The Perfect Xen Setup For Debian And Ubuntu - Page 3

3.2 Installing From The Sources

Run the following commands:

```
apt-get remove exim4 exim4-base lpr
nfs-common portmap pidentd pcmcia-cs pppoe
pppoeconf ppp pppconfig
apt-get install iproute bridge-utils python-
twisted gcc-3.3 binutils make libcurl3-dev
zlib1g-dev python-dev transfig bzip2 screen
ssh debootstrap libcurl3-dev libncurses5-dev
(1 line!)
cd ~
mkdir xen
cd xen
mkdir archive
cd archive
```

3.2.1 Install Xen

Now execute these commands:

```
wget http://www.kernel.org/pub/linux/kernel
/v2.6/linux-2.6.11.12.tar.bz2
wget http://www.cl.cam.ac.uk/Research
/SRG/netos/xen/downloads/xen-2.0.7-src.tgz
cd ..
```

```
tar -xvzf archive/xen-2.0.7-src.tgz
tar -xvjf archive/linux-2.6.11.12.tar.bz2

cd xen-2.0
make world

make install
sh ./install.sh
```

Now Xen is installed. In order to start the Xen services at boot time, do the following:

```
update-rc.d xend defaults 20 21
update-rc.d xendomains defaults 21 20
```

3.2.2 Compile A New *dom0* Kernel

Next we compile a new *dom0* kernel with Xen-, *iptables*-, *quota*-, and *dummy* support. The kernel will be installed to *~/xen/install* so that we can save it for other machines. From *~/xen/install* we will install it to the real locations. Xen works with kernel 2.6.11, so I take the latest 2.6.11 kernel (*2.6.11.12*) instead of newer kernels, e.g. 2.6.14.

```
cd ..
mv /lib/tls /lib/tls.disabled
cp -al linux-2.6.11.12 linux-2.6.11.12-xen0
cd xen-2.0/linux-2.6.11-xen-sparse/
./mkbuildtree ../../linux-2.6.11.12-xen0/
cd ../..
echo "-xen0" > linux-2.6.11.12-
xen0/localversion-xen
diff -Naur linux-2.6.11.12 linux-
2.6.11.12-xen0 > archive/linux-2.6.11.12-
xen0.patch

cd linux-2.6.11.12
patch -p1 < ../archive/linux-2.6.11.12-
xen0.patch
cd ../
mkdir install
cd linux-2.6.11.12-xen0
cp arch/xen/configs/xen0_defconfig .config
ARCH=xen; INSTALL_PATH=../install;
INSTALL_MOD_PATH=../install
export ARCH INSTALL_PATH INSTALL_MOD_PATH
make oldconfig
make menuconfig
```

In the kernel configuration menu that shows up we

have to enable *quota*, *iptables* and the *dummy* network driver as **modules**. This is where you enable these modules:

```
File systems --> [*] Quota support
<M> Old quota format support
<M> Quota format v2 support

Device Drivers ---> Networking support --->
<M> Dummy net driver support

Device Drivers ---> Networking support --->
Networking options ---> [*] Network packet
filtering (replaces ipchains) ---> IP:
Netfilter Configuration ---> <M> IP tables
support (required for filtering/masq/NAT)
```

[*] means: build into the kernel statically.
<M> means: build as a kernel module.

Now we install the kernel to *~/xen/install*:

```
make modules
make modules_install
make install
```

Finally, we copy the kernel to the "real" locations:

```
cd ../install/
cp boot/* /boot
cp -r lib/modules/2.6.11.12-xen0/
/lib/modules
cp -r usr/include/xen /usr/include
```

(If you are interested, this is my dom0 kernel configuration.)

3.2.3 Configure The Bootloader And Reboot

Next we add our new kernel to *Grub*, our bootloader. Edit */boot/grub/menu.lst*, and before the line *### BEGIN AUTOMAGIC KERNELS LIST* add the following stanza:

```
title        Xen 2.0.7 / XenLinux 2.6
.11.12-xen0
root        (hd0,0)
kernel        /xen.gz dom0_mem=65536
module        /vmlinuz-2.6.11.12-xen0
 root=/dev/hda6 ro console=tty0
```

Make sure that */dev/hda6* is your */* partition. Keep in mind what I said about *Grub* and partitioning in chapter 1!

Now reboot the system:

```
shutdown -r now
```

At the boot prompt, *Grub* should now list *Xen 2.0.7 / XenLinux 2.6.11.12-xen0* as the first kernel and boot it automatically. If your system comes up without problems, then everything is fine!

3.2.4 Compile A New *domU* Kernel

Now we compile a new *domU* kernel for our virtual machines. Again, we install the kernel in *~/xen/install* and copy it to the real location afterwards:

```
cd ~/xen
cd linux-2.6.11.12
make clean
cd ../
cp -al linux-2.6.11.12 linux-2.6.11.12-xenU
cd xen-2.0/linux-2.6.11-xen-sparse/
./mkbuildtree ../../linux-2.6.11.12-xenU/
cd ../..
echo "-xenU" > linux-2.6.11.12-
xenU/localversion-xen
diff -Naur linux-2.6.11.12 linux-
2.6.11.12-xenU > archive/linux-2.6.11.12-
xenU.patch
```

```
cd linux-2.6.11.12
patch -p1 < ../archive/linux-2.6.11.12-
xenU.patch
cd ../
rm -fr install
mkdir install
cd linux-2.6.11.12-xenU
cp arch/xen/configs/xenU_defconfig .config
ARCH=xen; INSTALL_PATH=../install;
INSTALL_MOD_PATH=../install
export ARCH INSTALL_PATH INSTALL_MOD_PATH
make oldconfig
make menuconfig
```

In the kernel comfiguration menu that shows up we have to enable *quota* and *iptables* as **modules** (it is **important** that they are **modules**. I could not get

iptables to work in a virtual machine when I compiled it into the kernel statically!). This is where you enable these modules:

```
File systems --> [*] Quota support
<M> Old quota format support
<M> Quota format v2 support

Device Drivers ---> Networking support --->
Networking options ---> [*] Network packet
filtering (replaces ipchains) ---> IP:
Netfilter Configuration ---> <M> IP tables
support (required for filtering/masq/NAT)
```

[*] means: build into the kernel statically.
<M> means: build as a kernel module.

Now we install the kernel to *~/xen/install*:

```
make modules
make modules_install
make install
```

Finally, we copy the kernel to the "real" locations:

```
cd ../install/
cp boot/* /boot
cp -r lib/modules/2.6.11.12-xenU/
/lib/modules
cp -r usr/include/xen /usr/include
```

(This is my domU kernel configuration.)

<< Prev **Next >>**

view as pdf | print

Sub pages

The Perfect Xen Setup For Debian And Ubuntu - Page 3 - Page 5
The Perfect Xen Setup For Debian And Ubuntu - Page 3 - Page 6

6 Comment(s)

Add comment

Name * Email *

p

Submit comment

Comments

From: Anonymous Reply

If you receive:
/root/xen-2.0/xen/include/asm/processor.h:175: error: array type has incomplete element type
You may need to specify the version of GCC you wish to use for the compile.
I found people recommending to use:
make CC=gcc-3.2 world
However, the best I could get was gcc-3.3 (ie. /usr/bin/gcc-3.3) so I did:
make CC=gcc-3.3 world

From: Anonymous Reply

My previous comment regarding make world also applies to make install

From: Anonymous Reply

When I compiled the kernel, I also needed to apt-get install patch to get the patch program.

From: Anonymous Reply

Steps that begin with "cd .." cause confusion. I suggest either changing to a specific folder "cd ~/xen" or at least commenting where you are changing to "#Change to the ~/xen folder" so the user can know they are in the correct path.
This is very useful. Thanks for taking the time to document the perfect xen setup!
Tim

> **From:** admin Reply
>
> As stated at the beginning of this tutorial, this is a "copy & paste" howto, i.e. you should simply follow the steps presented here without doing anything else in between (e.g. change directories or something like that) - then it will work without problems.

From: Anonymous Reply

Falko has done a brilliant job with this Howto. Apart from very minor things such as not mounting devpts after chrooting, it worked without a hitch. I've pasted a slightly modified version of section 3 below to show how I installed Xen 3.0.0 using Falko's Howto.
cheers,
Hizar
3.2 Installing From The Sources
Run the following commands:
apt-get remove exim4 exim4-base lpr nfs-common portmap pidentd pcmcia-cs pppoe pppoeconf ppp pppconfig
apt-get install iproute bridge-utils python-twisted gcc-3.3 binutils make libcurl3-dev zlib1g-dev python-dev transfig bzip2 screen ssh debootstrap libcurl-dev libncurses5-dev **(1 line!)**
cd ~
mkdir xen
cd xen
mkdir archive
cd archive
3.2.1 Install Xen
Now execute these commands:
wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.12.tar.bz2
wget http://www.cl.cam.ac.uk/Research/SRG/netos /xen/downloads/xen-3.0.0-src.tgz

cd ..
tar -xvzf archive/xen-unstable.7-src.tgz
tar -xvjf archive/linux-2.6.12.tar.bz2

cd xen-unstable
make world
make install
sh ./install.sh
Now Xen is installed. In order to start the Xen services at boot time, do the following:
update-rc.d xend defaults 20 21
update-rc.d xendomains defaults 21 20

3.2.2 Compile A New dom0 Kernel

Next we compile a new dom0 kernel with Xen-, iptables-, quota-, and dummy support. The kernel will be installed to ~/xen/install so that we can save it for other machines. From ~/xen/install we will install it to the real locations. Xen works with kernel 2.6.11, so I take the latest 2.6.12 kernel (2.6.12) instead of newer kernels, e.g. 2.6.14.
cd ..
mv /lib/tls /lib/tls.disabled
cp -al linux-2.6.12 linux-2.6.12-xen0
cd xen-unstable/linux-2.6-xen-sparse/
./mkbuildtree ../../linux-2.6.12-xen0/
cd ../..
echo "-xen0" > linux-2.6.12-xen0/localversion-xen
diff -Naur linux-2.6.12 linux-2.6.12-xen0 > archive/linux-2.6.12-xen0.patch
cd linux-2.6.12
patch -p1 < ../archive/linux-2.6.12-xen0.patch

3.2.2.1 Apply additional patches for Xen 3.0 (unstable) for dom0 Kernel

Patch –p1 < ~/xen/xen-unstable/patches/linux-2.6.12 /smp-alts.patch
Patch –p1 < ~/xen/xen-unstable/patches/linux-2.6.12 /rcu-nohz.patch
Patch –p1 < ~/xen/xen-unstable/patches/linux-2.6.12 /pmd-shared.patch
Patch –p1 < ~/xen/xen-unstable/patches/linux-2.6.12 /net-csum.patch
Patch –p1 < ~/xen/xen-unstable/patches/linux-2.6.12 /cpu-hotplug-init.patch
cd ../
mkdir install
cd linux-2.6.12-xen0
cp arch/xen/configs/xen0_defconfig .config
ARCH=xen; INSTALL_PATH=../install;
INSTALL_MOD_PATH=../install
export ARCH INSTALL_PATH INSTALL_MOD_PATH
make oldconfig
make menuconfig
In the kernel configuration menu that shows up we have to enable quota, iptables and the dummy network

driver as **modules**. This is where you enable these modules:
File systems --> [*] Quota support
<M> Old quota format support
<M> Quota format v2 support
Device Drivers ---> Networking support ---> <M> Dummy net driver support
Device Drivers ---> Networking support ---> Networking options ---> [*] Network packet filtering (replaces ipchains) ---> IP: Netfilter Configuration ---> <M> IP tables support (required for filtering/masq/NAT)
[*] means: build into the kernel statically.
<M> means: build as a kernel module.
Now we install the kernel to ~/xen/install:
make modules
make modules_install
make install
Finally, we copy the kernel to the "real" locations:
cd ../install/
cp boot/ /boot*
cp -r lib/modules/2.6.12-xen0/ /lib/modules
cp -r usr/include/xen /usr/include

3.2.3 Configure The Bootloader And Reboot

I configured my kernel with SMP support, hence the name I have given my kernel. Make sure you change the other settings to suit as per falko's howto.
title Xen 3.0.0 Debian SID dom0 2.6.12-6 SMP
root (hd0,0)
kernel /xen.gz dom0_mem=1024M vga=791
module /vmlinuz-2.6.12.6-xen0 root=/dev/hde2 ro console=tty0

Tutorials The Perfect Xen Setup For Debia...