http://en.wikipedia.org/wiki/Xen

INSTALLATION

root@serv1:~# apt-get install linux-image-2.6.26-2-xen-amd64 xen-hypervisor-amd64 xen-tools xen-linux-system-2.6.26-2-xen-amd64 linux-headers-2.6.26-2-xen-amd64 bridge-utils

add the following line to /etc/modules

loop max_loop=64

Reboot the server and start the hypervisor from the boot menu (Xen 3.2-1-amd64 / Debian GNU/Linux, kernel 2.6.26-2-xen-amd64)

CONFIGURATION

Edit /etc/xen/xend-config.xsp

#fundamental
(network-script network-bridge)
(vif-script vif-bridge)

#not sure if fundamental
(dom0-min-mem 196)
(dom0-cpus 0)
(vncpasswd '')

File /etc/xen/xen-tools.conf is not really useful for us at the moment, except maybe for the RAM (memory) dimension. In general, default options are indicated in that file (id-est options valid for any DomU created by xen-tools; see below for details).

Add:

extra = 'xencons=tty1'

to

/etc/xen-tools/xm.tmpl

CREATION OF VIRTUAL MACHINES

Method 2: Use of xen-tools

This method has been used to create some virtual machines (DomUs) (Debian Lenny, Ubuntu Jaunty, Centos5, Fedora11, Slackware13, Windows XP and Seven, and more) with different success degrees.

We decided not to remove any of these virtual machines referred documentation, in order to let users (and ourselves) have a sort of history of the different tests performed, and try themselves. Then BE AWARE that: - the underlined procedures can or cannot be still useful - the somewhere DomUs "vdd-personalized" .tar archive files, previously available on this site, are not anymore. We apologizes for that.

By the way we still continue to use some of these virtual machines, but we also have in use some "img"+"pygrub" style virtual machines (see below for details).

The local network dhcp-server is configured to assign an IP address based on the VM MAC address. The dhcp-server can be the ltsp server's one (id est the main VDD server's one) (see "LTSP how-to" section). Adjust MAC address according to your choices.

Debian Lenny DomU

```
root@serv1:~# cd /etc/xen-tools/partitions.d
root@serv1:/etc/xen-tools/partitions.d# nano lennyvm
[root]
size=5G
type=ext3
mountpoint=/
options=sync,errors=remount-ro
```

```
[swap]
size=1G
type=swap

[home]
size=6G
type=ext3
mountpoint=/home
options=nodev,nosuid
```

At this point we can create the DomU using lvm devices in "serv1" volume group

```
root@serv1:~# xen-create-image --lvm=serv1 --hostname=lennyvm --mac=00:16:3E:5F:07:10
--dhcp --dist=lenny
--role=udev --passwd --partitions=lennyvm
```

Ubuntu Jaunty DomU

The partition file is the same as for the Lenny DomU. So you can just copy it in a file named jauntyvm and store it in the same folder. Before creating the DomU it has been necessary to do the following:

```
root@serv1:~# ln -s /usr/lib/xen-tools/edgy.d /usr/lib/xen-tools/jaunty.d
root@serv1:~# ln -s /usr/share/debootstrap/scripts/edgy /usr/share/debootstrap/scripts/jaunty
```

At this point we can create the DomU using lvm devices in "serv1" volume group

```
root@serv1:~# xen-create-image --lvm=serv1 --hostname=jauntyvm --mac=00:16:3E:5F:07:11
--dhcp --dist=jaunty --role=udev
 --passwd --partitions=jauntyvm
```

Centos 5 DomU

The partition file is again the same as for the Lenny DomU. Before to create the DomU it has been necessary to add some packages to the default list. So edit /etc/rinse/centos5-packages and add the following lines

```
nss
nspr
python-iniparse
passwd
```

At this point we can create the DomU using lvm devices in "serv1" volume group

```
root@serv1:~# xen-create-image --lvm=serv1 --install-method=rinse --dist=centos-5
--hostname=centos5vm --mac=00:16:3E:5F:07:12
 --dhcp --passwd --partitions=centos5vm
```

Fedora 11 DomU

For this VM we followed a slightly different method. Beginnig from here http://doc.cheramy.name/xen_tools_installation_fedora_11_domu
and here
http://www.stacklet.com/downloads/images/fedora/11
, we built a tar archive with a tuned fedora installation and then used "tar" as the install-method option of xen-create-image. So you can get our tar archive and then launch the xen-create-image command, just as like in the following. Be aware to modify the $PATH according to where you stored the tar archive. Usual content for fedora11vm partitions file.

```
root@serv1:~# cd /usr/lib/xen-tools
root@serv1:~# ln -s fedora-core-6.d fedora-11.d
root@serv1:~# wget http://www.vdd-project.org/fedora/fedora11_mod.tar
root@serv1:~# xen-create-image --force --lvm=serv1 --install-method=tar
--install-source=$PATH/fedora11_mod.tar
--dist=fedora-core-11 --hostname=fedora11vm --mac=00:16:3E:5F:07:13 --dhcp
--partitions=fedora11vm --passwd
```

Still we have some working issues on this DomU ...

Slackware 13 DomU

For this VM too, we followed a similar method as for fedora, this time beginning from here http://www.stacklet.com/downloads/images/slackware/13.0
and then modifying the downloaded archive to tune the installation. Usual content for slackware13vm partitions file and same warning as above about modifying the $PATH.

```
root@serv1:~# wget http://www.vdd-project.org/slackware/slackware13_mod.tar
root@serv1:~# xen-create-image --force --lvm=serv1 --install-method=tar
--install-source=$PATH/slackware13_mod.tar -
-hostname=slackware13vm --mac=00:16:3E:5F:07:16 --dhcp --partitions=slackware13vm
```

Windows XP (and Windows Seven) DomU

As for building a Windows XP (or Seven) VM in XEN, stuff is somewhat different, above all because you need to use a different build method (linux kernel obviously doesn't fit for windows!). Still using xen-tools here but to build a full-virtualized machine instead of a para-virtualized one (as in previous, "linux-based", cases).

In building Windows XP or Seven DomU we referred to some useful links. Take a look at them to get some background informations about "hvm" and "ioemu". http://www.ezunix.org/index.php?title=Windows_XP_in_XEN
http://www.valent-blog.eu/2008/01/13/full-virtualization-di-windows-con-xen/
http://wiki.openlab-dist.org/index.php/Macchine_virtuali_Windows_su_XEN
http://lists.xensource.com/archives/html/xen-users/2006-06/msg00599.html
http://mediakey.dk/~cc/howto-install-windows-xp-vista-on-xen/

Here are some useful "warnings" about Debian Lenny Xen and "hvm": http://old.nabble.com/xen,debian-lenny-and-hvm(winxp)-td21033712.html

Requirements are: - a licensed and working Windows XP Pro or Seven OS (either in CD or .iso format) (be aware that installing a Microsoft Windows "virtual" machine legally requires the same license needed for a "real" one) - a CPU with Intel's VM or AMD's AMD-V technology (a xm dmesg | grep VMX in command line will tell you if your CPU is supporting that)

NOTE

As far as typical Windows installations can be used by just one user per-time, you'll need to create more Windows virtual machines. LVM snapshot utility will help you to do that, of course if your virtual machines are on LVs.

Let's see it for our "winxpvm" DomU, but we could do the same thing with any DomU. Let's assume here that you have an already Windows DomU named winxpvm, that uses a 6 Gb sized LV on VG named serv1; we want to exactly (software, users, configurations, etc.) copy that on a brand new DomU, named winxpvm2.

```
root@serv1:~# lvcreate -s -L 6G -n winxpvm_snap /dev/serv1/winxpvm
root@serv1:~# lvcreate -L 6G -n winxpvm2 serv1
root@serv1:~# dd if=/dev/serv1/winxpvm_snap of=/dev/serv1/winxpvm2 bs=1024k
root@serv1:~# cp /etc/xen/winxpvm.cfg /etc/xen/winxpvm2.cfg
```

That's it. Be aware of editing winxpvm2.cfg with correct hostname, disk and MAC address

(should be different from winxpvm).

Once you have the winxpvm_snap "LVM snapshot" volume, you can copy that again and again to create clones of the first Windows virtual machine. You also can "clone" on a different Dom0 (id est, a different xen/vdd server), for example from serv1 to serv2 (wher VG is called serv2).

```
  root@serv2:~# lvcreate -L 6G -n winxpvm serv2
root@serv1:~# dd if=/dev/serv1/winxpvm_snap bs=1024k | gzip -c | ssh serv2 "gunzip -c | dd
of=/dev/serv2/winxpvm bs=1024k"
root@serv1:~# scp /etc/xen/winxpvm.cfg serv2:/etc/xen/
```

Always remember to edit the winxpvm.cfg according to the context.

Method 3: Use of "OS img+pygrub"

This virtual machine building method is somewhat easier and more "universal". You need a linux-based OS .img file on your hard disk (and enough space on it), the grub package installed and a correctly configured .cfg file for each DomU. Basically this method, using a xen bootloader named "PyGrub" http://wiki.xensource.com/xenwiki/PyGrub , let a DomU boot with its own kernel instead of using the Dom0 kernel, as if it was a "real" machine that run grub bootloader. Furthermore, we used here "img" instead of "lvm" method to store DomUs, because we use existing and right img file, but, if doing the whole thing by yourself, you could use lvm + pygrub too (and maybe it would be better).

See http://wiki.debian.org/PyGrub for a few details in using PyGrub on Debian Lenny.

So the following are the steps to build a virtual machine using an existing OS img and pygrub tool: - get the desired OS img file (we downloaded it from http://www.stacklet.com) of course you also can create your personalized img file starting from a real or virtual OS installation, but be aware that the kernel should be a specific xen-DomU one; in stacklet site the OS img are equipped with that)

so, as an example:

```
  root@serv1:~# wget
http://stacklet.com/sites/default/files/centos/centos.5-4.x86-64.20091107.img.tar.bz2
```

root@serv1:~# tar xvjf centos.5-4.x86-64.20091107.img.tar.bz2

- then copy and edit the pygrub style .cfg in /etc/xen/ directory (we used the one that comes with the img file in the tar.bz2 archive) or just create it

root@serv1:~# cp centos.5-4.x86.xen3.pygrub.cfg /etc/xen/centos5vm.cfg
root@serv1:~# nano /etc/xen/centos5vm.cfg

```
bootloader = "/usr/lib/xen-3.2-1/bin/pygrub"
memory = 512
name = "centos5vm"
vif = [ 'mac = 00:16:3E:5F:07:12' ]
disk = ['file:/$PATH/centos.5-4.x86-64.gnome.img,sda1,w']
root = "/dev/sda1"
extra = "fastboot"
```

* be aware of the pygrub "bootloader" path and the "disk" $PATH (edit with your effective path)

- finally launch the DomU as usual with the following command; a grub-like screen will show, by wich you can choose the desired kernel to boot, if more than one.

root@serv1:~# xm create /etc/xen/centos5vm.cfg -c

FILE .CFG FOR LAUNCHING DOMU

Method 2: Use of xen-tools

The above commands in method 2 (xen-tools) install the DomUs on "serv1" volume group logical volumes and create a .cfg file for any DomU for launching it.

As an example, the following is the resulting .cfg files for Lenny DomU. The other files are very similar to this, except for the Windows XP DomU one (see above, "Windows XP DomU").

Debian Lenny DomU .cfg file

root@serv1:~# cat /etc/xen/lennyvm.cfg

```
#
# Configuration file for the Xen instance lennyvm, created
# by xen-tools 3.9
#

#
```

```
#  Kernel + memory size
#
kernel      = '/boot/vmlinuz-2.6.26-2-xen-686'
ramdisk     = '/boot/initrd.img-2.6.26-2-xen-686'
memory      = '512'

#
#  Disk device(s).
#
root        = '/dev/sda2 ro'
disk        = [
        'phy:/dev/serv1/lennyvm-swap,sda1,w',
              'phy:/dev/serv1/lennyvm-root,sda2,w',
              'phy:/dev/serv1/lennyvm-home,sda3,w',
             ]


#
#  Hostname
#
name        = 'lennyvm'

#
#  Networking
#
dhcp        = 'dhcp'
vif         = [ 'mac=00:16:3E:5F:07:10' ]

#
#  Behaviour
#
on_poweroff = 'destroy'
on_reboot   = 'restart'
on_crash    = 'restart'
extra = 'xencons=tty1'
```

With the following command we launch the DomU and access its console (with the -c flag).

```
  root@serv1:~# xm create /etc/xen/lennyvm.cfg -c
```

Once the DomU has been launched one can access to it both by ssh and by xen console

```
  root@serv1:~# ssh lennyvm
root@serv1:~# xm console lennyvm
```

To see the status of the DomUs use:

```
root@serv1:~# xm list
Name                        ID   Mem VCPUs      State   Time(s)
Domain-0                     0  6576     8     r-----   1181.0
centos5vm                    1   512     1     -b----     34.1
jauntyvm                     2   512     1     -b----    270.5
lennyvm                      3   512     1     -b----    119.2
```

Once accessed a DomU by xm console $DomU, use the combination ctrl +] to return to the Dom0 prompt console.

Other useful xm commands are:

xm shutdown $ID or $DomU to shutdown a DomU xm reboot $ID or $DomU to reboot a DomU xm destroy $ID or $DomU to force the shutdown of a DomU

VIRTUAL MACHINES TUNING

Some tuning (both in command line and in GUI environment) has been necessary on DomUs once they have been created. Here some fundamental tuning in command line env is outlined. Even if not specified, always remember to add one or more normal user(s) to system, with adduser $user, and to modify its groups membership with adduser or usermod (see the man pages for how to use those commands).

Debian Lenny DomU tuning

Add the following line to /etc/apt/sources.list

```
deb http://www.debian-multimedia.org lenny main
```

Then update and upgrade the system, install missing packages (is obviously at your choice) and configure locales (in these examples locales are all for Italy):

```
root@lennyvm:~# apt-get install debian-multimedia-keyring
root@lennyvm:~# apt-get update
```

root@lennyvm:~# apt-get upgrade
root@lennyvm:~# apt-get install rcconf nano hal ntpdate
root@lennyvm:~# apt-get install gnome-desktop-environment kde kde-i18n-it xfce
root@lennyvm:~# apt-get install iceweasel icedove iceweasel-l10n-it icedove-l10n-it
root@lennyvm:~# dpkg-reconfigure tzdata

Ubuntu Jaunty DomU tuning

Update and upgrade the system, install missing packages and configure locales:

root@jauntyvm:~# apt-get install rcconf nano hal ntpdate
root@jauntyvm:~# apt-get install gnome-desktop-environment kde kde-i18n-it
root@jauntyvm:~# apt-get install firefox firefox-3.5-gnome-support human-theme language-selector language-pack-gnome-it language-selector-qt
root@jauntyvm:~# apt-get install xfce4 xubuntu-default-settings usplash-theme-ubuntu
root@jauntyvm:~# dpkg-reconfigure tzdata

Centos 5 DomU and Fedora 11 DomU tuning

Use the yum package manager to add desired application. In particular install Gnome, KDE, XFCE and X Window meta package using the yum grouplist and yum groupinstall <..> utilities. Configure local time with tzselect.

{backbutton}