

Xen

- [Introduction](#)
 - [Introduction to Xen](#)
 - [Intended audience](#)
 - [Versions](#)
 - [Glossary](#)
- [Toolstacks \(a.k.a toolkits and toolboxes\)](#)
 - [libvirt and virsh](#)
 - [xapi and xe](#)
 - [xend and xm](#)
 - [xl](#)
- [Toolstack library](#)
 - [Libxenlight \(libxl\)](#)
- [Toolstack configuration](#)
 - [Debian 7 Wheezy](#)
- [xen-tools](#)
 - [References](#)
- [Storage](#)
- [Networking](#)
 - [Bridges](#)
 - [Traditional](#)
 - [Single MAC address on the LAN](#)
 - [Bridge references](#)
 - [Routing](#)
 - [NATting](#)
 - [References \(networking\)](#)
- [DomU configuration](#)
 - [Kernel parameters](#)
 - [Memory](#)
 - [vif](#)
- [Installation \(host/dom0\)](#)
- [Maintaining Debian in DomUs](#)
- [Installing Debian in DomUs](#)
 - [debootstrap](#)
 - [Debian installer](#)
 - [Netboot](#)
 - [.iso image](#)
 - [xen-create-image \(from xen-tools\)](#)
 - [Required Debian packages](#)
- [Controlling DomUs](#)
 - [Start a DomU](#)
 - [Stop a DomU](#)
- [Monitoring DomUs](#)
 - [Console](#)
- [Troubleshooting](#)
 - [PV, PV-HVM or pure HVM?](#)
 - [Troubleshooting references](#)
- [References](#)
 - [Files and directories](#)
- [Technical](#)
 - [Architecture](#)
- [Debian packages](#)
 - [libxen-4.1](#)
 - [libxenstore3.0](#)
 - [xen-hypervisor-4.1-amd64](#)
 - [xen-linux-system-3.2.0-4-amd64](#)
 - [xen-linux-system-amd64](#)
 - [xen-system-amd64](#)
 - [xen-tools](#)
 - [xen-utils-4.1](#)
 - [xen-utils-common](#)
 - [xenstore-utils](#)
- [References](#)
 - [Introductory](#)
 - [Entry points](#)
 - [Setup logs](#)

Introduction

This page is intended to explain Xen terminology and how the Xen user space components fit together.

The problem for a Xen newcomer is not a shortage of documentation. Quite the opposite; there is plenty of official Xen documentation and unofficial Xen blog posts. The problem is understanding the documentation. Most of the documentation assumes prior knowledge of Xen terminology and how the Xen user space components fit together. But there is little introductory documentation explaining those things. This document attempts to do so.

Introduction to Xen

http://wiki.xenproject.org/wiki/Xen_Beginners_Guide#What_is_this_Xen_Project_software_all_about.3F

Intended audience

People wanting a high level view of Xen user space tools, how they fit together and Xen terminology – especially on Debian.

Versions

Xen has changed significantly between releases and, less so, as packaged for various Linux distributions. This page is based on:

- OS: Debian 7 Wheezy 64 bit
- Xen 4.1
 - Toolstack: xl (other toolstacks are mentioned to provide context)
- Xen-tools 4.3.1

Glossary

- **Domain** a Xen guest a.k.a virtual machine and VM.
- **domain-id** The numeric id of a DomU, a.k.a domain-id. Dynamically assigned. Shown in `xl list` output.
- **Dom0** The Xen domain/VM used a) to control the hypervisor b) as the primary interface to the hardware.
- **DomU<D>** Unprivileged Xen domains/VMs. D is the DomU number.
- **Enlightened guest** An OS designed/modified for use in a Xen DomU.
- **Full virtualisation** A domain/VM providing full emulation of physical hardware. The OS can be the same as runs on real hardware.
- **Guest domain** Same as domu*.
- **Hardware Virtual Machine (HVM)** Same as full virtualisation.
- **Hypervisor** a software system that allows the execution of multiple virtual guest operating systems simultaneously on a single physical machine. Xen's hypervisor is Type 1 or "bare-metal", meaning it runs on the physical machine as opposed to within an operating system.
- **HVM** Hardware Virtual Machine.
- **Para-virtualisation** uses modified guest operating systems a.k.a. enlightened guests. They don't require virtual hardware devices, instead they make special calls to the hypervisor that allow them to access CPUs, storage and network resources. Better performance than full virtualisation.
- **PV** para-virtualisation.
- **Virtual Machine Monitor (VMM)** Same as hypervisor.
- **Xen Store** A database of information shared by the hypervisor, the kernels, the drivers and the xen daemon. The memory for Xen Store is provided by Dom0. In Dom0 (and in enlightened guests?) it can be accessed via `/proc/xen/xenbus/`

Toolstacks (a.k.a toolkits and toolboxes)

A toolstack is a set of user-space tools used for administering Xen.

Several toolstacks are available for the interactive or scripted administration of Xen resources.

Primary reference: http://wiki.xen.org/wiki/Choice_of_Toolstacks

Apparently, once a toolstack has been chosen, it is not easy to change. From Debian 7 Wheezy's `/etc/default/xen` (where the chosen toolstack is configured):

```
# Attention: You need to reboot after changing this!
```

libvirt and virsh

libvirt is a library for managing KVM, OpenVZ, VMware, VirtualBox, Xen and others. For Xen, it is the least feature-complete of all the toolstacks.

virsh is a user/script shell to interface with libvirt. References: <http://libvirt.org/virshcmdref.html>

xapi and xe

The most feature-complete of all the toolstacks.

xend and xm

Deprecated since Xen 4.1 but the default in Xen 4.1 as packaged for Debian. Will be removed from Xen 4.2.

xend comprises:

- xm command
- xend-config.sxp config file format

- xmdomain.cfg config file format

xl

Strategic since Xen 4.1.

Designed to be command line compatible with xend.

Unlike xm, xl does not do dom0 network configuration; standard OS tools must be used.

For help (⚠ not --help or -h): `xl help`

In case there is no local xl man page: <http://xenbits.xen.org/docs/unstable/man/xl.1.html> and (Xen 4.2) <http://manpages.ubuntu.com/manpages/raring/en/man1/xl.1.html>

Many xl commands require a domain-id. It is dynamically associated with the domain-name. The `xl list` command shows the domain-id,

xl's -v option can be repeated for greater verbosity (info not in the man page).

Toolstack library

Libxenlight (libxl)

The strategic Xen toolstack library, designed to hide xenstore, libxenctrl, and libxenguest from higher levels.

Toolstack configuration

Debian 7 Wheezy

The Xen toolstack on Debian 7 Wheezy is configurable. For the xl toolstack edit `/etc/default/xen` to contain:

On blav2, we had explicitly configured using xl:

```
TOOLSTACK=xl
```

Now, when root runs command `xl`:

1. The shell runs `/usr/sbin/xl` which is a symlink to `/usr/lib/xen-common/bin/xen-toolstack-wrapper`
2. `/usr/lib/xen-common/bin/xen-toolstack-wrapper`:
 - a. Sources `/usr/lib/xen-common/bin/xen-toolstack` to set `TOOLSTACK` to `/usr/lib/xen-4.1/bin/xl`
 - b. Sources `/usr/lib/xen-common/bin/xen-dir` to set `dir` to `/usr/lib/xen-4.1`
 - c. Sources `/usr/lib/xen-common/bin/xen-version` to set `VERSION` to 4.1
 - i. `/usr/lib/xen-common/bin/xen-version` gets Xen version info from `/sys/hypervisor/version/*`
 - d. Sets `PATH` to `/usr/lib/xen-4.1/bin`
 - e. Sources `/etc/default/xen` to set `TOOLSTACK` to `xl` (would fall back to hardcoded default otherwise)
 - f. Sets `TOOLSTACK` to `/usr/lib/xen-4.1/bin/xl`
 - g. execs `/usr/lib/xen-4.1/bin/xl`

When root runs command `xm` (does not work, as intended):

1. The shell runs `/usr/sbin/xm` which is a symlink to `/usr/lib/xen-common/bin/xen-toolstack-wrapper`
2. `/usr/lib/xen-common/bin/xen-toolstack-wrapper`:
 - a. Sources `/usr/lib/xen-common/bin/xen-toolstack` to set `TOOLSTACK` to `/usr/lib/xen-4.1/bin/xl`
 - b. Sources `/usr/lib/xen-common/bin/xen-dir` to set `dir` to `/usr/lib/xen-4.1`
 - c. Sources `/usr/lib/xen-common/bin/xen-version` to set `VERSION` to 4.1
 - i. `/usr/lib/xen-common/bin/xen-version` gets Xen version info from `/sys/hypervisor/version/*`
 - d. Sets `PATH` to `/usr/lib/xen-4.1/bin`
 - e. Sources `/etc/default/xen` to set `TOOLSTACK` to `xl` (would fall back to hardcoded default otherwise)
 - f. Sets `TOOLSTACK` to `/usr/lib/xen-4.1/bin/xl`
 - g. Generates message **"ERROR: A different toolstack (xl) have been selected!"**

xen-tools

Works at a higher level than Xen toolstacks,

Includes the very convenient `xen-create-image` which largely automates creating domUs.

Logs in the `/var/log/xen-tools/` directory.

- Xen-Tools.org documentation: <http://xen-tools.org/software/xen-tools/>
- Xen-Tools.org FAQ: <http://xen-tools.org/software/xen-tools/faq.html>
- Xen Project Community Blog: Xen-Tools page: <http://blog.xen.org/index.php/2012/08/31/xen-tools-a-straightforward-vm-provisioninginstallation-tool/>

The recommended way to provide storage for a DomU is by an LVM on Dom0 which the DomU uses as a virtual HDD.

Reference: http://wiki.xenproject.org/wiki/Storage_options

The Dom0's standard Linux networking features can be used with Xen: bridges and [Netfilter](#) ([iptables](#)/[ipchains](#)):

- These facilities are not exclusive; they can be used in combination.

- **peth*** The physical ethernet interfaces in a Xen bridge, connected to the LAN
- **tap<D,I>** Virtual interfaces in a Xen bridge for an HVM DomU. Normally D is the DomU's domain-id and I is the DomU's zero-based interface number.
- **vif<D,I>** Virtual interfaces in a Xen bridge for an PV DomU. Normally D is the DomU domain-id and I is the DomU's zero-based interface number.
- **veth*** TBC
- **xenbr*** A Xen bridge

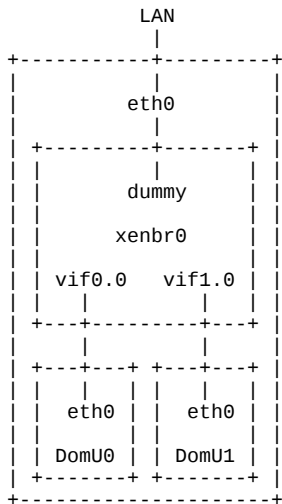
In the traditional simple bridged setup, one of Dom0's physical interfaces, say eth0, gets the bridge's address, broadcast, netmask and gateway settings. The bridge configuration will include, for example `bridge_ports eth0`. Each DomU's virtual interface is connected to the bridge/

Here's a more sophisticated example showing Dom0 with two physical interfaces, each with its own bridge and each DomU with two virtual interfaces, one connected to each bridge. It is useful for high-availability configurations. The diagram is from http://wiki.xenproject.org/wiki/Xen_Networking#ASCII_Art_Examples_of_Xen_Networking_Topologies:



The traditional bridged setup exposes the MAC addresses of the DomU's interfaces on the LAN. In some cases this will not work because the LAN router (s)/switch(es) will not accept a single physical interface having more than one MAC address.

The solution is to configure a dummy interface on the bridge for the DomU's physical interface, say eth0. This leaves eth0 with its own address(es), broadcast, netmask and gateway settings.



/etc/network/interfaces example for single MAC address on the LAN. The DomUs are configured to use 192.168.42.100 as default gateway:

```

...
# Configure netfilter
pre-up iptables-restore < /etc/iptables.conf
...
# dummy0
auto dummy0
iface dummy0 inet manual

# xenbr0
auto xenbr0
iface xenbr0 inet static
    bridge_ports dummy0
    address 192.168.42.100
    broadcast 192.168.42.255
    netmask 255.255.255.0
    bridge_stp off          # Disable Spanning Tree Protocol
    bridge_waitport 0       # No delay before a port becomes available
    bridge_fd 0             # No forwarding delay

```

Bridge references

- TLDP's Ethernet Bridge + netfilter Howto (ish, old): <http://www.tldp.org/HOWTO/Ethernet-Bridge-netfilter-HOWTO.html>
- Xen: Common problems with network bridges: <http://www.novell.com/support/kb/doc.php?id=7001989>
- Considering the "bridge_stp off" parameter. http://wiki.xen.org/wiki/Xen_Bridge_Loop#Why_STP_is_turned_off.3F

Routing

When routing is used the kernel must be configured with:

```

net.ipv4.ip_forward = 1
net.ipv4.conf.eth0.proxy_arp = 1

```

On Debian, to set them during boot, put them in /etc/sysctl.conf

NATting

To set up NATting (example is for eth0):

```

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

```

TODO: give command to save current netfilters rules for application during boot.

References (networking)

- Network Configuration Examples (Xen 4.1+): http://wiki.xenproject.org/wiki/Network_Configuration_Examples_%28Xen_4.1%2B%29
- Network Configuration Examples (Xen WIKI): http://wiki.xen.org/wiki/Host_Configuration/Networking
- Bridging Network Connections (Debian WIKI): <https://wiki.debian.org/BridgeNetworkConnections>

DomU configuration

Each DomU (the VM itself, not its OS etc.) is configured by a file in /etc/xen. Its name is the DomU's hostname with .cfg appended.

Its format, for use with the xl toolstack, is documented at <http://xenbits.xen.org/docs/unstable/man/xl.cfg.5.html>.

⚠️ xl shutdown and reboot does not effect changed configuration items; a create is required.

Kernel parameters

Use the extra configuration item.

Memory

Set by the memory configuration item. Numeric only (no M or G suffix allowed). Units are MB (or MiB?).

vif

```
vif (Default=[]; Value="type=TYPE, mac=MAC, bridge=BRIDGE, ip=IPADDR, script=SCRIPT," + \ "backend=DOM,
vifname=NAME, rate=RATE, model=MODEL, accel=ACCEL" )
```

- The vif is configured by calling the given configuration script.
- If **type** is not specified, default is netfront.
- If **mac** is not specified a random MAC address is used. If not specified then the network backend chooses it's own MAC address.
- If **bridge** is not specified the first bridge found is used.
- If **script** is not specified the default script is used. ⚠️ In practice, the full path to the script had to be given.
- If **backend** is not specified the default backend driver domain is used.
- If **vifname** is not specified the backend virtual interface will have name vifD.N where D is the domain id and N is the interface id. ⚠️ In practice a name had to be given.
- If **rate** is not specified the default rate is used.
- If **model** is not specified the default model is used.
- If **accel** is not specified an accelerator plugin module is not used. This option may be repeated to add more than one vif. Specifying vifs will increase the number of interfaces as needed.

Installation (host/dom0)

```
aptitude install xen-linux-system
```

Adjust grub so the default boot item is Xen (ref: <https://xen-orchestra.com/cant-find-hypervisor-information-in-sysfs/>)

```
dpkg-divert --divert /etc/grub.d/08_linux_xen --rename /etc/grub.d/20_linux_xen
update-grub
shutdown -r now
```

Maintaining Debian in DomUs

This is listed before "Creating Debian DomUs" because a newly created Debian should be updated during installation.

Installing an apt cacher (apt-cacher-ng) on Dom0 will save bandwidth and speed maintenance.

Installing Debian in DomUs

There are several ways to install Debian in a DomU. The xen-create-image method is very convenient. It uses the Debian Installer's debootstrap.

debootstrap

Installs into a chroot. Can use only one repository (which may be a local cache).

References:

- Debian WIKI: <https://wiki.debian.org/Debootstrap>

- Installing new Debian systems with debootstrap (2006): http://www.debian-administration.org/article/426/Installing_new_Debian_systems_with_debootstrap
- Installing Debian GNU/Linux from a Unix/Linux System: <http://www.debian.org/releases/stable/amd64/apds03.html.en>

Debian installer

"Debian installer" is the standard Debian installation system; it supports installation of standard Debian as a Xen PV.

References:

- http://wiki.xenproject.org/wiki/Debian_Guest_Installation_Using_Debian_Installer
- Debian GNU/Linux Installation Guide (Wheezy 64-bit version): <http://www.debian.org/releases/stable/amd64/>

Netboot

<http://www.debian.org/releases/stable/amd64/ch04s05.html.en>

.iso image

Not as easy as using xen-create-image

xen-create-image (from xen-tools)

xen-create-image is part of the xen-tools package.

When creating Debian systems it uses debbootstrap (described above).

Required Debian packages

ntp is required. Ref: <http://xen.1045712.n5.nabble.com/keeping-time-in-sync-tp5725115p5725116.html>

Controlling DomUs

Start a DomU

To create a DomU and boot its OS: `xl create <DomU config path>`

For example:

```
cd /etc/xen && xl create host.some_domain.org.cfg
```

Stop a DomU

To shutdown the OS and destroy the DomU: `xl shutdown <domain-id>`

For example:

```
xl list
xl shutdown 5
```

In case the DomU's OS has been stopped (or will not stop!): `xl destroy <domain-id>`

Monitoring DomUs

```
xl top
```

Console

To start a console when starting a DomU, use `xl create`'s `-c` option.

To attach to the console of a running DomU: `xl console <domain-id>`

For example:

```
xl console 2
```

Detach from the console by `Ctl+]` or, for `putty`, `Ctl+5`

Troubleshooting

PV, PV-HVM or pure HVM?

```
uname -a; lsmod | grep xen
```

If `uname -a` lists a kernel with the string "xen" in it, then you have a modified kernel and it's likely a PV guest and you will see output from the `lsmod` command to confirm it. If you have output from the `grep` on `lsmod` but no sign of a modified kernel then you are PV-HVM. Without any sign of either, it's a straight HVM.

Thanks to Adam C <http://serverfault.com/questions/511923/determine-which-guest-is-running-on-xen-hvm-or-pv-guest> for the above.

Troubleshooting references

- Redhat: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Virtualization/chap-Virtualization-Troubleshooting_Xen.html

References

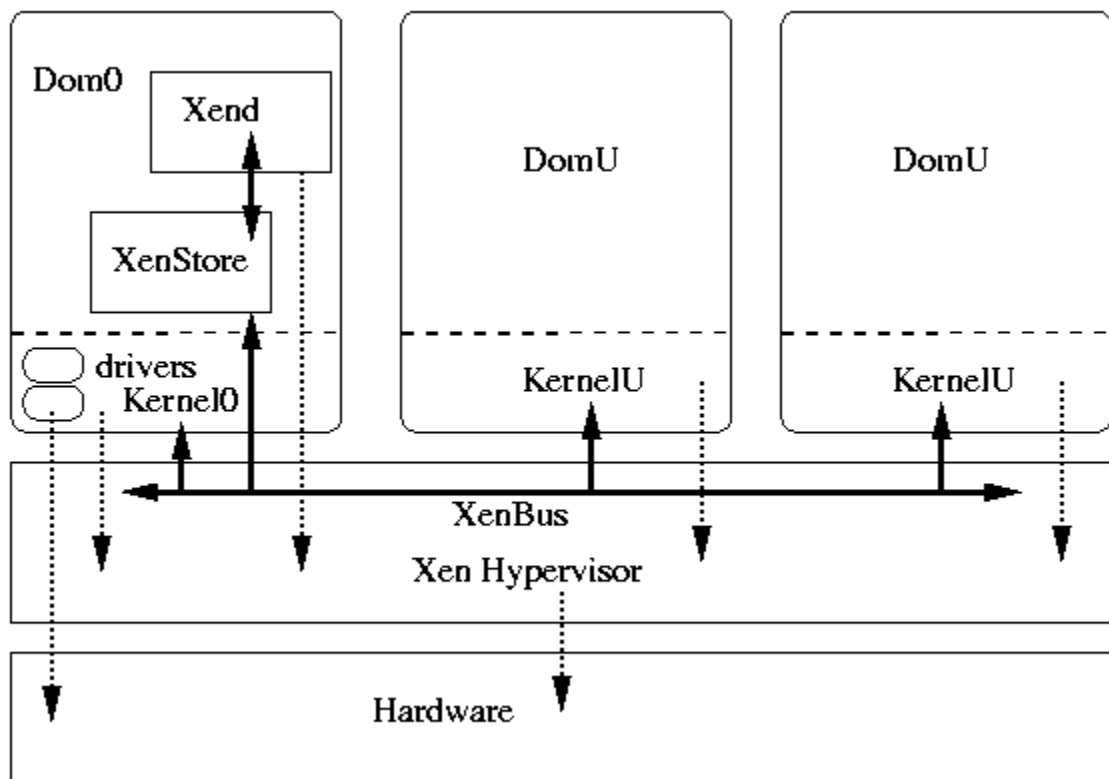
Files and directories

File or directory	Usage	Notes
/etc/default/xen	Toolstack selection	
/etc/default/xend		
/etc/default/xendomains	Domain startup and shutdown config	
/etc/xen/	Configuration	
/etc/xen/*.cfg	Individual DomU configs	
/etc/xen/scripts/	Scripts 😊 and scrippets	As installed, all bash scripts. More: Xen scripts
/etc/xen/xend-config.sxp	xend daemon (and xen-tools) configuration	
/etc/xen/xl	xl defaults config	
/proc/xen/	xen-kernel information	
/var/lib/libvirt/images	File-based images	
/var/lib/xen/dump/	Guest core dumps	
/var/log/xen	Logs	

Technical

Architecture

Very little information found on the 'net about Xen architecture. This diagram is from <http://libvirt.org/architecture.html>



Primary reference (no Xen 4.x equivalent found): <http://wiki.xen.org/wiki/XenIntro>

Debian packages

libxen-4.1

Includes /usr/lib/libxenctrl-4.1.so and /usr/lib/libxenguest-4.1.so

libxenstore3.0

Includes /usr/lib/libxenstore.so.3.0.0 and /usr/lib/libxenstore.so.3.0

xen-hypervisor-4.1-amd64

Includes /boot/xen-4.1-amd64.gz

xen-linux-system-3.2.0-4-amd64

Documentation only. TODO: presumably its dependencies are crucial; list them?

xen-linux-system-amd64

Documentation only. TODO: presumably its dependencies are crucial; list them?

xen-system-amd64

Documentation only. TODO: presumably its dependencies are crucial; list them?

xen-tools

Tools that work with the xm and xl toolstacks to administer Xen config files and images.

Provides (in /usr/bin):

- xt-customize-image
- xt-create-xen-config
- xen-list-images
- xen-delete-image
- xt-install-image
- xen-create-nfs
- xen-create-image
- xt-guess-suite-and-mirror
- xen-update-image

Plus, presumably in support:

- /etc/xen-tools/*
- /usr/lib/xen-tools/<OS name>.d/*

xen-utils-4.1

Tools to control the hypervisor from dom0.

Provides (in /usr/lib/xen-4.1/bin/):

- xentrace_setsize
- xenpm
- xen-python-path
- qemu-dm
- xsview
- xen-hvmcrash
- xentrace_setmask
- xenstored
- xentrace_format
- xenbaked
- readnotes
- xenconsole
- xc_restore
- gtraceview
- xen-bugtool
- xen-hptool
- xentop
- xend
- xc_save
- gtracestat
- xenctx
- xenwatchdogd
- xenpaging
- pygrub
- xencons
- xen-detect
- lsevtchn
- tap-ctl
- xl
- xentrace
- xm
- xen-hvmctx
- xenconsole
- xenstore-control
- xenmon
- xenlockprof
- xen-tmem-list-parse
- xenperf

Plus, presumably in support:

- /usr/lib/xen-4.1/lib/
- /usr/lib/xen-4.1/boot/hvmloader

xen-utils-common

Provides:

- /etc/default/xend
- /etc/default/xenddomains
- /etc/init.d/xen

- /etc/init.d/xendomains
- /etc/xen/scripts/*
- /etc/xen/xend-config.sxp
- /etc/xen/xend-pci-permissive.sxp
- /etc/xen/xend-pci-quirks.sxp
- /etc/xen/xl.conf
- /lib/udev/rules.d/xen-backend.rules
- /lib/udev/rules.d/xend.rules
- /usr/lib/xen-common/bin/*
- /usr/sbin/cpuperf-perfcnt
- /usr/sbin/cpuperf-xen
- /usr/sbin/xe
- /usr/sbin/xenperf
- /usr/sbin/xenpm
- /usr/sbin/xentop
- /usr/sbin/xentrace
- /usr/sbin/xentrace_format
- /usr/sbin/xentrace_setmask
- /usr/sbin/xentrace_setsize
- /usr/sbin/xl
- /usr/sbin/xm
- /usr/share/xen-utils-common/default.xen

xenstore-utils

Provides (all in /usr/sbin):

- xenstore-ls
- xenstore-write
- xenstore-exists
- xenstore-rm
- xenstore-chmod
- xenstore-watch
- xenstore-list
- xenstore-read

References

General references are listed below. Specific references are listed in the sections they apply to above.

Introductory

- Wikipedia: <http://en.wikipedia.org/wiki/Xen>
- Debian Wiki Xen page (in draft): <https://wiki.debian.org/Xen>
- Xen Project Beginners Guide: wiki.xenproject.org/wiki/Xen_Beginners_Guide
- XPSU13 VIDEO: Xen for Beginners (49 min): <http://www.xenproject.org/help/presentations-and-videos/video/latest/xpus13-beginners.html>

Entry points

These are entry points to references.

- Xen WIKI: http://wiki.xenproject.org/wiki/Main_Page
- Xen man pages: http://wiki.xenproject.org/wiki/Xen_Man_Pages

Setup logs

- Xen On Debian Wheezy With LVM (HowtoForge. Has bridge example with fixed IP): <http://www.howtoforge.com/xen-on-debian-wheezy-with-lvm>
- Using Xen With LVM-Based VMs Instead Of Image-Based VMs (HowtoForge. Etch): <http://www.howtoforge.com/using-xen-with-lvm-based-vms-instead-of-image-based-vms-debian-etch>
- Debian Wheezy -> Jessie and set up Xen (HowtoForge): <http://www.howtoforge.com/how-to-set-up-xen-4.3-on-debian-wheezy-7.0.2-and-then-upgrade-to-jessie>
- Setting up XEN on a Hetzner Dedicated Server (Squeeze, xen-create-image, xm): <https://gist.github.com/meskyanichi/3354956>
- Routed XEN VM based on LVM (step-by-step for Hetzner DS3000. Lenny): http://wiki.summercode.com/routed_xen_vm_based_lvm_step_by_step_guide_for_hetzner_ds3000
- Xen on 4 app servers (xl, no public IPs): http://wiki.xenproject.org/wiki/Xen_on_4_app_servers
- The Debian Administrator's Handbook, Virtualisation chapter (Nice. xen-tools, xm, bridged): <http://debian-handbook.info/browse/stable/sect.virtualization.html>
- How to install Xen in Debian (Armen's Wiki. Has scripts for both public and private IPs for virtual machines): <http://vardump.org/wiki.php/HowToInstallXenInDebianGNU/Linux>.