# Installation of Xen, XAPI (XenAPI), Tboot , RPcore and XAPI Proxy on Ubuntu 13.04 (alongwith Ubuntu 12.04 VM as domU)

XCP ( Xen Cloud Platform ) is the open source version similar to Citrix XenServer that uses the Xen Hypervisor. It is currently distributed as an ISO installer also called as XCP appliance. XCP uses XAPI or XenAPI to manage Xen hosts. XCP is based on CentOS 5.5 Project Kronos is an initiative to port the XAPI tool stack to Debian and Ubuntu. It is a management stack implemented in OCaml that configures and controls Xen hosts, attached storage, networking and virtual machine life cycle. It exposes a HTTP API and provides a command line interface (xe) for resource management.

Prerequisites :

• Ubuntu 13.04 server

• Small root file system partition and have a large space dedicated for a LVM volume.

• root access to the host

## Installing and configuring XAPI (XenAPI)

1. Install XCP-XAPI

*# apt-get install xcp-xapi*

– choose bridge when prompted for network backend

2. Setup GRUB to boot the Xen Hypervisor

*# sed -i ‘s/GRUB\_DEFAULT=.\*+/GRUB\_DEFAULT=”Xen 4.2-amd64″/’*

*/etc/default/grub*

3. Disable apparmor at boot

*# sed -i ‘s/GRUB\_CMDLINE\_LINUX=.\*+/GRUB\_CMDLINE\_LINUX=”apparmor=0″/’*

*/etc/default/grub*

4. Restrict dom0 to 2GB of memory and 2 vcps

*# vi /etc/default/grub*

after GRUB\_CMDLINE\_LINUX=”apparmor=0″ add line

GRUB\_CMDLINE\_XEN=”dom0\_mem=2G,max:2G dom0\_max\_vcpus=2″

5. Update Grub with the config changes we just made

*# update-grub*

6. Once the server is back online ensure that Xen is running

*# cat /proc/xen/capabilities* - should display “control\_d”

if 'cat /proc/xen/capabilities' doesn't return anything, add the following line to /etc/fstab :

none /proc/xen xenfs defaults 0 0

And then do mount -a.

7. Setup the default toolstack

*# vi /etc/default/xen*

– set ‘TOOLSTACK=xapi’

8. Disable xend from starting at boot

*# sed -i -e ‘s/xend\_start$/#xend\_start/’ -e ‘s/xend\_stop$/#xend\_stop/’ /etc/init.d/xen*

NOTE: only xend the deamon needs to be disabled from starting, /etc/init.d/xend

handles other things like modules and xenfs. Do not disable it from the runlevel

9. Disable service xendomains

*# update-rc.d xendomains disable*

10.Fix for qemu which emulates the console does not have the keymaps in the correct location

*# mkdir /usr/share/qemu;*

*# ln -s /usr/share/qemu-linaro/keymaps /usr/share/qemu/keymaps*

11. check ifconfig command output

• If the ethernet device is recognized as em1 instead of eth0.

Then change it by adding 'biosdevname=0' to GRUB\_CMDLINE\_LINUX in /etc/default/grub then hit update-grub command and reboot machine.

12.Setup bridge networking

*# vi /etc/network/interfaces*

*# This file describes the network interfaces available on your system*

*# and how to activate them. For more information, see interfaces(5).*

*# The loopback network interface*

*auto lo*

*iface lo inet loopback# The primary network interface*

*auto xenbr0*

*iface xenbr0 inet static*

*address < eth0 ip address here >*

*netmask < eth0 netmask address here >*

*network < eth0 network address here >*

*broadcast < eth0 broadcast address here >*

*gateway*

*# dns-\* options are implemented by the resolvconf package, if installed*

*dns-nameservers 4.2.2.2*

*bridge\_ports eth0*

*iface eth0 inet manual*

13.Configure xcp to use bridge networking instead of openswitch

*# vi /etc/xcp/network.conf*

replace “openswitch” with “bridge”

14.All set – ready to reboot and let xcp-xapi toolstack take over

*# reboot*

15.On restart – confirm that xcp is working

*# xe vm-list*

*uuid ( RO) : 92ddb581-e6a8-2e6d-045e-d35b22f01668*

*name-label ( RW): Control domain on host: ramanujan*

*power-state ( RO): running*

If your output looks similar – xapi is running on the server, if you get “Connection

refused” then xapi is not setup correctly

## Setting up the LVM storage volume:

Assuming that you configured a large partition for use as a LVM volume during installation, this

part sets up the partition and adds it as a local storage repository. By default the volume group, and a logical volume will be created if

1. verify that you have a LVM partition

*# fdisk -l*

• This should list a partition of type “Linux LVM”. If you don’t see a partition

and you have free space on the disk, create a new partition of type “Linux

LVM” (8e)

• If you have partition of type “Linux LVM” follow the steps below

2. create a physical volumes

*# pvcreate /dev/cciss/c0d0p2*

*# pvdisplay*

*You should see similar output*

*“/dev/cciss/c0d0p2″ is a new physical volume of “947.60 GiB”*

*— NEW Physical volume —*

*PV Name /dev/cciss/c0d0p2*

*VG Name*

*PV Size 947.60 GiB*

*Allocatable NO*

*PE Size 0*

*Total PE 0*

*Free PE 0*

*Allocated PE 0*

*PV UUID rNeGnf-TbJS-vfSm-t7la-wNCv-Lpc3-vjn33c*

3. create a volume group

*# vgcreate VolumeGroup /dev/cciss/c0d0p2*

*# pvdisplay - this should display the volume group we created on the*

*physical volume*

*— Physical volume —*

*PV Name /dev/cciss/c0d0p2*

*VG Name VolumeGroup*

*PV Size 947.60 GiB / not usable 2.90 MiB*

*Allocatable yes*

*PE Size 4.00 MiB*

*Total PE 242584*

*Free PE 242584*

*Allocated PE 0*

*PV UUID rNeGnf-TbJS-vfSm-t7la-wNCv-Lpc3-vjn33c*

4. create a logical volume on “VolumeGroup”

*# lvcreate –size 947G -n LocalStorage VolumeGroup*

*# lvdisplay* – this should display the logical volume we created on the volume group

*— Logical volume —*

*LV Name /dev/VolumeGroup/LocalStorage*

*VG Name VolumeGroup*

*LV UUID pCWgAs-cpfh-IAdU-uVMi-EJbo-iy2x-TlMzar*

*LV Write Access read/write*

*LV Status available*

*# open 0*

*LV Size 947.00 GiB*

*Current LE 242432*

*Segments 1*

*Allocation inherit*

*Read ahead sectors auto*

*- currently set to 256*

*Block device 252:0*

5. Register the logical volume for use with XAPI

*xe sr-create type=ext name-label=Local Storage deviceconfig:device=/dev/mapper/VolumeGroup-LocalStorage*

– this will take a while if the volume is large

*# xe sr-list name-label="Local Storage"*

*– this should display the storage repository*

*uuid ( RO) : 7dea0028-ee94-6c16-2f61-c699ed4a1d18*

*name-label ( RW): Local Storage*

*name-description ( RW):*

*host ( RO): ubuntu-xenserver-1*

*type ( RO): ext*

*content-type ( RO):*

*# xe pool-param-set uuid=<pool-uuid> default-SR=<sr-uuid>*

Get the pool-uuid from- xe pool-list

and get sr-uuid from - xe sr-list

## Setup TPM :

**Note:** Following instrcutions to setup TPM via BIOS are valid for Dell Latitude E6530 and may/may not work for you. In case these instruction don’t work for you, please refer to your server’s documentation for setting up TPM.

* There are two settings in BIOS that need to be enabled : TPM Security and TXT Execution.
* In BIOS (use F2), go to Security -> TPM Security and click the check box to enable TPM Security and click on Apply.
* This enables TPM Security but TPM is still in deactivated state. Select the Activate radio button and click on Apply to activate TPM. Click Exit.
* Get to the BIOS again and go to Virtualization -> TXT Execution/Technology and click on the check-box to enable it. Click on Apply and then Exit to save the settings.

1. Install the components needed for tpm and txt as follows :

*# apt-get install tpm-tools tpm-tools-dbg trousers libtspi1 libtspi-dev*

2. Once all these are installed, check with tpm\_version and tpm\_selftest to see if you can access the TPM. The output should be something like this :

*# tpm\_version*

*TPM 1.2 Version Info:*

*Chip Version: 1.2.37.19*

*Spec Level: 2*

*Errata Revision: 3*

*TPM Vendor ID: ATML*

*TPM Version: 01010000*

*Manufacturer Info: 41544d4c*

*# tpm\_selftest*

*TPM Test Results: 0000*

**Note**: Trousers should be running for this. Check with */etc/init.d/trousers status* or *ps ax | grep tcsd*. Restart using */etc/init.d/trousers restart.*

3. If TPM is working, take ownership using *tpm\_takeownership -z -y* command. Then run *tpm\_getpubek -z* to see if you get the public encryption key. If you are not able to get the key or take ownership of TPM, then TPM needs to be reset. You can do it either by using the tpm\_clear command ( needs reboot ) or by resetting TPM is BIOS. If everything is OK, then it means TPM is setup correctly.

4. You will need to have the SINIT file 3rd\_gen\_i5\_i7\_SINIT\_67.BIN downloaded to /boot from here : http://software.intel.com/en-us/articles/intel-trusted-execution-technology

5. Setup tboot by using the following command :

*# apt-get install tboot.*

This sets up tboot in grub and makes all available kernels to boot with tboot and adds entry for SINIT file as well. The grub.cfg will have all module parameters twice which is required for tboot.

7. Reboot the machine and select Xen with tboot as your boot option.

8. Once booted, check with *txt-stat* command to see if you have booted in an measured environment. If not check if you have the correct SINIT file and reboot again.

## Setup RPCore and xapi proxy :

1. Download RPCore and xapi\_proxy (latest from wiki/build server) and place it in a suitable location :

*# cd /root/*

*# scp -r root@<build-server-ip:path>/RPCore/ /root/*

*# scp -r root@<build-server-ip:path>/xapi\_proxy/ /root/*

2. Add the following to the /etc/bash.bashrc file to set the LD\_LIBRARY\_PATH :

*export LD\_LIBRARY\_PATH=/root/RPCore/lib:/root/xapi\_proxy/:$LD\_LIBRARY\_PATH*

*export IP\_ADDRESS=<dom0\_ip>*

*export RPCORE\_IPADDR=<dom0\_ip>*

*export RPCORE\_PORT=16005*

3. Add the following to /etc/rc.local :

*cp -r /root/RPCore/rptmp/ /tmp/*

*cd /root/RPCore/bin/debug/*

*./rpcorex &*

*cd /root/xapi\_proxy/*

*chmod a+x xapi\_proxy.py*

*nohup python xapi\_proxy.py &*

## Install nova plugins:

1. Create temporary files/directories:

*# NOVA\_ZIPBALL=$(mktemp)*

*# NOVA\_SOURCES=$(mktemp -d)*

2. Get the source from github. The example assumes the master branch is used, please amend the URL to match the version being used:

*# wget -qO "$NOVA\_ZIPBALL" https://github.com/openstack/nova/archive/master.zip*

*unzip "$NOVA\_ZIPBALL" -d "$NOVA\_SOURCES"*

3. Copy the plugins to the xapi plugin directory

*# PLUGINPATH=$(find $NOVA\_SOURCES -path '\*/xapi.d/plugins' -type d -print)*

*# cp -r $PLUGINPATH/\* /usr/lib/xcp/plugins/.*

4. Remove the temporary files/directories:

*# rm "$NOVA\_ZIPBALL"*

*# rm -rf "$NOVA\_SOURCES"*

5. We need to make some changes in plugin to make it work with our OS. Open the file

**/usr/lib/xcp/plugins/xenhost** and go to the function **def host\_data(self, arg\_dict)**Comment the following two lines

*config = \_get\_config\_dict()*

*ret\_dict.update(config)*

## Setting up Ubuntu 12.04 LTS as domU:

This document assumes the following :

* The server is running Ubuntu 13.04 and is booted in Xen 4.2.
* The server uses xcp-xapi/xe toolstack

Please follow the instructions below to setup a Ubuntu 12.04 VM using xe :

1. Check if xcp-xapi service is running and xe toolstack is available as follows :

*root@aguna:/home# service xcp-xapi status*

*\* xapi is running*

*root@aguna:/home#*

*root@aguna:/home# xe vm-list*

*uuid ( RO) : ed9de35c-142d-8543-c9c9-5ea5b4bad988*

*name-label ( RW): Control domain on host: aguna*

*power-state ( RO): running*

*root@aguna:/home#*

2. Check if there are any templates available :

*root@aguna:~# xe template-list*

If there are no templates then invoke the following command to create templates :

*root@aguna:~# /usr/lib/xcp/lib/create\_templates*

3. Create VM

a) Create VM with Ubuntu Lucid template and later we will set it to Precise(Ubuntu 12.04)

*root@aguna:~# VM\_ID=$(xe vm-install template=Ubuntu\ Lucid\ Lynx\ 10.04\ \(64-bit\) sr-name-label=<sr-name> new-name-label=<vmname>)*

**Note:** Specify the correct *sr-name-label* using *xe sr-list.*

b) Get UUID of xenbr0 network

*root@aguna:~# NET\_ID=$(xe network-list bridge=xenbr0 --minimal)*

c) Create virtual interface with xenbr0 network

*root@aguna:~# xe vif-create mac=random device=0 network-uuid=$NET\_ID vm-uuid=$VM\_ID*

d) Set the following vm-params :

*root@aguna:~# xe vm-param-set uuid=$VM\_ID other-config:install-repository="http://us.archive.ubuntu.com/ubuntu"*

*root@aguna:~# xe vm-param-set uuid=$VM\_ID PV-args="netcfg/get\_hostname=Ubuntu \ console=hvc0 debian-installer/locale=en\_US console-setup/layoutcode=us \ console-setup/ask\_detect=false interface=eth0 netcfg/disable\_dhcp=false \ preseed/file=/root/preseed-ubuntu-12.04.cfg"*

*root@aguna:~# xe vm-param-set uuid=${VM\_ID} other-config:disable\_pv\_vnc=1*

**Note:** Get the preseed-ubuntu-12.04.cfg from Annex 1 at the end of this document.

e) Set disk size for new VM

**Note:** Here it is being set to 8 GB - please check available size and chose the appropriate size for your deployment.

*root@aguna:~# xe vm-param-set other-config:disks="<provision><disk device=\"0\" \ size=\"8589934592\" sr=\"\" bootable=\"true\" type=\"system\"/></provision>" \ uuid=$VM\_ID*

f) Set the memory parameters

*root@aguna:~# xe vm-param-set memory-static-max=1024000000 uuid=$VM\_ID*

*root@aguna:~# xe vm-param-set memory-dynamic-max=1024000000 uuid=$VM\_ID*

*root@aguna:~# xe vm-param-set memory-dynamic-min=512000000 uuid=$VM\_ID*

*root@aguna:~# xe vm-param-set memory-static-min=512000000 uuid=$VM\_ID*

g) Start VM

*root@dom0:~# xe vm-start uuid=$VM\_ID*

h) Access console and continue installation

*root@dom0:~# xe vm-lis*t

It it shows vm in running state then we can access console of VM and continue Ubuntu Precise installation. Get domain-id from the following command :

*root@dom0:/# xe vm-param-list uuid=$VM\_ID | grep dom-id*

Access the console using either :

*root@dom0:/# /usr/lib/xen-4.2/bin/xenconsole <domain-id>*

OR

*root@dom0:/# xe console vm=<VM\_NAME>*

Note : You can get the *<VM\_NAME>* from *xe vm-list.*

Then continue with OS installation.

Once the setup is done and the VM boots up, login with the user *ubuntu* and password as *password.*

Annex 1

* Copy the following contents to /root/preceed-ubuntu-12.04.cfg file and save it :

#### Contents of the preconfiguration file (for squeeze)

### Localization

# Preseeding only locale sets language, country and locale.

d-i debian-installer/locale string en\_US

# The values can also be preseeded individually for greater flexibility.

#d-i debian-installer/language string en

#d-i debian-installer/country string NL

#d-i debian-installer/locale string en\_GB.UTF-8

# Optionally specify additional locales to be generated.

#d-i localechooser/supported-locales en\_US.UTF-8, nl\_NL.UTF-8

# Keyboard selection.

# Disable automatic (interactive) keymap detection.

d-i console-setup/ask\_detect boolean false

#d-i keyboard-configuration/modelcode string pc105

d-i keyboard-configuration/layoutcode string us

# To select a variant of the selected layout (if you leave this out, the

# basic form of the layout will be used):

#d-i keyboard-configuration/variantcode string dvorak

### Network configuration

# Disable network configuration entirely. This is useful for cdrom

# installations on non-networked devices where the network questions,

# warning and long timeouts are a nuisance.

#d-i netcfg/enable boolean false

# netcfg will choose an interface that has link if possible. This makes it

# skip displaying a list if there is more than one interface.

#d-i netcfg/choose\_interface select auto

# To pick a particular interface instead:

d-i netcfg/choose\_interface select eth0

# If you have a slow dhcp server and the installer times out waiting for

# it, this might be useful.

#d-i netcfg/dhcp\_timeout string 60

# If you prefer to configure the network manually, uncomment this line and

# the static network configuration below.

#d-i netcfg/disable\_autoconfig boolean true

# If you want the preconfiguration file to work on systems both with and

# without a dhcp server, uncomment these lines and the static network

# configuration below.

#d-i netcfg/dhcp\_failed note

#d-i netcfg/dhcp\_options select Configure network manually

# Static network configuration.

#d-i netcfg/get\_nameservers string 192.168.1.1

#d-i netcfg/get\_ipaddress string 192.168.1.42

#d-i netcfg/get\_netmask string 255.255.255.0

#d-i netcfg/get\_gateway string 192.168.1.1

#d-i netcfg/confirm\_static boolean true

# Any hostname and domain names assigned from dhcp take precedence over

# values set here. However, setting the values still prevents the questions

# from being shown, even if values come from dhcp.

d-i netcfg/get\_hostname string unassigned-hostname

d-i netcfg/get\_domain string unassigned-domain

# Disable that annoying WEP key dialog.

d-i netcfg/wireless\_wep string

# The wacky dhcp hostname that some ISPs use as a password of sorts.

#d-i netcfg/dhcp\_hostname string radish

# If non-free firmware is needed for the network or other hardware, you can

# configure the installer to always try to load it, without prompting. Or

# change to false to disable asking.

#d-i hw-detect/load\_firmware boolean true

### Network console

# Use the following settings if you wish to make use of the network-console

# component for remote installation over SSH. This only makes sense if you

# intend to perform the remainder of the installation manually.

#d-i anna/choose\_modules string network-console

#d-i network-console/password password r00tme

#d-i network-console/password-again password r00tme

### Mirror settings

# If you select ftp, the mirror/country string does not need to be set.

#d-i mirror/protocol string ftp

d-i mirror/country string manual

d-i mirror/http/hostname string archive.ubuntu.com

d-i mirror/http/directory string /ubuntu

d-i mirror/http/proxy string

# Alternatively: by default, the installer uses CC.archive.ubuntu.com where

# CC is the ISO-3166-2 code for the selected country. You can preseed this

# so that it does so without asking.

#d-i mirror/http/mirror select CC.archive.ubuntu.com

# Suite to install.

#d-i mirror/suite string squeeze

# Suite to use for loading installer components (optional).

#d-i mirror/udeb/suite string squeeze

# Components to use for loading installer components (optional).

#d-i mirror/udeb/components multiselect main, restricted

### Clock and time zone setup

# Controls whether or not the hardware clock is set to UTC.

d-i clock-setup/utc boolean true

# You may set this to any valid setting for $TZ; see the contents of

# /usr/share/zoneinfo/ for valid values.

d-i time/zone string US/Pacific

# Controls whether to use NTP to set the clock during the install

d-i clock-setup/ntp boolean true

# NTP server to use. The default is almost always fine here.

#d-i clock-setup/ntp-server string ntp.example.com

### Partitioning

## Partitioning example

# If the system has free space you can choose to only partition that space.

# This is only honoured if partman-auto/method (below) is not set.

# Alternatives: custom, some\_device, some\_device\_crypto, some\_device\_lvm.

#d-i partman-auto/init\_automatically\_partition select biggest\_free

# Alternatively, you may specify a disk to partition. If the system has only

# one disk the installer will default to using that, but otherwise the device

# name must be given in traditional, non-devfs format (so e.g. /dev/hda or

# /dev/sda, and not e.g. /dev/discs/disc0/disc).

# For example, to use the first SCSI/SATA hard disk:

#d-i partman-auto/disk string /dev/sda

# In addition, you'll need to specify the method to use.

# The presently available methods are:

# - regular: use the usual partition types for your architecture

# - lvm: use LVM to partition the disk

# - crypto: use LVM within an encrypted partition

d-i partman-auto/method string regular

# If one of the disks that are going to be automatically partitioned

# contains an old LVM configuration, the user will normally receive a

# warning. This can be preseeded away...

d-i partman-lvm/device\_remove\_lvm boolean true

# The same applies to pre-existing software RAID array:

d-i partman-md/device\_remove\_md boolean true

# And the same goes for the confirmation to write the lvm partitions.

d-i partman-lvm/confirm boolean true

# For LVM partitioning, you can select how much of the volume group to use

# for logical volumes.

#d-i partman-auto-lvm/guided\_size string max

#d-i partman-auto-lvm/guided\_size string 10GB

#d-i partman-auto-lvm/guided\_size string 50%

# You can choose one of the three predefined partitioning recipes:

# - atomic: all files in one partition

# - home: separate /home partition

# - multi: separate /home, /usr, /var, and /tmp partitions

d-i partman-auto/choose\_recipe select atomic

# Or provide a recipe of your own...

# If you have a way to get a recipe file into the d-i environment, you can

# just point at it.

#d-i partman-auto/expert\_recipe\_file string /hd-media/recipe

# If not, you can put an entire recipe into the preconfiguration file in one

# (logical) line. This example creates a small /boot partition, suitable

# swap, and uses the rest of the space for the root partition:

#d-i partman-auto/expert\_recipe string \

# boot-root :: \

# 40 50 100 ext3 \

# $primary{ } $bootable{ } \

# method{ format } format{ } \

# use\_filesystem{ } filesystem{ ext3 } \

# mountpoint{ /boot } \

# . \

# 500 10000 1000000000 ext3 \

# method{ format } format{ } \

# use\_filesystem{ } filesystem{ ext3 } \

# mountpoint{ / } \

# . \

# 64 512 300% linux-swap \

# method{ swap } format{ } \

# .

# If you just want to change the default filesystem from ext3 to something

# else, you can do that without providing a full recipe.

#d-i partman/default\_filesystem string ext4

# The full recipe format is documented in the file partman-auto-recipe.txt

# included in the 'debian-installer' package or available from D-I source

# repository. This also documents how to specify settings such as file

# system labels, volume group names and which physical devices to include

# in a volume group.

# This makes partman automatically partition without confirmation, provided

# that you told it what to do using one of the methods above.

d-i partman-partitioning/confirm\_write\_new\_label boolean true

d-i partman/choose\_partition select finish

d-i partman/confirm boolean true

d-i partman/confirm\_nooverwrite boolean true

## Partitioning using RAID

# The method should be set to "raid".

#d-i partman-auto/method string raid

# Specify the disks to be partitioned. They will all get the same layout,

# so this will only work if the disks are the same size.

#d-i partman-auto/disk string /dev/sda /dev/sdb

# Next you need to specify the physical partitions that will be used.

#d-i partman-auto/expert\_recipe string \

# multiraid :: \

# 1000 5000 4000 raid \

# $primary{ } method{ raid } \

# . \

# 64 512 300% raid \

# method{ raid } \

# . \

# 500 10000 1000000000 raid \

# method{ raid } \

# .

# Last you need to specify how the previously defined partitions will be

# used in the RAID setup. Remember to use the correct partition numbers

# for logical partitions. RAID levels 0, 1, 5, 6 and 10 are supported;

# devices are separated using "#".

# Parameters are:

# <raidtype> <devcount> <sparecount> <fstype> <mountpoint> \

# <devices> <sparedevices>

#d-i partman-auto-raid/recipe string \

# 1 2 0 ext3 / \

# /dev/sda1#/dev/sdb1 \

# . \

# 1 2 0 swap - \

# /dev/sda5#/dev/sdb5 \

# . \

# 0 2 0 ext3 /home \

# /dev/sda6#/dev/sdb6 \

# .

# For additional information see the file partman-auto-raid-recipe.txt

# included in the 'debian-installer' package or available from D-I source

# repository.

# This makes partman automatically partition without confirmation.

d-i partman-md/confirm boolean true

d-i partman-partitioning/confirm\_write\_new\_label boolean true

d-i partman/choose\_partition select finish

d-i partman/confirm boolean true

d-i partman/confirm\_nooverwrite boolean true

## Controlling how partitions are mounted

# The default is to mount by UUID, but you can also choose "traditional" to

# use traditional device names, or "label" to try filesystem labels before

# falling back to UUIDs.

#d-i partman/mount\_style select uuid

### Base system installation

# Configure APT to not install recommended packages by default. Use of this

# option can result in an incomplete system and should only be used by very

# experienced users.

#d-i base-installer/install-recommends boolean false

# The kernel image (meta) package to be installed; "none" can be used if no

# kernel is to be installed.

#d-i base-installer/kernel/image string linux-generic

### Account setup

# Skip creation of a root account (normal user account will be able to

# use sudo). The default is false; preseed this to true if you want to set

# a root password.

#d-i passwd/root-login boolean false

# Alternatively, to skip creation of a normal user account.

#d-i passwd/make-user boolean false

# Root password, either in clear text

#d-i passwd/root-password password r00tme

#d-i passwd/root-password-again password r00tme

# or encrypted using an MD5 hash.

#d-i passwd/root-password-crypted password [MD5 hash]

# To create a normal user account.

d-i passwd/user-fullname string Ubuntu User

d-i passwd/username string ubuntu

# Normal user's password, either in clear text

d-i passwd/user-password password password

d-i passwd/user-password-again password password

# or encrypted using an MD5 hash.

#d-i passwd/user-password-crypted password [MD5 hash]

# Create the first user with the specified UID instead of the default.

#d-i passwd/user-uid string 1010

# The installer will warn about weak passwords. If you are sure you know

# what you're doing and want to override it, uncomment this.

d-i user-setup/allow-password-weak boolean true

# The user account will be added to some standard initial groups. To

# override that, use this.

#d-i passwd/user-default-groups string audio cdrom video

# Set to true if you want to encrypt the first user's home directory.

d-i user-setup/encrypt-home boolean false

### Apt setup

# You can choose to install restricted and universe software, or to install

# software from the backports repository.

#d-i apt-setup/restricted boolean true

#d-i apt-setup/universe boolean true

#d-i apt-setup/backports boolean true

# Uncomment this if you don't want to use a network mirror.

#d-i apt-setup/use\_mirror boolean false

# Select which update services to use; define the mirrors to be used.

# Values shown below are the normal defaults.

#d-i apt-setup/services-select multiselect security

#d-i apt-setup/security\_host string security.ubuntu.com

#d-i apt-setup/security\_path string /ubuntu

# Additional repositories, local[0-9] available

#d-i apt-setup/local0/repository string \

# http://local.server/ubuntu squeeze main

#d-i apt-setup/local0/comment string local server

# Enable deb-src lines

#d-i apt-setup/local0/source boolean true

# URL to the public key of the local repository; you must provide a key or

# apt will complain about the unauthenticated repository and so the

# sources.list line will be left commented out

#d-i apt-setup/local0/key string http://local.server/key

# By default the installer requires that repositories be authenticated

# using a known gpg key. This setting can be used to disable that

# authentication. Warning: Insecure, not recommended.

#d-i debian-installer/allow\_unauthenticated boolean true

### Package selection

tasksel tasksel/first multiselect ubuntu-desktop

#tasksel tasksel/first multiselect lamp-server, print-server

#tasksel tasksel/first multiselect kubuntu-desktop

# Individual additional packages to install

#d-i pkgsel/include string openssh-server build-essential

# Whether to upgrade packages after debootstrap.

# Allowed values: none, safe-upgrade, full-upgrade

#d-i pkgsel/upgrade select none

# Language pack selection

#d-i pkgsel/language-packs multiselect de, en, zh

# Policy for applying updates. May be "none" (no automatic updates),

# "unattended-upgrades" (install security updates automatically), or

# "landscape" (manage system with Landscape).

d-i pkgsel/update-policy select none

# Some versions of the installer can report back on what software you have

# installed, and what software you use. The default is not to report back,

# but sending reports helps the project determine what software is most

# popular and include it on CDs.

#popularity-contest popularity-contest/participate boolean false

# By default, the system's locate database will be updated after the

# installer has finished installing most packages. This may take a while, so

# if you don't want it, you can set this to "false" to turn it off.

#d-i pkgsel/updatedb boolean true

### Boot loader installation

# Grub is the default boot loader (for x86). If you want lilo installed

# instead, uncomment this:

#d-i grub-installer/skip boolean true

# To also skip installing lilo, and install no bootloader, uncomment this

# too:

#d-i lilo-installer/skip boolean true

# With a few exceptions for unusual partitioning setups, GRUB 2 is now the

# default. If you need GRUB Legacy for some particular reason, then

# uncomment this:

#d-i grub-installer/grub2\_instead\_of\_grub\_legacy boolean false

# This is fairly safe to set, it makes grub install automatically to the MBR

# if no other operating system is detected on the machine.

d-i grub-installer/only\_debian boolean true

# This one makes grub-installer install to the MBR if it also finds some other

# OS, which is less safe as it might not be able to boot that other OS.

d-i grub-installer/with\_other\_os boolean true

# Alternatively, if you want to install to a location other than the mbr,

# uncomment and edit these lines:

#d-i grub-installer/only\_debian boolean false

#d-i grub-installer/with\_other\_os boolean false

#d-i grub-installer/bootdev string (hd0,0)

# To install grub to multiple disks:

#d-i grub-installer/bootdev string (hd0,0) (hd1,0) (hd2,0)

# Optional password for grub, either in clear text

#d-i grub-installer/password password r00tme

#d-i grub-installer/password-again password r00tme

# or encrypted using an MD5 hash, see grub-md5-crypt(8).

#d-i grub-installer/password-crypted password [MD5 hash]

# Use the following option to add additional boot parameters for the

# installed system (if supported by the bootloader installer).

# Note: options passed to the installer will be added automatically.

#d-i debian-installer/add-kernel-opts string nousb

### Finishing up the installation

# During installations from serial console, the regular virtual consoles

# (VT1-VT6) are normally disabled in /etc/inittab. Uncomment the next

# line to prevent this.

#d-i finish-install/keep-consoles boolean true

# Avoid that last message about the install being complete.

d-i finish-install/reboot\_in\_progress note

# This will prevent the installer from ejecting the CD during the reboot,

# which is useful in some situations.

#d-i cdrom-detect/eject boolean false

# This is how to make the installer shutdown when finished, but not

# reboot into the installed system.

#d-i debian-installer/exit/halt boolean true

# This will power off the machine instead of just halting it.

#d-i debian-installer/exit/poweroff boolean true

### X configuration

# X can detect the right driver for some cards, but if you're preseeding,

# you override whatever it chooses. Still, vesa will work most places.

#xserver-xorg xserver-xorg/config/device/driver select vesa

# A caveat with mouse autodetection is that if it fails, X will retry it

# over and over. So if it's preseeded to be done, there is a possibility of

# an infinite loop if the mouse is not autodetected.

#xserver-xorg xserver-xorg/autodetect\_mouse boolean true

# Monitor autodetection is recommended.

xserver-xorg xserver-xorg/autodetect\_monitor boolean true

# Uncomment if you have an LCD display.

#xserver-xorg xserver-xorg/config/monitor/lcd boolean true

# X has three configuration paths for the monitor. Here's how to preseed

# the "medium" path, which is always available. The "simple" path may not

# be available, and the "advanced" path asks too many questions.

xserver-xorg xserver-xorg/config/monitor/selection-method \

select medium

xserver-xorg xserver-xorg/config/monitor/mode-list \

select 1024x768 @ 60 Hz

### Preseeding other packages

# Depending on what software you choose to install, or if things go wrong

# during the installation process, it's possible that other questions may

# be asked. You can preseed those too, of course. To get a list of every

# possible question that could be asked during an install, do an

# installation, and then run these commands:

# debconf-get-selections --installer > file

# debconf-get-selections >> file

#### Advanced options

### Running custom commands during the installation

# d-i preseeding is inherently not secure. Nothing in the installer checks

# for attempts at buffer overflows or other exploits of the values of a

# preconfiguration file like this one. Only use preconfiguration files from

# trusted locations! To drive that home, and because it's generally useful,

# here's a way to run any shell command you'd like inside the installer,

# automatically.

# This first command is run as early as possible, just after

# preseeding is read.

#d-i preseed/early\_command string anna-install some-udeb

# This command is run immediately before the partitioner starts. It may be

# useful to apply dynamic partitioner preseeding that depends on the state

# of the disks (which may not be visible when preseed/early\_command runs).

#d-i partman/early\_command \

# string debconf-set partman-auto/disk "$(list-devices disk | head -n1)"

# This command is run just before the install finishes, but when there is

# still a usable /target directory. You can chroot to /target and use it

# directly, or use the apt-install and in-target commands to easily install

# packages and run commands in the target system.

#d-i preseed/late\_command string apt-install zsh; in-target chsh -s /bin/zsh

d-i preseed/late\_command string in-target perl -pi -e 's/(errors=remount-ro)/noatime,nodiratime,$1,barrier=0/' /etc/fstab