

Xen

Motivation

Lorsqu'on est un peu courts en boxes (et qu'en empiler 4 sur le bureau n'est pas nécessairement l'idéal en terme tant de bruit que de conso électrique, cablage toussa) j'ai écrit une procédure destinée à mettre en place une machine dédiée à la virtualisation.

Le but final est de pouvoir mettre à disposition un OS rapidement, avec si possible des sélections de packages prédéfinies.

Procédure d'installation

Installer une sarge tout ce qu'il y a de plus classique puis updater en sid tout de suite le sources.list :

```
deb http://ftp2.fr.debian.org/debian/ sid main
deb-src http://ftp2.fr.debian.org/debian/ sid main
```

puis faire :

```
apt-get install xen-hypervisor-3.0-i386 xen-utils-3.0 iproute bridge-utils libc6-xen kernel-pac
```

ajouter :

```
title Xen 3.0 / XenLinux 2.6
kernel /boot/xen-3.0-i386.gz
module /boot/xen-linux-2.6.16-xen root=/dev/hda1 ro
```

après "### END DEBIAN AUTOMAGIC KERNELS LIST" dans /boot/grub/menu.lst

ensuite on va se builder un kernel kivabien [™] à partir des sources vanilla. Prendre le 2.6.16 [http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.16.tar.bz2] et le patch ici [https://alioth.debian.org/download.php/1561/linux-2.6.16-xen3.0.2-hg9629.patch.gz] puis :

```
tar -jxvf linux-2.6.16.tar.bz2
gunzip linux-2.6.16-xen3.0.2-hg9629.patch.gz
cp linux-2.6.16-xen3.0.2-hg9629.patch linux-2.6.16
cd linux-2.6.16
patch -p1 < linux-2.6.16-xen3.0.2-hg9629.patch
make menuconfig
```

Ensuite on génère le paquet debian convivial :

```
make-kpkg --append-to-version -xen kernel_image
```

Pendant que ça compile on peut aller matter les dernières quotes bashfr [http://www.bashfr.org/?sort=latest]. On installe le noyau :

```
dpkg -i linux-xen0-2.6.16-xen_10.00.Custom_i386.deb
```

Rebooter sur le nouveau kernel, croiser les doigts :X (j'oublie toujours une option)

Normalement xend est lancé, il reste donc à créer les VMs

On va utiliser le backend 'fichier' (il est aussi possible d'utiliser une partition pour faire ça) :

```
dd if=/dev/zero of=/srv/domains/VM.img bs=1M count=2000
mkfs.ext3 /srv/domains/VM.img
```

(passer outre les protestations de mkfs parce que c'est un fichier)

```
mount -o loop /srv/domains/VM.img /mnt/xen
```

Et on bootstrap une sarge :

```
debootstrap sarge /mnt/xen ftp://ftp2.fr.debian.org/debian
```

On lui donne le bon noyau :

```
cp linux-xen0-2.6.16-xen_10.00.Custom_i386.deb /mnt/xen/root
```

Et hop ! chroot !

```
chroot /mnt/xen /bin/bash
```

les commandes suivantes sont tapées _dans_ le chroot (enfin après si vous voulez br0tcher la machine ça vous regarde hein) On installe le nouveau kernel pour la VM

```
cd /root && dpkg -i linux-xen0-2.6.16-xen_10.00.Custom_i386.deb
```

Le minimum vital :

```
apt-setup
```

Choisir un miroir pas trop loin, histoire de pas sentir que le link est roumain. Ensuite :

```
apt-get install localeconf
```

Choisir les choix par défaut, à priori c'est bon (perso c'est en_US-ISO-8559-15) Si il propose d'upgrader la glibc, accepter.

Ensuite on passe à la conf système basique :

/etc/fstab :

```
/dev/hda1 / ext3 defaults,errors=remount-ro 0 1
```

/etc/hostname :

```
nom_de_la_vm
```

/etc/resolv.conf :

```
search votre domaine
nameserver 192.168.100.253
```

/etc/network/interfaces :

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.100.98
    netmask 255.255.255.0
    network 192.168.100.0
    broadcast 192.168.100.255
```

```
gateway 192.168.100.253
```

J'ai rencontré un problème avec les getty, voici comment j'ai fait mon /etc/inittab :

```
1:2345:respawn:/sbin/getty 38400 console
#2:23:respawn:/sbin/getty 38400 tty2
#3:23:respawn:/sbin/getty 38400 tty3
#4:23:respawn:/sbin/getty 38400 tty4
#5:23:respawn:/sbin/getty 38400 tty5
#6:23:respawn:/sbin/getty 38400 tty6
```

(avec *console* à la place de *tty1*)

Installer l'indispensable SSH :

```
apt-get install ssh module-init-tools
```

Désactiver le tls

```
mv /lib/tls /lib/tls.disabled
```

Et un truc qui peut être utile :

```
passwd root
```

Et hop !

```
Ctrl-D
```

fin de la partie dans le chroot

Maintenant on configure un peu le dom0 :

/etc/xen/VM

```
# Kernel to use
kernel = "/boot/xen0-linux-2.6.16-xen"
# Memory in megabytes
memory = 128
# Your domain's name
name = "nom_de_la_vm"
# Root device
root = "/dev/hda1 ro"
# Don't forget to specify your bridge correctly here
vif = [ 'mac=aa:00:00:00:00:d2, bridge=br-xen' ]
# Disk
disk = [ 'file:/srv/domains/VM.img,hda1,w' ]
```

et le réseau (sur le dom0) : /etc/network/interfaces

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 0.0.0.0
    netmask 0.0.0.0

# the bridge for xen
auto br-xen
iface br-xen inet static
    address 192.168.100.134
    netmask 255.255.255.0
    gateway 192.168.100.253
    bridge_ports eth0
    bridge_maxwait 0
```

On recharge le réseau :

```
/etc/init.d/networking restart
```

Le moment que vous attendiez tous

```
xm create -c VM
```

enjoy !

Votre VM pendant la pause café

Répéter la procédure peut être fastidieux, source d'erreurs tout ça. Alors j'ai fait un petit script, histoire que ça s'installe pendant la pause café.

```
#!/bin/bash
default_place="/srv/domains/"
echo "Nom de la nouvelle VM :"
read name
echo "IP de la nouvelle VM :"
read ip
echo "Gateway de la nouvelle VM:"
read gateway
echo "Adresse MAC de la nouvelle VM :"
read mac
echo "Taille de la nouvelle VM (Mb) :"
read size
echo "Mémoire de la nouvelle VM (Mb) :"
read ram
echo "Distribution à utiliser ? (sarge/etch) :"
read version

echo "Emplacement du package pour le kernel xen :"
read kernel
kernel_file=`basename ${kernel}`

echo "Point de montage de l'image :"
read mount_point

# Creation image
echo "Creation de l'image disque"
dd if=/dev/zero of=${default_place}${name}.img bs=1M count=${size}
yes | mkfs.ext3 ${default_place}${name}.img

# Montage
mount -o loop -t ext3 ${default_place}${name}.img ${mount_point}

# Bootstrap
debootstrap ${version} ${mount_point} ftp://ftp.fr.debian.org/debian

# Copie noyau
cp ${kernel} ${mount_point}/root

# Installation kernel
chroot ${mount_point} /usr/bin/dpkg -i /root/${kernel_file}

# choix du miroir
chroot ${mount_point} apt-setup

# Les locales
chroot ${mount_point} /usr/bin/apt-get -y install localeconf

## Les fichiers indispensables au système ##
# /etc/fstab
echo "/dev/hda1      /                  ext3      defaults,errors=remount-ro 0      1" > ${mount_point}/etc/fstab
# /etc/hostname
echo "${name}" > ${mount_point}/etc/hostname
```

```

# /etc/resolv.conf
cp /etc/resolv.conf ${mount_point}/etc/resolv.conf

# /etc/network/interfaces
echo "auto lo" > /tmp/${name}.if.tmp
echo "iface lo inet loopback" >> /tmp/${name}.if.tmp
echo "" >> /tmp/${name}.if.tmp
echo "auto eth0" >> /tmp/${name}.if.tmp
echo "iface eth0 inet static" >> /tmp/${name}.if.tmp
echo "    address ${ip}" >> /tmp/${name}.if.tmp
echo "    netmask 255.255.255.0" >> /tmp/${name}.if.tmp
echo "    #network " >> /tmp/${name}.if.tmp
echo "    #broadcast " >> /tmp/${name}.if.tmp
echo "    gateway ${gateway}" >> /tmp/${name}.if.tmp

cp /tmp/${name}.if.tmp ${mount_point}/etc/network/interfaces
rm -f /tmp/${name}.if.tmp

# /etc/inittab (un peu sale [tres] comme méthode)
cp /etc/inittab /tmp/${name}.inittab

sed -e 's/tty1/console/g' /tmp/${name}.inittab > /tmp/${name}.inittab.2
sed -e 's/2:23/#2:23/g' /tmp/${name}.inittab.2 > /tmp/${name}.inittab
sed -e 's/3:23/#3:23/g' /tmp/${name}.inittab > /tmp/${name}.inittab.2
sed -e 's/4:23/#4:23/g' /tmp/${name}.inittab.2 > /tmp/${name}.inittab
sed -e 's/5:23/#5:23/g' /tmp/${name}.inittab > /tmp/${name}.inittab.2
sed -e 's/6:23/#6:23/g' /tmp/${name}.inittab.2 > /tmp/${name}.inittab

cp /tmp/${name}.inittab ${mount_point}/etc/inittab
rm /tmp/${name}.inittab /tmp/${name}.inittab.2

# ssh
chroot ${mount_point} /usr/bin/apt-get -y install ssh module-init-tools

# desactivation du TLS
mv ${mount_point}/lib/tls ${mount_point}/lib/tls.disabled

# mot de passe root
echo "Saisie du mot de passe root pour la VM ${name}:"
chroot ${mount_point} /usr/bin/passwd root

# demontage
umount ${mount_point}

# Creation du fichier de config de la VM
echo "kernel = \"/boot/xen0-linux-2.6.16-xen\"" > /etc/xen/${name}
echo "memory = ${ram}" >> /etc/xen/${name}
echo "name = \"${name}\"" >> /etc/xen/${name}
echo "root = \"/dev/hda1 ro\"" >> /etc/xen/${name}
echo "vif = [ 'mac=${mac}, bridge=br-xen' ]" >> /etc/xen/${name}
tmp_name=${default_place}${name}.img
echo "disk = [ 'file:${tmp_name},hda1,w' ]" >> /etc/xen/${name}

echo "La VM est prête, lancez la avec :"
echo "xm create ${name}"

```

Les xen-tools [<http://www.steve.org.uk/Software/xen-tools/>] font la même chose, en mieux, alors ne vous cassez pas la tête. Ce script a juste un but didactique

Les erreurs cons

- “Error: Device 0 (vif) could not be connected” : il manque probablement le bridge indiqué dans la config de xen.
- “Error: Device 769 (vbd) could not be connected” : le module “loop” n’est pas chargé, ou il n’y a plus de loop-devices de disponibles.

Sources principales

- <http://julien.danjou.info/xen.html> [<http://julien.danjou.info/xen.html>]
- http://www.howtoforge.com/perfect_xen_setup_debian_ubuntu [http://www.howtoforge.com/perfect_xen_setup_debian_ubuntu]

Changelog

- 24/05 : Création par __nico
- 26/05 : Ajout du script de création facilitée

linux/xen.txt · Last modified: 2006/07/06 14:55 by nico