

<http://www-128.ibm.com/developerworks/linux/library/l-xen/> - A moment of Xen: Virtualize Linux to test your apps. Build a sandbox to keep your applications separated, safe, and happy.

[http://www.howtoforge.com/xen\\_3.0\\_ubuntu\\_dapper\\_drake](http://www.howtoforge.com/xen_3.0_ubuntu_dapper_drake)

I have included the Xen configuration files from the presentation at the end of this How To. Please note that is a revised document that contains significant changes from the one at <http://www.howtoforge.com>.

## How To Set Up Xen 3.0 From Binaries In Ubuntu 6.06 LTS (Dapper Drake)

Version 1.0

Author: Ásgeir Bjarni Ingvarsson <istari@hlekkir.com>

Last Edited: June 13. 2006

Modified : Clint Tinsley <c\_tinsley@msn.com>

Last Edited: August 1. 2006

### 1 Install a clean base server system.

Setup Ubuntu like you normally would. I have created 3 partitions on hda, but you can do it any way you like. Just keep your differences in mind when you edit the config files.

- ? hda1 is swap
- ? hda2 is 2 GB and mounted as /
- ? hda3 is the rest of the disk and mounted as /xen-images

*To setup the above partitioning requires manually partitioning the system. I accepted the default parting in installing from the Dapper server installation CD which creates hda1 as root and hda2 as swap.*

### 2 Update the system and install Xen

#### 2.1 Install Xen and configure it

Open a root shell so you don't have to type sudo for every command.

```
sudo -s
```

Update the system and install necessary packages.

```
apt-get update
apt-get upgrade
apt-get install iproute python python-twisted bridge-utils dhcp3-server debootstrap
```

Download and extract the Xen 3.0 tarball from [XenSource](http://bits.xen-source.com/Xen/latest/xen-3.0.2-install-x86_32.tgz).  
wget [http://bits.xen-source.com/Xen/latest/xen-3.0.2-install-x86\\_32.tgz](http://bits.xen-source.com/Xen/latest/xen-3.0.2-install-x86_32.tgz)

```
tar xvf xen-3.0.1-install-x86_32.tgz
cd xen-3.0.1-install
./install.sh
```

Check for error messages - it should say OK to all.

Next we create modules.dep and map files for the new kernel. (see /lib/modules for the correct kernel version)

```
/sbin/depmod -a 2.6.16-xen
```

Edit /etc/mkinitramfs/modules and append the following line:

```
loop max_loop=64
```

If you run out of loop devices later just increase max\_loop and rebuild the initrd.

Create an initrd image. Use the same version number as before.

```
cd /boot
mkinitramfs -o initrd.img-2.6.16-xen 2.6.16-xen
```

Edit /boot/grub/menu.lst placing the following lines **before** the Automagic section

```
title Xen 3.0 / XenLinux 2.6
kernel /boot/xen-3.gz dom0_mem=128M
module /boot/vmlinuz-2.6-xen root=/dev/hda1 ro
module /boot/initrd.img-2.6.16-xen
```

Make Xen start up and autostart selected guests when the system starts up. xend must start before, and must be stopped after xendomains.

```
update-rc.d xend start 30 2 3 4 5 . stop 31 0 1 6 .
update-rc.d xendomains start 31 2 3 4 5 . stop 30 0 1 6 .
```

Disable Thread-Local Storage (remember to check this after every update) - critical

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

## 2.2a System configuration for DHCP and NAT

Modify /etc/xen/xend-config.sxp to support DHCP and NAT (default configuration is for bridging).

Locate the line below and insert a # sign at the front of the line (rem it!)

```
(network-script network-bridge)
```

Then add a new line directly after it so that it all looks like this:

```
 #(network-script network-bridge)
```

```
(network-script 'network-nat dhcp=yes')
```

About 14 lines down, find the line below and insert a # sign at the front of the line.

```
(vif-script vif-bridge)
```

so that it all looks like this:

```
 #(vif-script vif-bridge)
```

About 9 lines further down, below the NAT section add the line shown after the remarked line for #(network-script network-nat)

```
(vif-script 'vif-nat dhcp=yes')
```

Save changes.

## **2.2b The following workarounds may not be required later, but I had to do them.**

Rename xen-backend.rules

```
mv /etc/udev/rules.d/xen-backend.rules /etc/udev/rules.d/92-xen-backend.rules
```

To make sure that /var/run/xenstored and /var/run/xend exist. Edit /etc/init.d/xend and insert the following lines after the check for /proc/xen/capabilities

```
if [ ! -d /var/run/xend ] ; then
    mkdir -p /var/run/xend
fi

if [ ! -d /var/run/xenstored ] ; then
    mkdir -p /var/run/xenstored
fi
```

Edit /etc/init.d/xenddomains and change the LOCKFILE line to read

```
LOCKFILE=/var/lock/xenddomains
```

## **2.3 All done for this part so reboot.**

When the system is back up try the following command to verify that everything is ok.

```
xm list
```

You should see something similar to this:

Name	ID	Mem(MiB)	VCPUs	State	Time(s)
Domain-0	0	128	1	r-----	42.3

### 3 Configuring the guest domains

#### 3.1 Create disk images and bootstrap them

Create a mountpoint for the images.

```
mkdir /xen-images/mnt
```

Create a 2 GB image file and 500 MB swap file, for larger images increase count.

```
dd if=/dev/zero of=/xen-images/guest_base.img bs=1024k count=2000
dd if=/dev/zero of=/xen-images/guest_base-swap.img bs=1024k count=500
```

Change permissions for the image files. No one should have access to your Domain-0 computer since that would compromise security for all of the guest domains, but this is a good idea anyway and doesn't affect Xen.

```
chmod 640 /xen-images/guest_base*
```

Format `guest_base.img` as `ext3` and then format `guest_base-swap.img` as `swap`. When it says `"/xen-images/guest_base.img is not a block special device."` answer yes to proceed anyway.

```
mkfs.ext3 /xen-images/guest_base.img
mkswap /xen-images/guest_base-swap.img
```

Mount the guest image and bootstrap it.

```
mount /dev/cdrom /media/cdrom
mount -o loop /xen-images/guest_base.img /xen-images/mnt
debootstrap --arch i386 dapper /xen-images/mnt file:/media/cdrom
```

It should be noted that `debootstrap` command line will not work for all install cds. With Dapper, it does work with the server CD but not the regular Dapper installation CD. You can also use web installs such as <http://us.archive.ubuntu.com/ubuntu> but the some cautions exist as you may end up “bad” image.

Copy your `/etc/apt/sources.list` to the new image

```
cp /etc/apt/sources.list /xen-images/mnt/etc/apt/
```

Copy the kernel modules.

`cp -dpR /lib/modules/2.6.16-xen /xen-images/mnt/lib/modules/`  
If you don't do this, the system will boot with errors but still work.

Disable Thread-Local Storage. – If you don't do this, you will have a very poorly performing system.

`mv /xen-images/mnt/lib/tls /xen-images/mnt/lib/tls.disabled`

Configure networking for the guest `/xen-images/mnt/etc/network/interfaces`

Default recommendation:

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

# The primary network interface
# Uncomment this and configure after the system has booted for the
first time
#auto eth0
#iface eth0 inet static
#       address 192.168.0.101
#       netmask 255.255.255.0
#       broadcast 192.168.0.255
#       gateway 192.168.0.1
#       dns-nameservers 192.168.0.1
```

Clint's modified for DHCP assigned addresses:

(Remember, that this a Dapper image. Network configuration files vary from distribution to distribution.)

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5)
```

```
auto eth0 lo
iface eth0 inet static
    address 10.0.2.1
    netmask 255.255.0.0
    broadcast 10.0.2.255
    gateway 10.0.2.128
    dns-nameservers 172.30.0.65
iface lo inet loopback
```

Create `/xen-images/mnt/etc/hosts` and make it look like this.

```
127.0.0.1        localhost localhost.localdomain
127.0.0.1        guest

# The following lines are desirable for IPv6 capable hosts
::1             ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
```

```
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Create `/xen-images/mnt/etc/hostname`

```
echo guest > /xen-images/mnt/etc/hostname
```

Edit `/xen-images/mnt/etc/fstab` so it looks like this.

```
proc          /proc        proc         defaults    0 0
/dev/hda1     /            ext3        defaults,errors=remount -ro  0 1
/dev/hda2     none        swap        sw          0 0
```

Unmount the image.

```
umount /xen-images/mnt
```

If you get this warning "umount: /xen-images/mnt: device is busy" don't worry, it's not important.

### 3.2 Boot up the base image to finalize the configuration

Create a config file for the guest `/etc/xen/baseimage` (dhcp)

```
# -*- mode: python; -*-
kernel = "/boot/vmlinuz-2.6.16-xen"
ramdisk = "/boot/initrd.img-2.6.16-xen"
memory = 256
name = "baseimage"
dhcp = "dhcp"
vif = ['ip=10.0.2.1']
disk = ['file:/xen-images/guest_base.img,hda1,w', 'file:/xen-images/guest_base-swap.img,hda2,w']
# ip = "192.168.0.101"
# netmask = "255.255.255.0"
# gateway = "192.168.0.1"
hostname = "baseimage"
root = "/dev/hda1 ro"
extra = "4"
```

Start the guest.

```
xm create baseimage -c
```

Login as root and set the root password. The rest of this howto is suite to your preferences. You should have a working image and system at this point. The following is also distribution specific, your mileage may vary.

Enable shadow passwords.

shadowconfig on

Create a backup of /etc/network/interfaces

```
cp /etc/network/interfaces /etc/network/interfaces.bak
```

Edit /etc/network/interfaces and put in correct values so you can access the Internet

Start networking.

```
ifup -a
```

Install packages (add ubuntu-desktop and more if you want but beware of running out of disk space during the install/upgrades and potentially trashing your image).

```
apt-get update
```

```
apt-get install ubuntu-base configure-debian openssh-server
```

Disable Thread-Local Storage if it has been updated.

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

Configure the system any way you like, I recommend that you try every menu item. You should do utils->console-data before utils->console-common.

```
configure-debian
```

Make /etc/sudoers writable.

```
chmod 660 /etc/sudoers
```

Add the new user you created to /etc/sudoers

Fix permissions for /etc/sudoers

```
chmod 440 /etc/sudoers
```

Stop networking.

```
ifdown -a
```

Replace /etc/network/interfaces with the backup copy.

```
mv /etc/network/interfaces.bak /etc/network/interfaces
```

You are now done setting up the base image so shut it down.

## 4 Creating your first guest domain

Make a copy of the base guest images.

```
cp /xen-images/guest_base.img /xen-images/guestdom1.img
cp /xen-images/guest_base-swap.img /xen-images/guestdom1-swap.img
```

Rename /etc/xen/baseimage

```
mv /etc/xen/baseimage /etc/xen/guestdom1
```

Edit /etc/xen/guestdom1 and change name, disk and hostname to the following values.

```
name = "guestdom1"
disk = ['file:/xen-images/guestdom1.img,hda1,w', 'file:/xen-images/guestdom1-swap.img,hda2,w']
hostname = "guestdom1"
```

You will want to change the following files for each new guest you create.

- ? /etc/hostname
- ? /etc/hosts
- ? /etc/network/interfaces

To make the guest start up automatically and make Domain 0 wait for it before shutting down.

```
ln -s /etc/xen/guestdom1 /etc/xen/auto/
```

Or if you want to start and stop the guest manually.

```
xm create guestdom1
```

For further information check the [Xen documentation](http://www.cl.cam.ac.uk/Research/SRG/netos/xen/documentation.html) at <http://www.cl.cam.ac.uk/Research/SRG/netos/xen/documentation.html>

<http://jailtime.org> has a number of prebuilt Operating System Images that can be ran on Xen natively.

References:

[http://www.howtoforge.com/perfect\\_xen\\_setup\\_debian\\_ubuntu](http://www.howtoforge.com/perfect_xen_setup_debian_ubuntu)

<https://wiki.ubuntu.com/XenOnUbuntuBinaryInstall>

<http://www.ubuntuforums.org/showthread.php?t=178027>

<http://en.opensuse.org/XEN> - SuSE

<http://fedoraproject.org/wiki/FedoraXenQuickstartFC5> - Fedora Core 5

<http://www.linux-noob.com/forums/index.php?showtopic=2074> - CentOS

Other points of interest:

<https://wiki.ubuntu.com/XenVirtualMachine/XenOnUbuntuDapper>

Clint's configuration files for Xen:

- 1) grub menu.lst – Dom0
- 2) Xen Configuration File: xend-config.sxp
- 3) Webserver Configuration file – guest DomU
- 4) Directory listing of /xenimages/ guest image files.
- 5) dhcpd.conf file for dom0
- 6) Webserver /etc/network/interfaces configure file

Grub menu.lst

Ubuntu 6.0.6 grub menu modified for Xen boot.

```
# menu.lst - See: grub(8), info grub, update-grub(8)
#             grub-install(8), grub-floppy(8),
#             grub-md5-crypt, /usr/share/doc/grub
#             and /usr/share/doc/grub-doc/.

## default num
# Set the default entry to the entry number NUM. Numbering starts from
# 0, and
# the entry number 0 is the default if the command is not used.
#
# You can specify 'saved' instead of a number. In this case, the
# default entry
# is the entry saved with the command 'savedefault'.
# WARNING: If you are using dmraid do not change this entry to 'saved'
# or your
# array will desync and will not let you boot your system.
default                0

## timeout sec
# Set a timeout, in SEC seconds, before automatically booting the
# default entry
# (normally the first entry defined).
timeout                3

## hiddenmenu
# Hides the menu by default (press ESC to see the menu)
hiddenmenu

# Pretty colours
#color cyan/blue white/blue

## password [--md5] passwd
# If used in the first section of a menu file, disable all interactive
# editing
# control (menu entry editor and command-line) and entries protected
# by the
# command 'lock'
# e.g. password topsecret
```

```

# password --md5 $1$gLhU0/$aW78kHK1QfV3P2b2znUoe/
# password topsecret

#
# examples
#
# title          Windows 95/98/NT/2000
# root           (hd0,0)
# makeactive
# chainloader    +1
#
# title          Linux
# root           (hd0,1)
# kernel         /vmlinuz root=/dev/hda2 ro
#

#
# Put static boot stanzas before and/or after AUTOMAGIC KERNEL LIST

title Xen 3.0.2 / XenLinux 2.6.16
kernel /boot/xen-3.gz dom0_mem=128M
module /boot/vmlinuz-2.6-xen root=/dev/hda1 ro
module /boot/initrd.img-2.6.16-xen

# Note: I have limited dom0, the host "hypervisor" to 128M of Ram.

### BEGIN AUTOMAGIC KERNELS LIST
## lines between the AUTOMAGIC KERNELS LIST markers will be modified
## by the debian update-grub script except for the default options
below

## DO NOT UNCOMMENT THEM, Just edit them to your needs

## ## Start Default Options ##
## default kernel options
## default kernel options for automagic boot options
## If you want special options for specific kernels use kopt_x_y_z
## where x.y.z is kernel version. Minor versions can be omitted.
## e.g. kopt=root=/dev/hda1 ro
##      kopt_2_6_8=root=/dev/hdc1 ro
##      kopt_2_6_8_2_686=root=/dev/hdc2 ro
# kopt=root=/dev/hda1 ro

## default grub root device
## e.g. groot=(hd0,0)
# groot=(hd0,0)

## should update-grub create alternative automagic boot options
## e.g. alternative=true
##      alternative=false
# alternative=true

## should update-grub lock alternative automagic boot options
## e.g. lockalternative=true
##      lockalternative=false
# lockalternative=false

```

```

## additional options to use with the default boot option, but not with
the
## alternatives
## e.g. defoptions=vga=791 resume=/dev/hda5
# defoptions=quiet splash

## altoption boot targets option
## multiple altoptions lines are allowed
## e.g. altoptions=(extra menu suffix) extra boot options
##      altoptions=(recovery mode) single
# altoptions=(recovery mode) single

## controls how many kernels should be put into the menu.lst
## only counts the first occurrence of a kernel, not the
## alternative kernel options
## e.g. howmany=all
##      howmany=7
# howmany=all

## should update-grub create memtest86 boot option
## e.g. memtest86=true
##      memtest86=false
# memtest86=true

## should update-grub adjust the value of the default booted system
## can be true or false
# updatedefaultentry=false

## ## End Default Options ##

title      Ubuntu, kernel 2.6.15-23-server
root       (hd0,0)
kernel     /boot/vmlinuz-2.6.15-23-server root=/dev/hda1 ro
quiet splash
initrd     /boot/initrd.img-2.6.15-23-server
savedefault
boot

title      Ubuntu, kernel 2.6.15-23-server (recovery mode)
root       (hd0,0)
kernel     /boot/vmlinuz-2.6.15-23-server root=/dev/hda1 ro
single
initrd     /boot/initrd.img-2.6.15-23-server
boot

title      Ubuntu, memtest86+
root       (hd0,0)
kernel     /boot/memtest86+.bin
boot

### END DEBIAN AUTOMAGIC KERNELS LIST

Xen Configuration File: xend-config.sxp modified from default to vif DHCP NAT.

# -*- sh -*-

```

```
#
# Xend configuration file.
#

# This example configuration is appropriate for an installation that
# utilizes a bridged network configuration. Access to xend via http
# is disabled.

# Commented out entries show the default for that entry, unless
otherwise
# specified.

#(logfile /var/log/xend.log)
#(loglevel DEBUG)

#(xend-http-server no)
#(xend-unix-server no)
#(xend-tcp-xmlrpc-server no)
#(xend-unix-xmlrpc-server yes)
#(xend-relocation-server no)
(xend-relocation-server yes)

#(xend-unix-path /var/lib/xend/xend-socket)

# Port xend should use for the HTTP interface, if xend-http-server is
set.
#(xend-port 8000)

# Port xend should use for the relocation interface, if xend-
relocation-server
# is set.
#(xend-relocation-port 8002)

# Address xend should listen on for HTTP connections, if xend-http-
server is
# set.
# Specifying 'localhost' prevents remote connections.
# Specifying the empty string '' (the default) allows all connections.
#(xend-address '')
#(xend-address localhost)

# Address xend should listen on for relocation-socket connections, if
# xend-relocation-server is set.
# Meaning and default as for xend-address above.
#(xend-relocation-address '')

# The hosts allowed to talk to the relocation port. If this is empty
(the
# default), then all connections are allowed (assuming that the
connection
# arrives on a port and interface on which we are listening; see
# xend-relocation-port and xend-relocation-address above). Otherwise,
this
# should be a space-separated sequence of regular expressions. Any
host with
# a fully-qualified domain name or an IP address that matches one of
these
```

```

# regular expressions will be accepted.
#
# For example:
# (xend-relocation-hosts-allow '^localhost$ ^.*\.example\.org$')
#
#(xend-relocation-hosts-allow '')
(xend-relocation-hosts-allow '^localhost$')

# The limit (in kilobytes) on the size of the console buffer
#(console-limit 1024)

##
# To bridge network traffic, like this:
#
# dom0: fake eth0 -> vif0.0 -+
#                               |
#                               bridge -> real eth0 -> the network
#                               |
# domU: fake eth0 -> vifN.0 -+
#
# use
#
# (network-script network-bridge)
#
# Your default ethernet device is used as the outgoing interface, by
# default.
# To use a different one (e.g. eth1) use
#
# (network-script 'network-bridge netdev=eth1')
#
# The bridge is named xenbr0, by default. To rename the bridge, use
#
# (network-script 'network-bridge bridge=<name>')
#
# It is possible to use the network-bridge script in more complicated
# scenarios, such as having two outgoing interfaces, with two bridges,
# and
# two fake interfaces per guest domain. To do things like this, write
# yourself a wrapper script, and call network-bridge from it, as
# appropriate.
#
# (network-script network-bridge) default remarked out
# Network script network NAT DHCP added
(network-script 'network-nat dhcp=yes')

# The script used to control virtual interfaces. This can be
# overridden on a
# per-vif basis when creating a domain or a configuring a new vif. The
# vif-bridge script is designed for use with the network-bridge script,
# or
# similar configurations.
#
# If you have overridden the bridge name using
# (network-script 'network-bridge bridge=<name>') then you may wish to
# do the
# same here. The bridge name can also be set when creating a domain or

```

```

# configuring a new vif, but a value specified here would act as a
default.
#
# If you are using only one bridge, the vif-bridge script will discover
that,
# so there is no need to specify it explicitly.
#
# (vif-script vif-bridge) default remarked out

## Use the following if network traffic is routed, as an alternative to
the
# settings for bridged networking given above.
# (network-script network-route)
# (vif-script      vif-route)

## Use the following if network traffic is routed with NAT, as an
alternative
# to the settings for bridged networking given above.
# (network-script network-nat)
# vif script vif NAT DHCP added
(vif-script 'vif-nat dhcp=yes')

# Dom0 will balloon out when needed to free memory for domU.
# dom0-min-mem is the lowest memory level (in MB) dom0 will get down
to.
# If dom0-min-mem=0, dom0 will never balloon out.
(dom0-min-mem 196)

# In SMP system, dom0 will use dom0-cpus # of CPUS
# If dom0-cpus = 0, dom0 will take all cpus available
(dom0-cpus 0)

# Whether to enable core-dumps when domains crash.
#(enable-dump no)

"Webserver" Configuration file to load webserver virtual domU guest.

# Webserver Config
kernel = "/boot/vmlinuz-2.6-xen"
memory = 256
name = "webserver"
dhcp = "dhcp"
# vif = ['bridge=xenbr0'] default 'bridge=xenbr0' remarked out
vif = ['ip=10.0.2.1']
disk =
['file:/xenimages/webserver,hda1,w', 'file:/xenimages/websrvr_swap,hda2,
w']
# Defined IP address for bridge mode are remarked out.
# ip = "192.168.0.101"
# netmask = "255.255.255.0"
# gateway = "192.168.0.1"
hostname = "webserver"
root = "/dev/hda1 ro"
xtra = "4"

/xenimages/ directory listing:

```

```
-rw-r----- 1 root root 2097152000 Aug  1 22:11 webservice
-rw-r--r--  1 root root  524288000 Jul 30 13:58 websrvr_swap
```

dhcpd.conf file for dom0:

```
option domain-name-servers 205.171.3.65, 172.30.0.65 ;
option domain-name "domain.actdslttmp" ;
subnet 10.0.0.0 netmask 255.255.0.0 {}
```

Webserver /etc/network/interfaces configure file:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
auto eth0 lo
iface eth0 inet static
    address      10.0.2.1
    netmask      255.255.0.0
    broadcast    10.0.2.255
    gateway      10.0.2.128
    dns-nameservers 192.168.0.17
iface lo inet loopback
```

Note that the above static addresses maybe determined by examining the /etc/dhcp3/dhcpd.conf file after starting the DomU session for the "webservice" The gateway address are also available from the ifconfig listing for the Dom0 "host" session after starting the DomU session. The command for starting the webserver DomU session is **xm create -c webservice** .