

# OpenVZ sur Debian GNU/Linux 5.0 (Lenny)

Agence universitaire de la Francophonie

26 mars 2009

# Plan

- 1 Installation de l'hôte
- 2 «LVMisation»
- 3 Réseautique

- 1 Installation de l'hôte
- 2 «LVMisation»
- 3 Réseautique

# Partitionnement : LVM is your friend

## Proposition de partitionnement

- / 1Go sur partition "classique"
- *swap* sur partition "classique"
- /var 2Go sur volume logique (LVM)
- /var/lib/vz 5Go sur volume logique (LVM)

Ce n'est qu'une proposition... Il n'y a jamais de partitionnement "à tous les coups on gagne"

# Sources

## Classiques

```
deb http://miroir.proche/debian lenny main
deb http://security.debian.org/ lenny/updates main
deb http://volatile.debian.org/debian-volatile
lenny/volatile main
```

## Spécifique pour un "vzctl" plus moderne (CT et bridge)

```
deb http://debian.sysfs.org/debian lenny openvz
```

## Et c'est parti !

### Installation noyau patché et outils userland

- `aptitude install linux-image-openssl-686`
- `-686` : quel que soit le processeur (Intel, AMD, etc...)
- installe également `vzctl` par dépendance, depuis [debian.systems.org](http://debian.systems.org)<sup>a</sup>

---

<sup>a</sup>la version `vzctl` proposée par `systems` utilise le vocable "CT" (au lieu de l'ancien VE) et intègre la gestion des bridges, entre autres améliorations

... et on passe en VZ !

reboot ... au démarrage, on vérifie que le choix du noyau `openssl` est bien premier listé sur le menu du démarreur `grub`

# Recherche de templates

## Les templates

- fichiers `.tar.gz` à placer dans `/var/lib/vz/templates/cache`
- source : <http://download.openvz.org/template/>
- mais pour créer un modèle Debian, `debootstrap` est très simple à utiliser

- 1 Installation de l'hôte
- 2 «LVMisation»
- 3 Réseautique



# LVMisation

`vzctl create` ne sait pas créer une machine dans un répertoire existant : on va créer la machine puis la déplacer dans une partition

## LVMisation d'un CT

- `vzctl create nnn --ostemplate ...`
- `cd /var/lib/vz/private/ ; mv nnn nnn.orig`
- `lvcreate -L 2G -n machine-root vg0  
mkfs.ext3 /dev/vg0/machine-root`
- ajout dans le `fstab` pour montage sur  
`/var/lib/vz/private/nnn`
- `mkdir /var/lib/vz/private/nnn ; mount -a`
- `cd /var/lib/vz/private/ ; mv nnn.orig/* nnn/`

Ouf...

## Sous-partitions en LVM (1/2)

### Sous-partitions LVM : exemple d'un /var

- préliminaire : CT stoppé
- `lvcreate -L 5G -n machine-var vg0  
mkfs.ext3 /dev/vg0/machine-var`
- puis déplacement des données vers cette partition
  - `mount /dev/vg0/machine-var /mnt`
  - `mv private/nnn/var/* /mnt`
  - `umount /mnt`
- très utile : ajout dans le fstab pour check au démarrage (point de montage *none*, options *noauto*, pass 2)

## Sous-partitions (2/2) : script de mount

### Script `/etc/vz/conf/nnn.mount`

```
#!/bin/sh  
mount -n /dev/vg0/machine-var /var/lib/vz/root/nnn/var a
```

---

<sup>a</sup>Attention à ne pas oublier l'option `-n` du `mount`, elle a son importance pour éviter des soucis de démontage (`umount`) lorsqu'on arrête le container

Et ne pas oublier de rendre le script exécutable :

```
chmod +x nnn.mount
```

- 1 Installation de l'hôte
- 2 «LVMisation»
- 3 Réseautique**

## Réseautique classique

### Réseautique «classique»

`vzctl set nnn --ipadd ...` : donne une adresse IP au container *nnn*

### Conséquence sur l'hôte

- l'hôte est une passerelle
- il a une adresse IP vers «Internet»
- permettre le *forward* IP :  
echo 1 > /proc/sys/net/ipv4/ip\_forward  
et/ou net.ipv4.ip\_forward=1 dans /etc/sysctl.conf  
(Debian)
- si nécessaire, SNAT (voire MASQUERADE) via *netfilter*
- dans l'autre sens : DNAT si les CT doivent offrir des services «directement»

## Réseau en bridge (1/2)

Activation de la gestion bridge (vzctl >= 3.0.23)

Création d'un fichier `/etc/vz/vznet.conf` avec cette seule ligne :  
`EXTERNAL_SCRIPT=/usr/sbin/vznetaddr`

Création d'un bridge sur l'hôte

```
# dans /etc/network/interface sur l'hôte
auto vz0
  iface vz0 inet manual
    bridge_ports eth0
```

Note : pour avoir une IP sur l'hôte, il faut l'attribuer à l'interface bridge (vz0 dans l'exemple)

## Réseau en bridge (2/2)

### Configuration d'un CT

```
vzctl set 101 --netif_add  
"eth0,00:E1:00:00:10:1A,veth101.1,00:E1:00:00:10:1B,vz0"  
--savea
```

---

<sup>a</sup>je préfère faire l'opération sur un CT arrêté, et vérifier le fichier `/etc/vz/conf/nnn.conf` créé avant de faire un start

Une fois le CT démarré, il dispose d'une interface `eth0` classique, connectée au même réseau Ethernet que l'interface de l'hôte reliée au bridge.