debian-5.0-x86"; do

########################
#### En cas d'erreur ####
########################

if [ -z "$template" ]
then
    echo "Erreur: entrez un des chiffres proposes." 1>&2

#############################
#### Cas du premier choix ####
#############################

elif [ "$REPLY" -eq 1 ]
then
    template=ubuntu-8.04-x86-minimal
    break

#############################
#### Cas du second choix ####
#############################

elif [ "$REPLY" -eq 2 ]
```

```
then
    template=debian-5.0-x86-minimal
    break
fi
done

########################################
#### Numero de la machine virtuelle ####
########################################
echo -n "Entrez le numero du VPS (min=100) : "
read numero

##################################
#### Nom d'hôte de la machine ####
##################################
echo -n "Entrez son hostname : "
read hostname

#############################################################################
###############################
#### Ip de la machine, je vous conseille de choisir un sous-réseau (on
verra le nattage plus tard ####
#############################################################################
###############################
echo -n "Entrez l'adresse IP du VPS : "
read adresseip

##############################
#### Serveur DNS de la VE ####
##############################
echo -n "Entrez le serveur de nom : "
read serveurdns

#############################################################
#### Nettoyage de l'affichage et affichage des partitions ####
#############################################################
clear
df -h

########################################
#### Taille à allouer à la VE en Mo ####
########################################

echo -n "Entrez le seuil d'alerte d'utilisation du disque dur en Mo : "
read diskutil

echo -n "Entrer la limite d'utilisation du disque dur en Mo : "
read diskmax

clear
```

```
############################
#### Limitation du CPU ? ####
############################
echo -n "Entrer l'utilisation CPU maximum en % /!\1cpu= /100% 2cpu=
/200%...(0=unlimited): "
read cpulimit

############################
#### Des stats sur les VE ####
############################
vzcpucheck -v

##########################
#### Ressources des VE ####
##########################
echo -n "Entrer la capacite max d'utilisation CPU en unites de temp : "
read cpuunits

##########################
#### Création de la VE ####
##########################
vzctl create "$numero" --ostemplate "$template"

#####################################
#### Etablissement des paramètres ####
#####################################

vzctl set "$numero" --nameserver "$serveurdns" --hostname "$hostname" -
-ipadd "$adresseip" --onboot yes --diskspace "$diskutil"M:"$diskmax"M -
-kmemsize unlimited --oomguarpages unlimited  --privvmpages unlimited -
-vmguarpages unlimited --numproc unlimited --numtcpsock unlimited --
numothersock unlimited --lockedpages unlimited --numpty unlimited --
numiptent unlimited --shmpages unlimited --numsiginfo unlimited --
numflock unlimited --numfile unlimited --cpuunits  "$cpuunits" --
cpulimit "$cpulimit" --tcpsndbuf unlimited --tcprcvbuf unlimited --
othersockbuf unlimited --dgramrcvbuf unlimited --dcachesize unlimited -
-diskinode unlimited --physpages unlimited --save

############################
#### Demarrage de la VE ####
############################
vzctl start "$numero"
```

Configuration de la machine virtuelle

Si vous n'avez pas envie de passer par mon script, voici un fichier de configuration tout prêt (explications ci-dessous) :

```
cat /etc/vz/conf/100.conf
```

Résultat :

```
ONBOOT="yes"

# UBC parameters (in form of barrier:limit)
KMEMSIZE="2147483647:2147483647"
LOCKEDPAGES="2147483647:2147483647"
PRIVVMPAGES="2147483647:2147483647"
SHMPAGES="2147483647:2147483647"
NUMPROC="2147483647:2147483647"
PHYSPAGES="2147483647:2147483647"
VMGUARPAGES="2147483647:2147483647"
OOMGUARPAGES="2147483647:2147483647"
NUMTCPSOCK="2147483647:2147483647"
NUMFLOCK="2147483647:2147483647"
NUMPTY="2147483647:2147483647"
NUMSIGINFO="2147483647:2147483647"
TCPSNDBUF="2147483647:2147483647"
TCPRCVBUF="2147483647:2147483647"
OTHERSOCKBUF="2147483647:2147483647"
DGRAMRCVBUF="2147483647:2147483647"
NUMOTHERSOCK="2147483647:2147483647"
DCACHESIZE="2147483647:2147483647"
NUMFILE="2147483647:2147483647"
AVNUMPROC="180:180"
NUMIPTENT="2147483647:2147483647"

# Disk quota parameters (in form of softlimit:hardlimit)
DISKSPACE="25360000:25872000"
DISKINODES="2147483647:2147483647"
QUOTATIME="0"

# CPU fair sheduler parameter
CPUUNITS="100000"


VE_ROOT="/var/lib/vz/root/$VEID"
VE_PRIVATE="/var/lib/vz/private/$VEID"
OSTEMPLATE="debian-5.0-x86"
ORIGIN_SAMPLE="vps.basic"
IP_ADDRESS="192.168.0.1"
HOSTNAME="iroffer"
NAMESERVER="208.67.222.222"
CPULIMIT="0"
```

Explications des valeurs importantes de ce fichier :

```
→ CPULIMIT : pas de limitation CPU dans la VE
→ NAMESERVER : le serveur DNS
→ HOSTNAME : le nom d'hôte de la VE
→ IP_ADDRESS : adresse IP de la VE
→ OSTEMPLATE : template sur lequel à été créé la VE

VE_ROOT=”/var/lib/vz/root/$VEID”
VE_PRIVATE=”/var/lib/vz/private/$VEID” ← chemin du contenu de la VE
DISKSPACE=“25360000:25872000” : espace disque alloué (ici 25 Go)
KMEMSIZE=“2147483647:2147483647” : RAM allouée (ici 2 Go)

ONBOOT=“yes” : démarrage automatique de la VE au reboot du serveur Physique.
Routage des ports extérieurs sur les VE
```

Rediriger les ports

Ensuite il faut rediriger des ports pour que les VE soit accessible de l'extérieur.

Le script suivant évitera de passer par les fastidieuses commandes iptables.
Il inscrira les “routes” additionnelles dans le script ”/root/natVE.sh” créé lors de l'installation.

- On crée le script :

```
nano /root/natVE.sh
```

- On rend le script exécutable :

```
chmod 700 /root/natVE.sh
```

- Et on utilise le script suivant :
- Script pour activer la Nat :

```
#!/bin/bash

#################################
#### Afficher la conf reseau ####
#################################

echo "    ---------Configuration réseau------------"
cat /etc/network/interfaces

#################################
#### Afficher les VE activés ####
#################################

echo "    ----------Les différents VE en cours d'utilisation-------------"
vzlist

#########################################
```

```
#### Selection du protocole : TCD/UDP ####
########################################

PS3="Selection -> "
echo "Sélectionner le type de port"
select porttype in "udp" "tcp"; do

#########################
#### En cas d'erreur ####
#########################

if [ -z "$porttype" ]
then
   echo "Erreur: entrez un des chiffres proposés." 1>&2

##############################
#### Cas du premier choix ####
##############################

elif [ "$REPLY" -eq 1 ]
then
    porttype=udp
    break

##############################
#### Cas du second choix ####
##############################

elif [ "$REPLY" -eq 2 ]
then
   porttype=tcp
   break
fi
done

###################################
#### Selection de l'Interface ####
###################################

PS3="Selection -> "
echo "Sélectionner l'interface réseau du serveur accessible depuis internet"
select ethtype in "eth0" "eth1" "wlan0"; do

#########################
#### En cas d'erreur ####
#########################
if [ -z "$ethtype" ]
then
   echo "Erreur: entrez un des chiffres proposés." 1>&2

##############################
```

```
#### Cas du premier choix ####
##############################

elif [ "$REPLY" -eq 1 ]
then
   ethtype=eth0
   break

##############################
#### Cas du second choix ####
##############################

elif [ "$REPLY" -eq 2 ]
then
   ethtype=eth1
   break

################################
#### Cas du troisieme choix ####
################################
elif [ "$REPLY" -eq 3 ]
then
   ethtype=wlan0
   break
fi
done

########################################
#### Infos diverses pour le nattage ####
########################################

echo -n "Entrez le numéro de la VE : "
read numero

echo -n "Entrer l'adresse IP du VE : "
read adresseipve

echo -n "Entrer l'adresse IP de la carte réseau lan du serveur : "
read adresseipserv

#########################
#### Saisie des ports ####
#########################

echo -n "Etrez le port de destination sur le VE : "
read portve

echo -n "Etrez le port source (coté serveur) : "
read portserv

########################################
```

```
#### Création de la règles iptables ####
########################################

iptables -t nat -A PREROUTING -p "$porttype" -d "$adresseipserv" --dport
"$portserv"   -i "$ethtype" -j DNAT --to-destination
"$adresseipve":"$portve"

########################################
#### Copie de la règle de nattage ####
########################################

echo "#### Nat sur VE "$numero" port "$adresseipserv":"$portserv" ->
"$adresseipve":"$portve"" >> /root/nat_VE_Reboot.sh
echo "/sbin/iptables -t nat -A PREROUTING -p "$porttype" -d "$adresseipserv"
--dport "$portserv" -i "$ethtype" -j DNAT --to-destination
"$adresseipve":"$portve"" >> /root/natVE.sh
```

- On peut également inscrire le script comme une commande pour root :

```
echo "alias natVE='/root/scripts/natVE.sh'" >> /root/.bashrc
```

- A la prochaine connection de root, il suffira de taper :

```
natVE
```

Pour faire les choses proprement, vous devrez ajouter ces lignes en début de votre script :

```
#!/bin/bash

# Vider les tables actuelles
/sbin/iptables -t filter -F

# Vider les regles personnelles
/sbin/iptables -t filter -X

# Vider les regle de nattage
/sbin/iptables -t nat -X
/sbin/iptables -t nat -F
/sbin/iptables -t mangle -X
/sbin/iptables -t mangle -F

# Ne pas casser les connexions etablies
/sbin/iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

# Interdire toute connexion entrante et sortante
/sbin/iptables -t filter -P INPUT DROP
/sbin/iptables -t filter -P FORWARD ACCEPT
/sbin/iptables -t filter -P OUTPUT ACCEPT

# Autoriser loopback
/sbin/iptables -t filter -A INPUT -i lo -j ACCEPT
```

```
/sbin/iptables -t filter -A OUTPUT -o lo -j ACCEPT

#Autoriser le net au VE
/sbin/iptables -t filter -A INPUT -i venet0 -j ACCEPT
/sbin/iptables -t filter -A OUTPUT -o venet0 -j ACCEPT

#Autoriser tout en sortie
/sbin/iptables -t filter -A OUTPUT -o eth0 -j ACCEPT

# ICMP (Ping)
/sbin/iptables -t filter -A INPUT -p icmp -j ACCEPT
/sbin/iptables -t filter -A OUTPUT -p icmp -j ACCEPT

### Quand les VE accedent au net c'est l'@ip publique qui est vue
/sbin/iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to IP_SERVEUR
```

Clonage d'une VE

Soit 222 l'id de l'ancienne ve et 333 l'id de la nouvelle ve à créer:

- On créé un script contenant :

```
OLDVE=222
NEWVE=333
vzctl stop $OLDVE
mkdir /vz/root/$NEWVE
cp /etc/vz/conf/$OLDVE.conf /etc/vz/conf/$NEWVE.conf
mkdir /vz/private/$NEWVE
pushd /vz/private/$OLDVE; tar c --numeric-owner * | tar x --numeric-owner -C
/vz/private/$NEWVE; popd
vi /etc/vz/conf/$NEWVE.conf # Change the IP_ADDRESS
vzctl start $NEWVE; vzctl start $OLDVE
```

Commandes utiles

```
→ vzlist : liste les VE actives
→ vzlist -a : liste toutes les VE existantes
→ vzctl start $numero : demarre la VE possedant ce $numero
→ vzctl stop $numero : stop la VE possedant ce $numero
→ vzctl restart $numero : redemarre la VE possedant ce $numero
→ vzctl destroy $numero : detruit la VE possedant ce $numero. La VE doit
être au préalable éteinte.
→ vzctl exec $numero ma_commande : exécute ma_commande dans la VE n° $numéro
→ vzctl create $numero –ostemplate $template_name : créé une VE à partir du
$template_name se trouvant dans /vz/template/cache
→ vzctl chkpnt $numero [–dumpfile <name>] : prend un snapshot de la VE
```

```
→ vzctl restore $numero [–dumpfile <name>] : restaure à partir d'un snapshot
```

Création d'une template

Cet exemple est pour une Ubuntu Hardy 8.04

Pre-requis, debootstrap :

```
apt-get install debootstrap
```

- On va créér le répertoire de travail :

```
cd /vz/private
```

```
mkdir hardy-chroot
```

- On lance le debootstrap pour Ubuntu Hardy et on remplace ARCH par notre architecture :

```
debootstrap [--arch ''ARCH''] hardy hardy-chroot
```

- Après le debootstrap, on va déplacer notre répertoire de travail vers un autre portant un numéro valide :

```
mv /vz/private/hardy-chroot /vz/private/777
```

- Tous les fichiers doivent appartenir à root :

```
chown -Rv root /vz/private/777
```

- On configure les paramètres de base de notre container :

```
vzctl set 777 --applyconfig vps.basic --save
```

- Détermination de la template du container :

```
echo "OSTEMPLATE=ubuntu-8.04" | sudo tee -a /etc/vz/conf/777.conf >/dev/null
```

- On affecte une IP à la VE :

```
vzctl set 777 --ipadd x.x.x.x --save
```

- On lui affecte maintenant un serveur DNS pour la résolution de nom :

```
vzctl set 777 --nameserver x.x.x.x --save
```

- On supprime les fichiers qui vont poser problème :

```
rm /vz/private/777/etc/rcS.d/S10udev /vz/private/777/etc/rc2.d/S11klogd
```

- On démarre le container :

```
vzctl start 777
```

- On entre dans le container :

```
vzctl enter 777
```

<note importante>Attention !! Vérifiez bien que vous êtes dans la VE. Je ne serais pas tenu responsable des éventuels dégats que vous causerez sur votre machine autrement.

- On supprime les paquets inutiles :

```
apt-get remove --purge busybox-initramfs console-setup dmidecode eject \
ethtool initramfs-tools klibc-utils laptop-detect libiw29 libklibc \
libvolume-id0 mii-diag module-init-tools ntpdate pciutils pcmciautils
ubuntu-minimal \
udev usbutils wireless-tools wpasupplicant xkb-data tasksel tasksel-data
```

```
apt-get remove --purge --auto-remove dhcp3-client dhcp3-common
```

- Nettoyage de udev :

```
rm -fr /lib/udev
```

- On stoppe les tty et on les suprimme:

```
initctl stop tty1
```

```
initctl stop tty2
```

```
initctl stop tty3
```

```
initctl stop tty4
```

```
initctl stop tty5
```

```
initctl stop tty6
```

```
rm /etc/event.d/tty*
```

```
rm /etc/init/tty*
```

- On met les bons droits sur le /root :

```
chmod 700 /root
```

- On désactive le login root :

```
usermod -p '!' root
```

- On simule le modprobe :

```
ln -s /bin/true /sbin/modprobe
```

- On créé un dépôt minimaliste : COUNTRY=<YOURCOUNTRY>.

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

```
# Binary
deb http://${COUNTRY}archive.ubuntu.com/ubuntu/ hardy main restricted
universe multiverse
deb http://${COUNTRY}archive.ubuntu.com/ubuntu/ hardy-updates main
restricted universe multiverse
deb http://security.ubuntu.com/ubuntu hardy-security main restricted
universe multiverse

# Binary Canonical
# deb http://archive.canonical.com/ubuntu hardy partner

# Binary backport
# deb http://${COUNTRY}archive.ubuntu.com/ubuntu/ hardy-backports main
restricted universe multiverse

# Source
# deb-src http://${COUNTRY}archive.ubuntu.com/ubuntu/ hardy main restricted
universe multiverse
# deb-src http://${COUNTRY}archive.ubuntu.com/ubuntu/ hardy-updates main
restricted universe multiverse
# deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
universe multiverse

# Source backport
# deb-src http://${COUNTRY}archive.ubuntu.com/ubuntu/ hardy-backports main
restricted universe multiverse

# Source Canonical
# deb-src http://archive.canonical.com/ubuntu hardy partner
```

- On met à jour :

```
apt-get update && apt-get upgrade
```

On installe des paquets utiles :

```
apt-get install ssh quota
```

- On fixe un bug sur les clefs ssh :

```
rm -f /etc/ssh/ssh_host_*
```

```
cat << EOF > /etc/rc2.d/S15ssh_gen_host_keys
```

```
#!/bin/sh
ssh-keygen -f /etc/ssh/ssh_host_rsa_key -t rsa -N ''
ssh-keygen -f /etc/ssh/ssh_host_dsa_key -t dsa -N ''
```

```
rm -f \$0
```

```
chmod a+x /etc/rc2.d/S15ssh_gen_host_keys
```

- On simule le mtab :

```
rm -f /etc/mtab
```

```
ln -s /proc/mounts /etc/mtab
```

- On supprime le mtab au boot :

```
update-rc.d -f mtab.sh remove
```

- On supprime les services inutiles au boot :

```
update-rc.d -f klogd remove
```

- On définit le hostname par défaut :

```
echo "localhost" > /etc/hostname
```

- On définit le /etc/hosts :

```
echo "127.0.0.1 localhost.localdomain localhost" > /etc/hosts
```

- On ajoute les ptys à /dev :

```
cd /dev && /sbin/MAKEDEV ptyp
```

- On supprime les serveurs DNS :

```
echo "" > /etc/resolv.conf
```

- On nettoie le cache de apt (éviter de faire une VE lourde) :

```
apt-get clean
```

- On nettoie aussi les logs :

```
echo "" > /var/log/messages; > /var/log/auth.log; > /var/log/kern.log; >
/var/log/bootstrap.log; \
> /var/log/dpkg.log; > /var/log/syslog; > /var/log/daemon.log; >
/var/log/apt/term.log; rm -f /var/log/*.0
/var/log/*.1
```

- Et voilà, on quitte la VE :

```
exit
```

Préparation de la compression du template.

On supprime l'ip temporaire qu'on lui a affecté.

```
vzctl set 777 --ipdel all --save
```

- On stoppe la VE :

```
vzctl stop 777
```

- On se met à l'intérieur de la VE :

```
cd /vz/private/777
```

- Maintenant, on compresse notre template :

```
tar czfv /vz/template/cache/ubuntu-8.04-<arch>-minimal.tar.gz
```

.

- On procède au nettoyage :

```
vzctl destroy 777
```

```
rm -f /etc/vz/conf/777.conf.destroyed
```

Proxmox

Introduction

Proxmox Virtual Environment est un logiciel libre de virtualisation, plus précisément un hyperviseur de machine virtuelle. Il est développé et maintenu par Proxmox Server Solutions GmbH avec un support financier de l'Internet Foundation Austria (IPA).

Proxmox VE installe les outils complets du système d'exploitation et de gestion en 3 à 5 minutes (dépend du matériel utilisé).

Ce logiciel est un « système à nu ». Le terme de « système à nu » signifie que vous commencez à partir d'un serveur vide et qu'il n'y a donc nul besoin d'installer un système d'exploitation auparavant.

Le logiciel inclut :

- Système d'exploitation complet (Debian Lenny 64 bits)
- Partitionnement de disque dur avec LVM2
- Noyau Proxmox VE avec support de OpenVZ et KVM
- Outils de sauvegarde et de restauration
- Interface d'administration par le web

source : Wikipédia on Proxmox

Bon, plus simplement, ça va vous permettre de faire tout ce que vous faisiez avec OpenVZ (voir même plus vu que ça gère du KVM et du Qemu). Screenshots

Ca vaut toujours mieux qu'un long discours :

Informations générales :

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !