OpenVZ

виртуализирующее Linux-ядро

практическое руководство на русском языке

компиляция из общедоступных источников
перевод - Майк "Граф Ноль" Ламберт.

Содержание

1.Введение

1.1 Краткое содержание руководства.

Данный документ предназначен для освоения и решения практических вопросов, возникающих при работе с системой виртуализации OpenVZ. Документ является переводом с английского сборника материалов из открытых источников по OpenVZ.

Переводчик не несет ответственности за несоответствии текущей версии руководства актуальной версии OpenVZ.

1.2. Целевая аудитория.

Целевой аудиторией данного руководства являются лица, заинтересованные в использовании и обслуживании OpenVZ. Подразумевается, что читатель имеет начальные представления о виртуализации, а также навыки администрирования Linux-систем из командной строки.

1.3 Краткая историческая справка.

OpenVZ является системой виртуализации уровня операционной системы. В отличие от систем виртуализации уровня оборудования (VMware, Xen, Microsoft Virtual Server), OpenVZ может виртуализировать только unix-like системы (Linux, xBSD, Solaris), так как базируется на измененном ядре ОС. Это накладывает существенно меньшие расходы на виртуализацию, чем виртуализация аппаратного уровня.

OpenVZ основывается на коммерческом ядре Virtuozzo от компании SWSoft. Можно сказать, что OpenVZ - это облегченное и бесплатное Virtuozzo. Основные отличия OpenVZ от Virtuozzo:

feature set	OpenVZ	Virtuozzo
Основная ОС	Linux, Solaris, xBSD	Linux, Solaris, xBSD, Windows Server
Name-based хостинг	нет	да
Лицензия	GPL	Proprietary
Интеграция с Plesk, HSPComplete	неизвестно	да
Поставка	RPM, deb, Source TGZ	Out-Of-The-Box.

1.4 От переводчика

Кроме оригинального переведенного материала в тексте будут встречаться мои комментарии.

Они будут выделены вот так

1.5 Определения, встречающиеся в тексте.

Аппаратный узел, хост, Hardware Node, HN, HW, VE0, host - физический компьютер, на котором установлено ядро OpenVZ.

VE, VPS, VDS, виртуальная среда, виртуальный сервер, виртуальный частный сервер, виртуальный выделенный сервер - независимая программная сущность, располагающаяся на аппаратном узле, работающая под управлением OpenVZ, и проявляющая себя как отдельный Linux-сервер.

VE area, private area, приватная зона VE - дисковый каталог, предназначенный для хранения данных и ПО конкретного VE.

OS template, VE template, шаблон ОС, шаблон VE - архивный файл, содержащий необходимое ПО и данные для развертывания типового VE с конкретной ОС и приложениями.

Template cache, кэш шаблонов - дисковый каталог, содержащий архивы с различными шаблонами.

"Заморозка", freeze - процесс сохранения полного состояния VE, включая запущенные процессы и сетевые соединения для последующего адекватного восстановления состояния.

2. Быстрый старт.

2.1 Установка OpenVZ на RedHat-дистрибутивы.

Требования

Подразумевается ,что вы используете Fedora Core 5 или RedHat Enterprise Linux/CentOS 4. На текущий момент ядро OpenVZ пытается поддерживать то же оборудование, что и ядра RedHat. Для подробных сведений о совместимом оборудовании, обратитесь к Virtuozzo HCL.

Файловая система

Рекомендуется использовать отдельный раздел для хранения данных VE. (по умолчанию /vz/private/<veid>). Причина этого в том, что если вы захотите использовать дисковые квоты на каждый VE, используемые OpenVZ, вы не сможете использовать обычные квоты Linux. На том же разделе. В данном контексте OpenVZ-квота не является чистой квотой, она действует как обычная дисковая квота, но в рамках VE, а не аппаратного узла.
Избегайте использовать корневой раздел для хранения данных VE. В такой ситуации администратор VE может превысить дисковый барьер в 5%A, полностью заполнить корневой раздел аппаратного узла, и вызвать крах системы. Квоты OpenVZ поддерживаются только для файловых систем ext2/ext3

rpm или yum?

В том случае, если на вашей системе установлен yum, вы можете эффективно устанавливать и обновлять OpenVZ. Если вы не хотите использовать yum, или он отсутствует, вы используете rpm. Детальное руководство по yum и rpm смотрите ниже.

Конфигурирование yum

Сначала настройте репозиторий OpenVZ для yum.

Загрузите файл openvz.repo в каталог /etc/yum.repos.d/ с правами администратора системы .

```
# cd /etc/yum.repos.d
# wget http://download.openvz.org/openvz.repo
# rpm --import  http://download.openvz.org/RPM-GPG-Key-OpenVZ
```

Если вы не можете найти этот каталог, значит yum на вашей системе не установлен, или он слишком ранней версии. Используйте rpm.

Установка ядра

Для начала определитесь с типом ядра, подробнее смотрите Kernel flavors.

С использованием yum

```
# yum install ovzkernel[-flavor]
```

Где [-flavor] необязателен и может быть **-smp** или **-enterprise**. Смотрите kernel flavors для подробной информации.

С использованием rpm

Загрузите бинарные пакеты RPM со страницы Download » Kernel, или напрямую с download.openvz.org/kernel, или одного из зеркал (mirrors). Вам потребуется только один RPM-пакет, в зависимости от вашего оборудования.

Установите ядро:

```
# rpm -ihv ovzkernel[-flavor]*.rpm
```

Где [-flavor] необязателен и может быть **-smp** или **-enterprise**. Смотрите kernel flavors для подробной информации.

> **Примечание**: **не используйте** rpm с ключом -U (где -U подразумевает *upgrade*), иначе все установленные в системе ядра буду деинсталлированы.

Настройка загрузчика

В случае, если вы используете загрузчик GRUB, добавьте следующие строки в файл **/boot/grub/grub.conf**:

```
title Fedora Core (2.6.8-022stab029.1)
    root (hd0,0)
    kernel /vmlinuz-2.6.8-022stab029.1 ro root=/dev/sda5 quiet rhgb vga=0x31B
    initrd /initrd-2.6.8-022stab029.1.img
```

Смените ***Fedora Core*** на ***OpenVZ*** (для того чтобы различать ядра при загрузке, ядра OpenVZ не оказывают влияния на невиртуализирующие ядра). Уберите дополнительные параметры ядра из командной строки, оставив только параметр ***root=***…. Именованная часть файла ***/etc/grub.conf*** должна выглядеть примерно так:

```
title OpenVZ (2.6.8-022stab029.1)
    root (hd0,0)
    kernel /vmlinuz-2.6.8-022stab029.1 ro root=/dev/sda5
    initrd /initrd-2.6.8-022stab029.1.img
```

Настройка системы

Делайте это обязательно перед тем, как загрузитесь с ядром OpenVZ.

sysctl

Некоторое количество параметров ядра должно быть изменено для корректной работы OpenVZ. Эти параметры хранятся в файле ***/etc/sysctl.conf*** . Отредактированная часть должна выглядеть примерно так.

```
# On Hardware Node we generally need
# packet forwarding enabled and proxy arp disabled
net.ipv4.ip_forward = 1
net.ipv4.conf.default.proxy_arp = 0
# Enables source route verification
net.ipv4.conf.all.rp_filter = 1
# Enables the magic-sysrq key
kernel.sysrq = 1
# TCP Explict Congestion Notification
#net.ipv4.tcp_ecn = 0
# we do not want all our interfaces to send redirects
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
```

SELinux

SELinux должен быть отключен. Для этого поместите следующую строку в файл ***/etc/sysconfig/selinux***:

SELINUX=disabled

Отслеживание соединений

В стабильных ядрах OpenVZ (2.6.8-based) отслеживание соединений netfilter для VE0 по умолчанию запрещено. Если у вас установлен stateful файрволл на аппаратном узле (это установка по умолчанию) вы должны отключить его, или разрешить отслеживание соединений для VE0.

Для разрешения остлеживания соединений добавьте следующую строку в файл ***/etc/modprobe.conf:***

options ip_conntrack ip_conntrack_enable_ve0=1

> **Примечание**: в ядрах, позднее чем 2.6.8, остлеживание соединений разрешено по умолчанию

Перезагрузка с ядром OpenVZ

Перезагрузите компьютер и выберите "OpenVZ" в меню загрузчика. Если ядро OpenVZ kernel загрузилось успешно, переходите к установке инструментария.

Установка инструментария

OpenVZ использует некоторое количество пользовательских утилит:

vzctl
: утилита для манипулирования VPS (создание, уничтожение, установка параметров …)

vzquota
: утилита для квотирования VPSs. Чаще используется неявно через vzctl.

yum

 # yum install vzctl vzquota

rpm

Загрузите RPMs с утилитами со страницы Download » Utils, или напрямую с download.openvz.org/utils, или его зеркал (mirrors). Выполните команду:

 # rpm -Uhv vzctl*.rpm vzquota*.rpm

Если rpm сообщит об отсутсвующих зависимостях, установите их, затем повторите установку.

После установки ядра и инструментария мы можем запускать подсистему OpenVZ.

Запуск OpenVZ

Под администратором выполните команду:

 # /sbin/service vz start

Эта команда загрузит все необходимые ядерные модули. Кроме того, все автозапускаемые VPS будут запущены. При следующей перезагрузке скрипт сработает автоматически.

Заключение

OpenVZ установлена на ваш компьютер. Для загрузки ядра OpenVZ по умолчанию, исправьте файл ***/boot/grub/grub.conf*** так, чтоб он указывал на ядро OpenVZ. Например, если ядро OpenVZ первое в списке, напишите параметр «default» равным 0: ***default 0***. Подробнее смотрите справку ***man grub.conf***.

2.2 Установка OpenVZ на Debian-дистрибутивы.

Sarge

Пакеты OpenVZ на http://debian.systs.org/ нацелены на легкий, быстрый способ установки.

Изменение настройки источников apt

Добавьте в файл ***/etc/apt/sources.list***

deb http://debian.systs.org/ stable openvz

и обновите список пакетов

 # apt-get update

Скомпилированные ядра на debian.systs.org (dso)

Ядра, расположенные debian.systs.org (dso) используют ту же конфигурацию, что и ядра с оригинального сайта OpenVZ. (большинство модулей вкомпилировано в ядра!)

Ядра для архитектур i386 и amd64

ovzkernel-2.6.9
ovzkernel-2.6.9-smp

ovzkernel-2.6.18
ovzkernel-2.6.18-smp

только i386:

ovzkernel-2.6.18-enterprise

Инструментарий OpenVZ для i386 и amd64

vzctl
vzquota
vzprocps

шаблоны ОС для i386 и amd64 : Debian 3.1 Minimal

vzctl-ostmpl-debian

установка ядра, инструментария и шаблона ОС Debian

Пример: установить стабильное ядро OpenVZ, инструментарий, и шаблон ОС.

 # aptitude install ovzkernel-2.6.9 vzctl vzquota vzctl-ostmpl-debian

Обновите конфигурацию загрузчика. (см. Файл ***/etc/kernel-img.conf***)

Загрузчик "GRUB":

 # /sbin/grub-update

Перезагрузитесь с новым ядром OpenVZ для Debian

 # reboot

Это все! :-)

Далее вы можете переходить к работе с VE.

Etch

OpenVZ теперь часть репозитория Debian Etch.

Установка ядра

Скомпилированные ядра

Ищите на http://download.openvz.org/kernel/debian/etch/

Список компилированных ядер:

linux-image-2.6.18-openvz-486_02_i386.deb
linux-image-2.6.18-openvz-686_02_i386.deb
linux-image-2.6.18-openvz-amd64_01_amd64.deb
linux-image-2.6.18-openvz-ia64_01_ia64.deb
linux-image-2.6.18-openvz-k7_02_i386.deb
linux-image-2.6.18-openvz-sparc64-smp_01_sparc.deb
linux-image-2.6.18-openvz-sparc64_01_sparc.deb

Пример: Установка компилированного ядра OpenVZ для Debian под процессор i686:

 # wget http://download.openvz.org/kernel/debian/etch/linux-image-2.6.18-openvz-686_02_i386.deb
 # dpkg -i linux-image-2.6.18-openvz-686_02_i386.deb

Компиляция собственного ядра («Дао Debian»)

Для установки исходного кода ядра, и патча OpenVZ выполните:

 # apt-get install kernel-package linux-source-2.6.18 kernel-patch-openvz libncurses5-dev

Распакуйте тексты ядра:

 # cd /usr/src
 # tar xjf linux-source-2.6.18.tar.bz2
 # cd linux-source-2.6.18

Вам нужен файл конфигурации ядра. Можно использовать файл конфигурации ядра Debian:

 # cp /boot/config-2.6.18-3-686 .config

Или загрузить файл конфигурации ядра версии 2.6.18 из http://download.openvz.org/kernel/devel/current/configs/

wget http://download.openvz.org/kernel/devel/current/configs/kernel-2.6.18-028test010-i686.config.ovz -O .config

Теперь вы можете модифицировать ядро и изменить конфигурацию:
 # ../kernel-patches/all/apply/openvz
 # make menuconfig

Вам нужны следующие настройки ядра OpenVZ:

(взято из ядра OpenVZ 2.6.18-028test010.1 на процессоре 686)

Filesystem
_ [*] Second extended fs support (CONFIG_EXT2_FS)
_ [*] Ext3 journalling file system support (CONFIG_EXT3_FS)
_ [*] Quota Support (CONFIG_QUOTA)
_ [*] Compatibility with older quotactl interface (CONFIG_QUOTA_COMPAT)
_ [*]Quota format v2 support (CONFIG_QFMT_V2)
_ [*] VPS filesystem (CONFIG_SIM_FS)
_ [*] Virtuozzo Disk Quota support (CONFIG_VZ_QUOTA)
\-> [*] Per-user and per-group quota in Virtuozzo quota partitions (VZ_QUOTA_UGID)

OpenVZ ... (what else :-)
_[*] Virtual Environment support (CONFIG_VE)
_ <M> VE calls interface (CONFIG_VE_CALLS)
_ <M> VE networking (CONFIG_VE_NETDEV)

_ <M> Virtual ethernet device (CONFIG_VE_ETHDEV)
_ <M> VE device (CONFIG_VZ_DEV)
_ [*] VE netfiltering (CONFIG_VE_IPTABLES)
_ <M> VE watchdog module (CONFIG_VZ_WDOG)
_ <M> Checkpointing & restoring Virtual Environments (CONFIG_VZ_CHECKPOINT)

User resources ... (User Beancounters)
_ [*] Enable user resource accounting (CONFIG_USER_RESOURCE)
_ [*] Account physical memory usage (CONFIG_USER_RSS_ACCOUNTING)
_ [*] Account disk IO (CONFIG_UBC_IO_ACCT)
_ [*] Account swap usage (CONFIG_USER_SWAP_ACCOUNTING)
_ [*] Report resource usage in /proc (CONFIG_USER_RESOURCE_PROC)
_ [*] User resources debug features (CONFIG_UBC_DEBUG)
_ [*] Debug kmemsize with cache counters (CONFIG_UBC_DEBUG_KMEM)

INFO: Better to build the kernel-headers as well, so afterward other kernel-modules can
built without whole kernel tree (e.g. drbd -> drbd0.7-module-source)
See also :
 # make-kpkg --targets

Скомпилируйте ядро под администратором системы или с ключом --rootcmd!

 # make-kpkg --append_to_version=-1-openvz --added_patches=openvz --revision=1 --initrd binary-arch

Или все вместе, включая конфигурацию в один шаг

 # make-kpkg --append_to_version=-1-openvz --added_patches=openvz --revision=1 --initrd --config menuconfig binary-arch

Установите ядро и обновите initramfs:

 # dpkg -i ../linux-image-2.6.18-1-openvz_1_i386.deb
 # update-initramfs -c -k 2.6.18-1-openvz
INFO: update-initramfs is done, when make-kpkg is use with --initrd option
INFO: update-grub can be configured by /etc/kernel-img.conf

Обновите загрузчик, если вы не сделали этого ранее GRUB :

 # /usr/sbin/update-grub
INFO: since the Debian ETCH-release the location of update-grub is moved from /sbin/update-grub to /usr/sbin/update-grub !

Установка инструментария

 # apt-get install vzctl vzquota

Изменение настроек

Дао Debian:

Для того, чтоб виртуальные сервера имели доступ к сети, вам необходимо разрешить ip-перенаправление. Установите параметр "***ip_forward***" в ***yes*** в файле ***/etc/network/option***.

 # editor /etc/network/options

В некоторых случаях вам будет необходимо разрешить proxy_arp для сетевых устройств, которые вы планируете использовать из виртуальных серверов. Вы можете добавить их к определенному сетевому интерфейсу в конфигурации сети (***/etc/network/interfaces***) заменив строки аналогичными нижеописанным, заменяя маску %DEV% на имя вашего устройства (например eth0).

Пример

```
[...]
# device: %DEV%
iface %DEV% inet static
      address 192.168.0.2
      netmask 255.255.255.0
      network 192.168.2.0
      broadcast 192.168.2.255
      gateway 192.168.2.1

      up sysctl -w net.ipv4.conf.%DEV%.proxy_arp=0
      pre-down sysctl -w net.ipv4.conf.%DEV%.proxy_arp=1
 [...]
```

Или используйте каталоги ***/etc/network/if-up/*** и ***/etc/network/if-down.d/***.

Разрешите клавищу ***magic-sysrq*** в файле ***/etc/sysctl.conf***

«Дао OpenVZ Linux »

Добавьте в "***/etc/sysctl.conf***" следующие настройки

```
# On Hardware Node we generally need
# packet forwarding enabled and proxy arp disabled
net.ipv4.ip_forward = 1
net.ipv4.conf.default.proxy_arp = 0

# Enables source route verification
net.ipv4.conf.all.rp_filter = 1

# Enables the magic-sysrq key
kernel.sysrq = 1

# TCP Explict Congestion Notification
# net.ipv4.tcp_ecn = 0

# we do not want all our interfaces to send redirects
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
```

Совет: сделайте символическую ссылку с ***/var/lib/vz*** в ***/vz*** для совместимости с OpenVZ на RPM-машинах.
(на Debian корень раздела vz установлен в ***/var/lib/vz***)

```
# ln -s /var/lib/vz /vz
```

Перед перезагрузкой сервера, проверьте, что все необходимые для функционирования вашей системы модули разрешены; Возможно вам потребуется INITRD (initramdisk) или статически прикомпилированные к ядру модули.

```
# reboot
```

И это все!

3. Установка OpenVZ на новый компьютер на примере CentOS 4 Server.

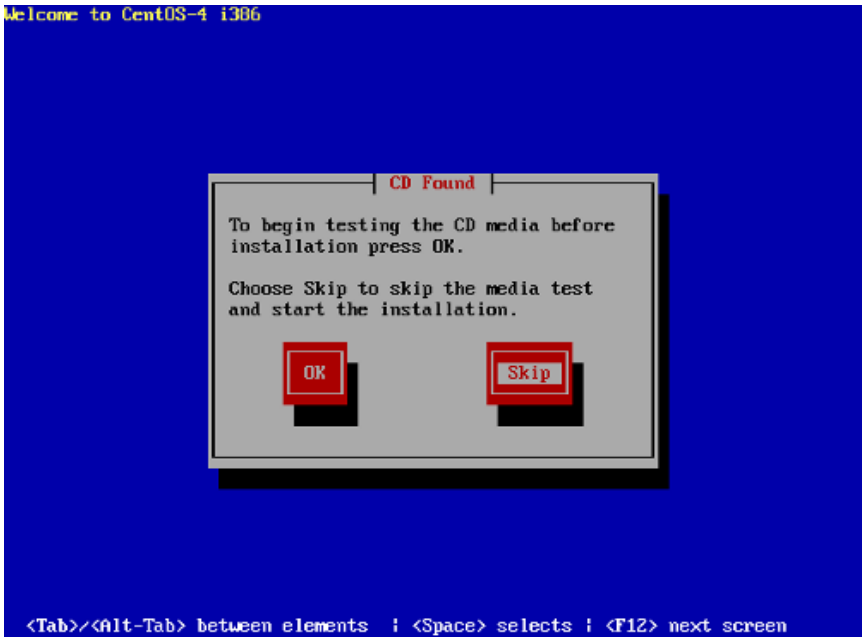
3.1. Предварительные требования.

- Оборудование
  Процессор Pentium, 256 MB ОЗУ, 6 Гб жесткий диск
- ПО
  CentOS
- Другое
  Сетевая карта и доступ к Интернету.

3.2 Загрузка с установочного CD-ROM.

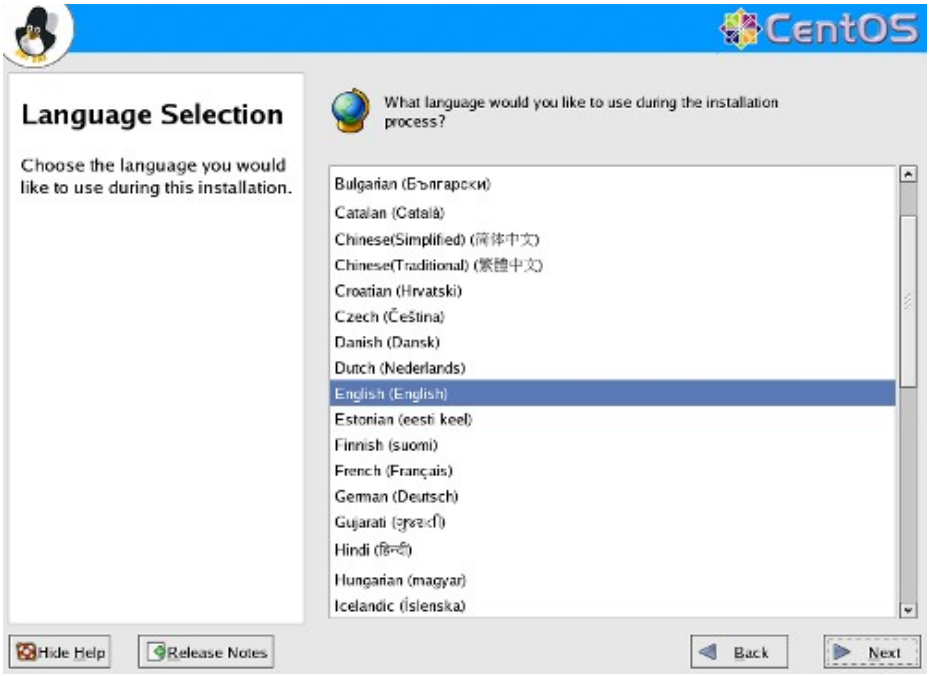
Загрузитесь с первого CD установочного комплекта CentOS 4.4. В случае если это DVD-версия, или CentOS 4.4 Server CD – это единственный диск в комплекте.

После загрузки система попросит у вас проверить качество носителя. Так как это достаточно долгий процесс, мы опустим его:

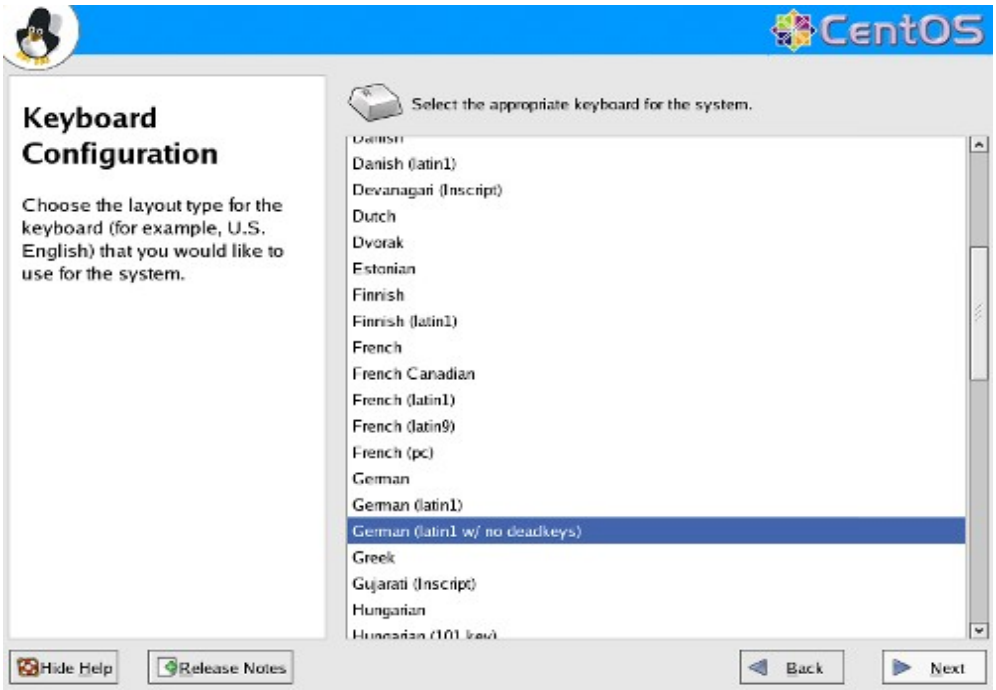


Вы увидите приветствие программы установки CentOS. Нажмите кнопку «Next»:

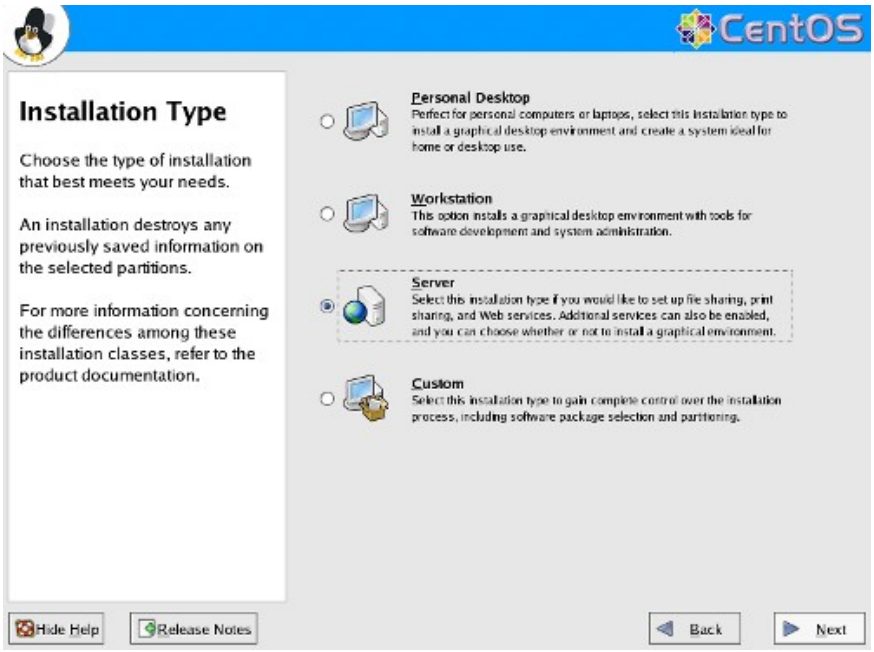
Выберите язык системы. Я настоятельно советую вам выбрать свой родной язык:



Выберите раскладку клавиатуры:

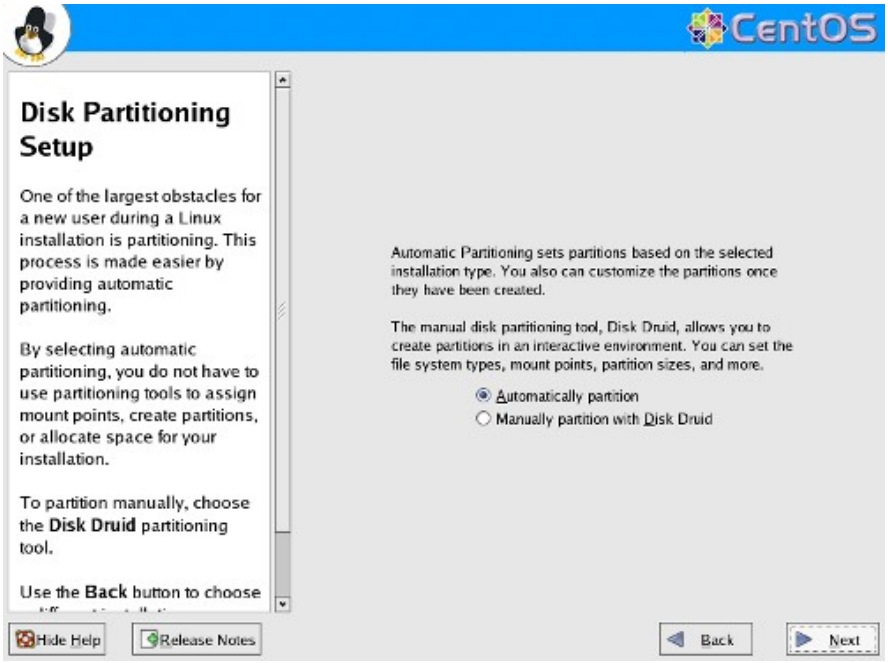


На следующем этапе вам будет предложено выбрать тип установки. Так как мы создаем аппаратный узел OpenVZ для виртуальных серверов – выбирайте “Server”:

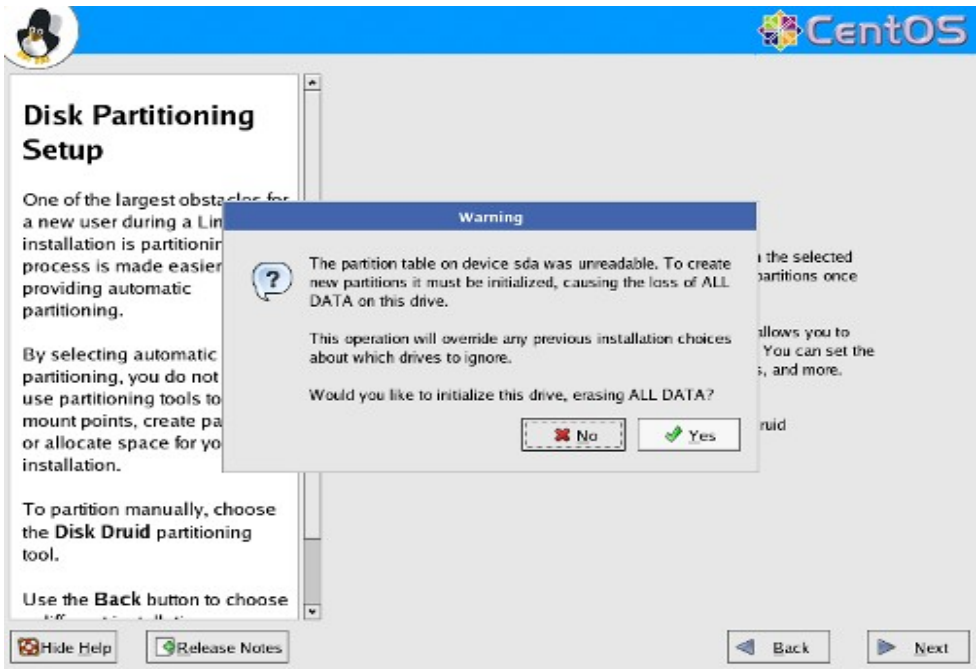


3.3 Разметка жесткого диска.

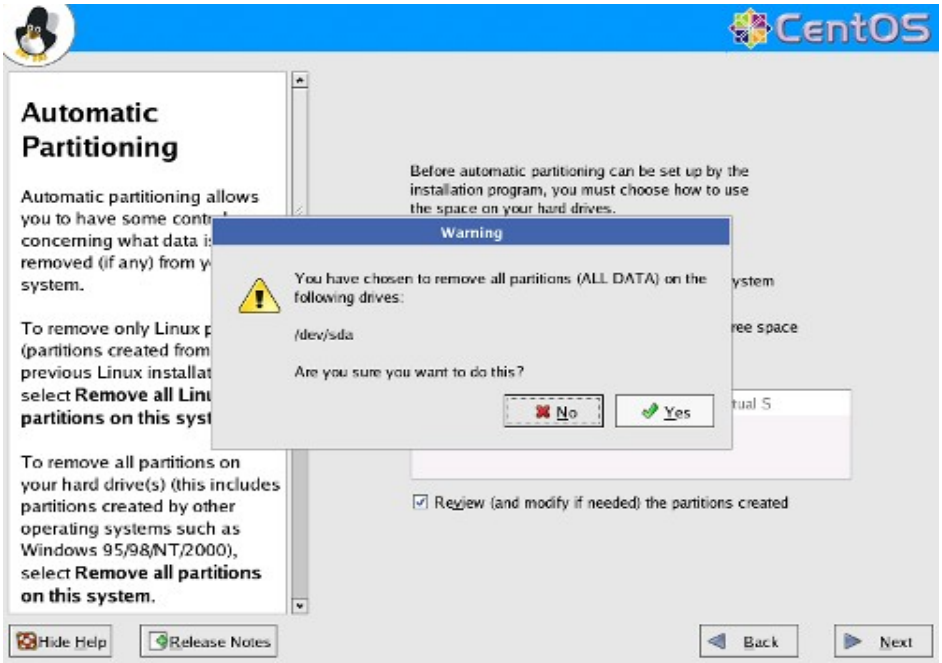
На этапе разбиения диска выберите «ручное разбиение с помощью DiskDruid». Это связано с особенностями хранения данных виртуальных серверов под управлением OpenVZ.



Так как мы устанавливаем систему на новый компьютер, мы ответим «Да» на предупреждении DiskDruid о риске потери всех данных.

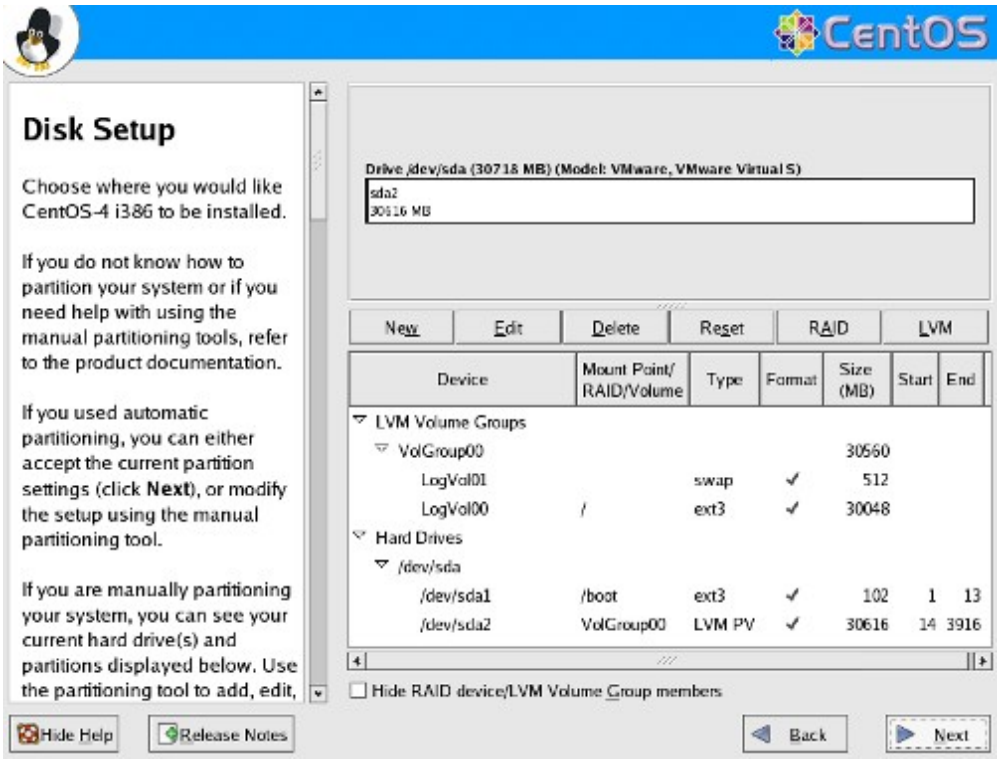


Выберите «Удалить все разделы»



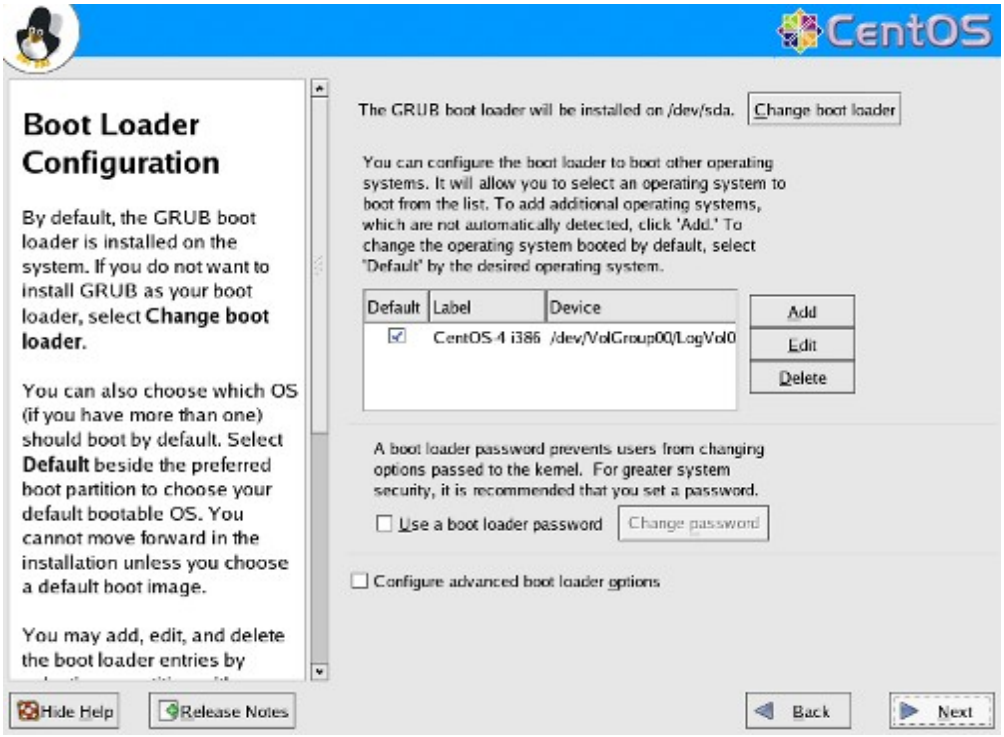
Кроме того, подтверждаем удаление всех Linux-разделов.
Создайте диск со следующей разбивкой на аппаратном узле:

Точка монтирования	**Тип файловой системы**	**Комментарии**
/	ext3	Корневой раздел для файлов ОС аппаратного узла. 4-6 Гб
/swap	swap	Пространство подкачки Linux. 2 x ОЗУ
/vz	ext3	Корневой раздел для файлов VPS. Все свободное место.

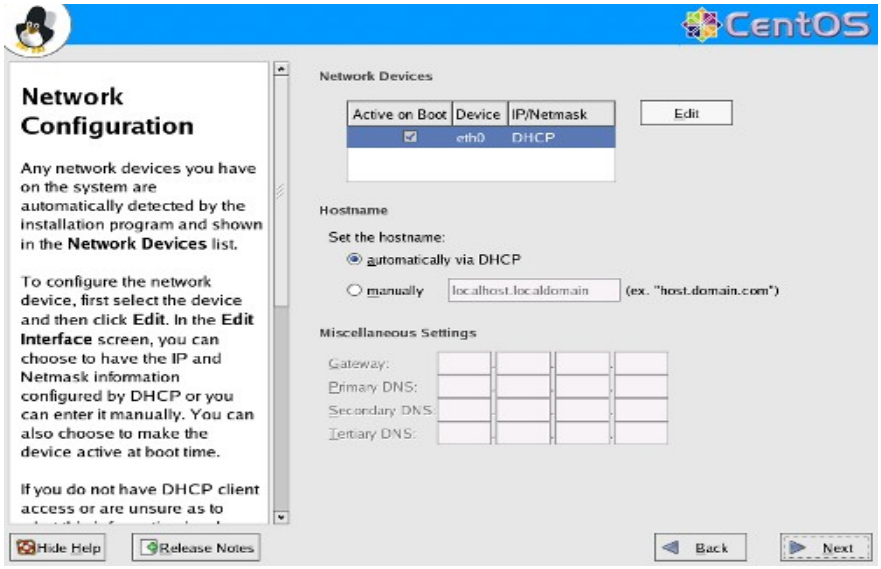


3.4 Настройка загрузчика, сети, и безопасности.

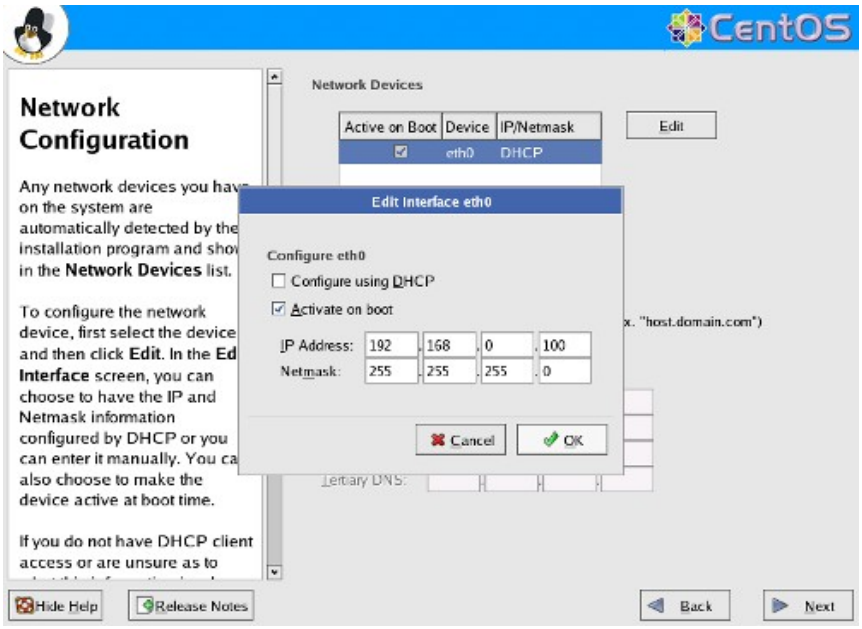
Настройки загрузчика GRUB можно оставить по умолчанию:

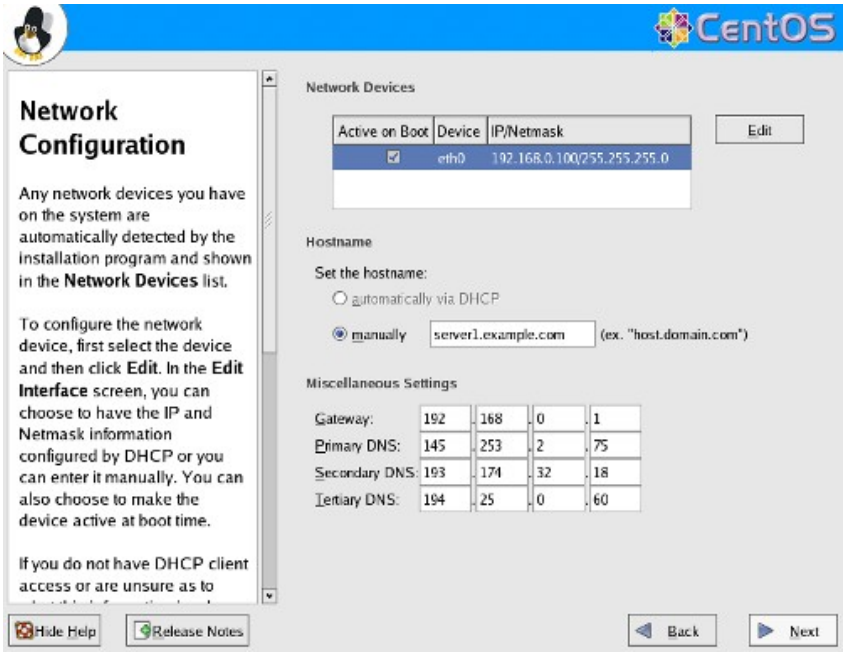


Настройки сети зависят от того, каким образом в ней выделяются адреса IP. Но так как мы создаем сервер – присвоение статического IP-адреса не является грубой ошибкой. Нажмите кнопку «Изменить» в парвом верхнем углу. В появившемся диалоге снимите «Использовать DHCP» и присвойте сетевой карте статический IP-адрес. В этом руководстве это будет адрес 192.168.0.100

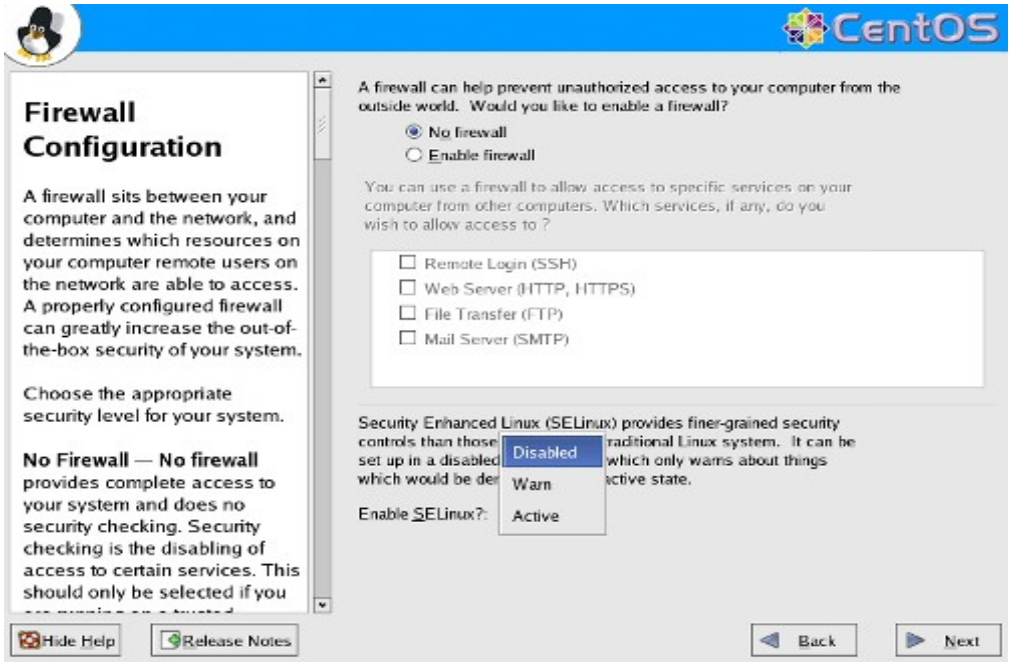


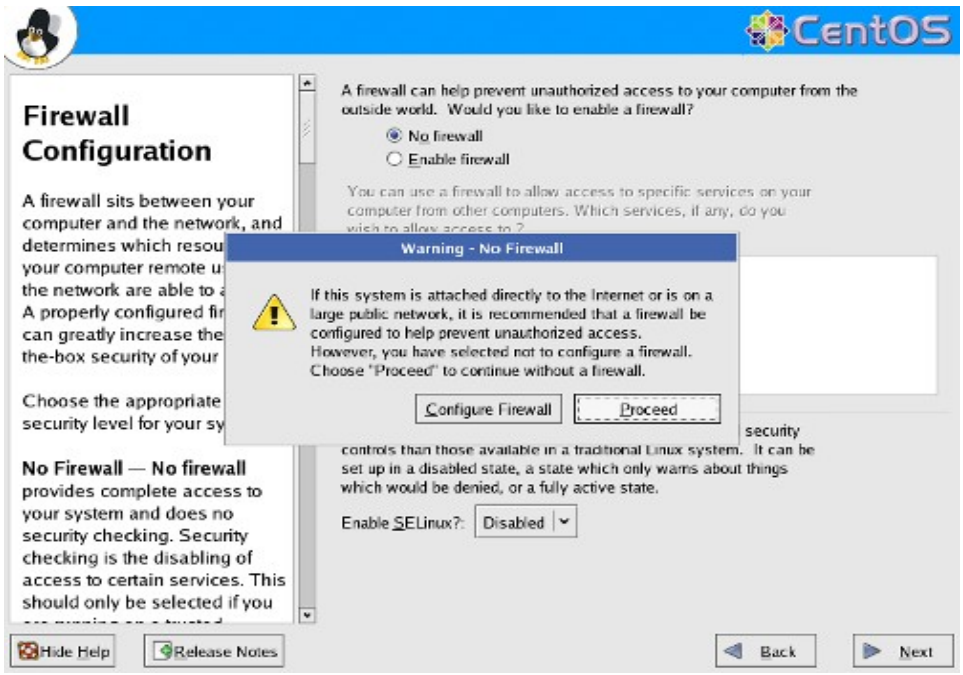
Установите имя узла, шлюз по умолчанию, и введите до 3 DNS-серверов.



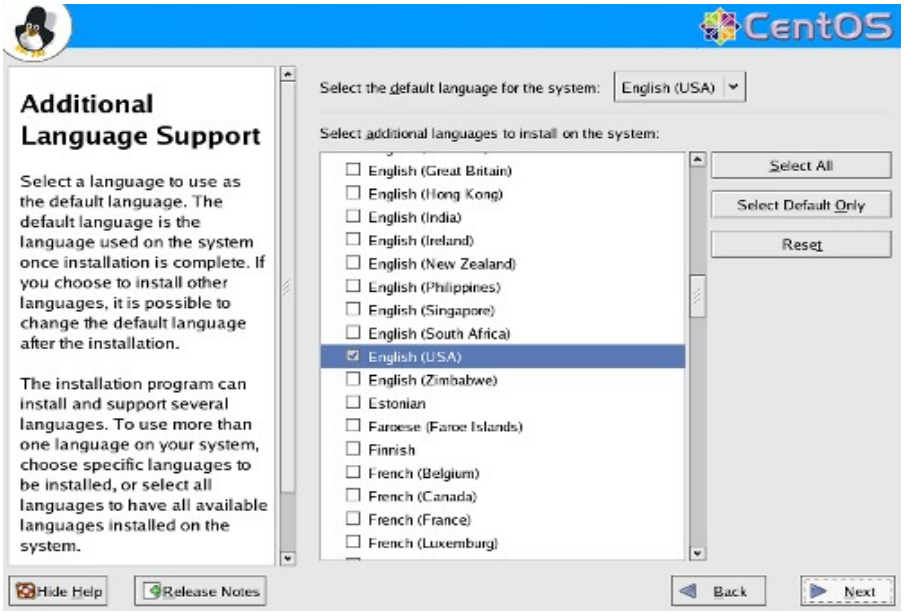


SELinux - это расширение безопасности CentOS. Советую отключить ее, ибо в данном случае от нее больше проблем чем выгод. Файрволл тоже рекомендуется отключить.

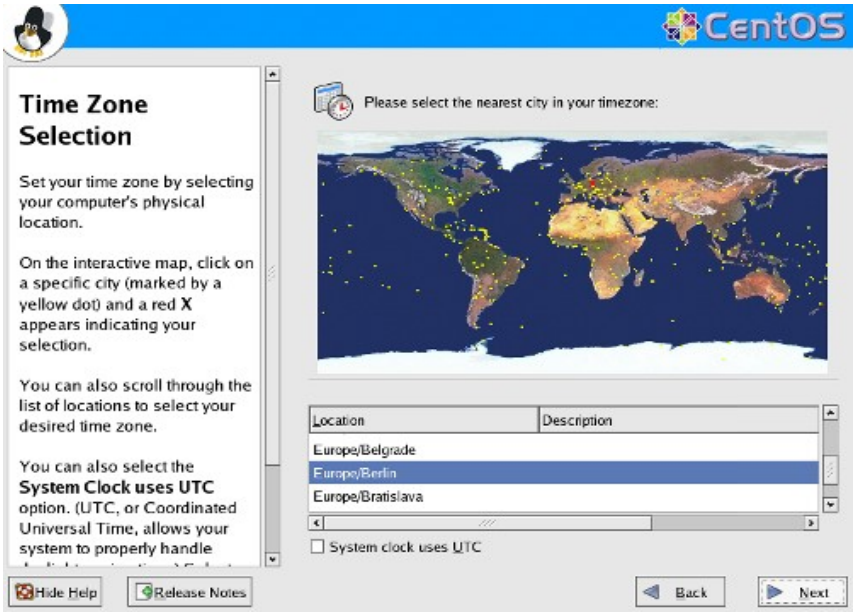




Выберите язык системы и добавьте те языки, которые необходимо использовать:



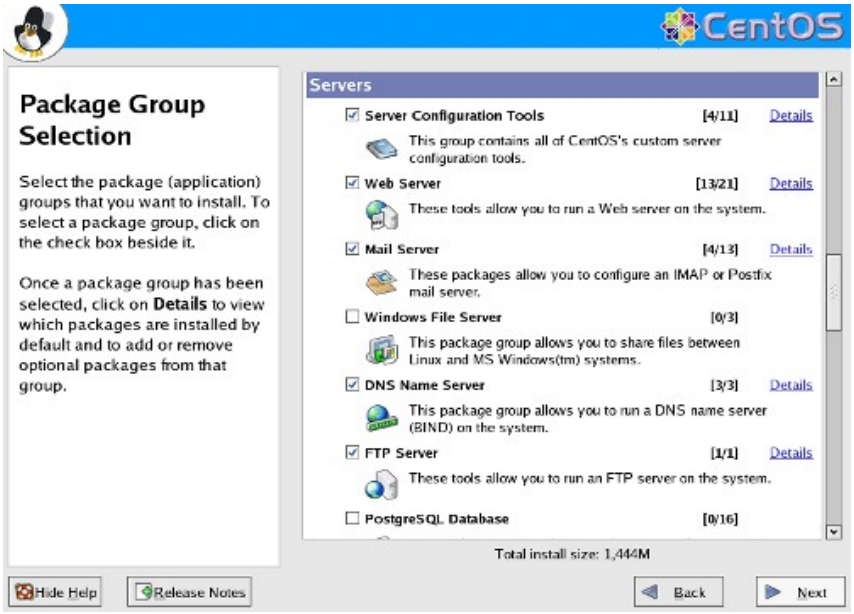
Выберите часовой пояс:



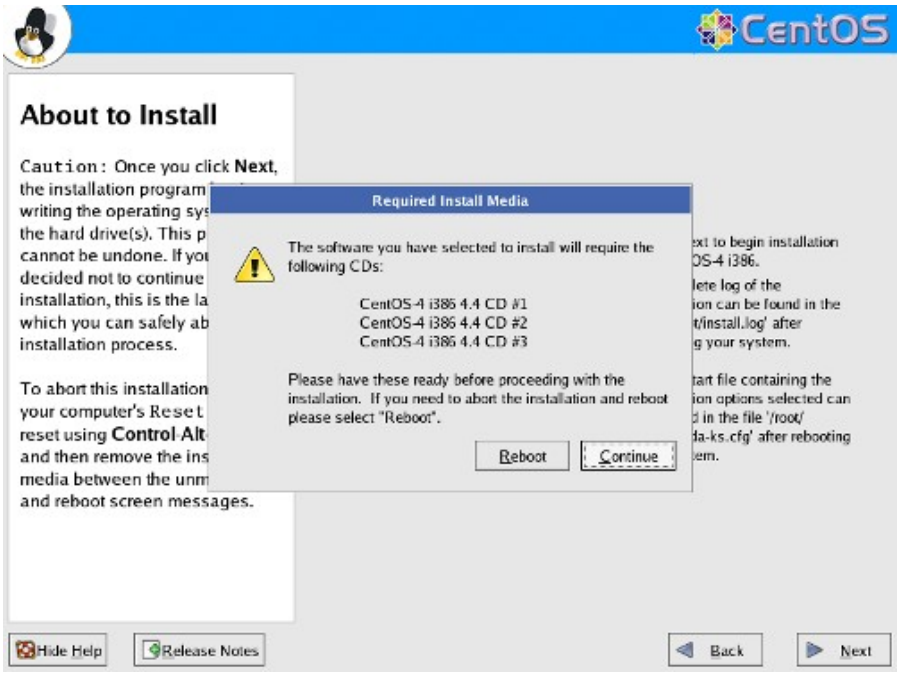
Установите пароль администратора:

3.5 Выбор пакетов. Безопасность аппаратного узла OpenVZ.

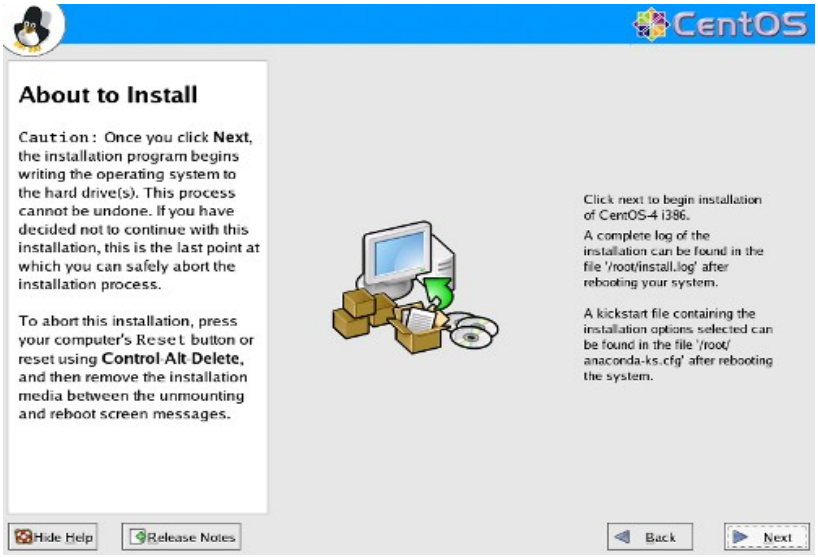
В руководстве безопасности OpenVZ настоятельно рекомендуется не использовать на уровне аппаратного узла никаких сетевых сервисов кроме SSHD. Практически, для управления аппаратным узлом вам может потребоваться более широкий набор сетевых служб. Если вы уверены, что кроме SSH на устанавливаемом аппаратном узле ничего использоваться не будет – снимите флажок «сервер». Вы получите «голую» систему без каких-либо приложений.



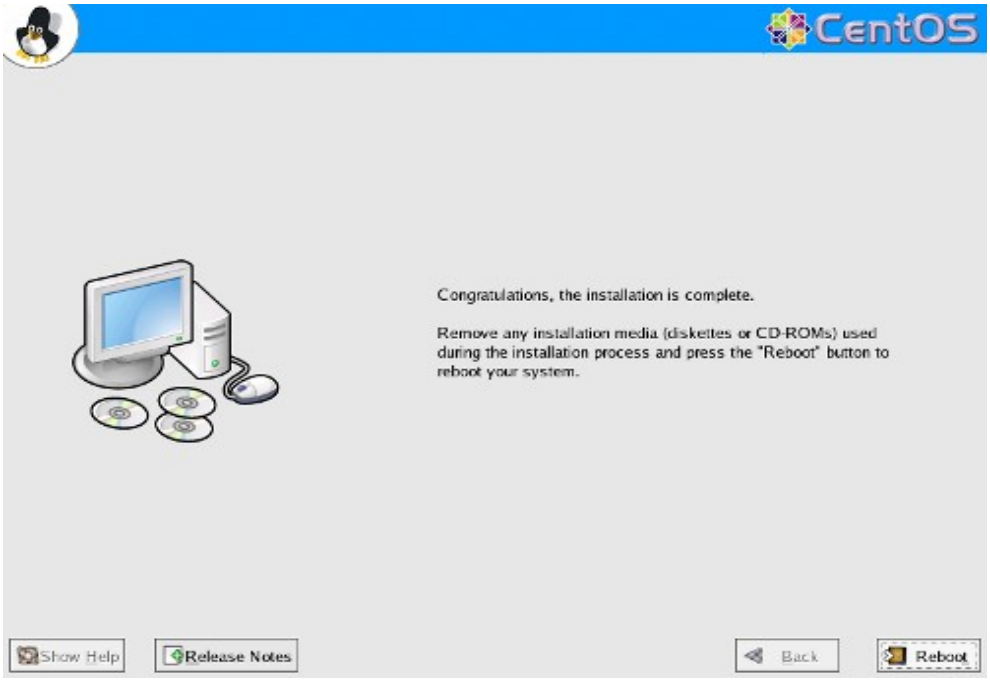
При необходимости инсталлятор попросит вас заменить диск:



После этого начинается перенос системы на жесткий диск:



После установки выньте CD и перезагрузите компьютер



после перезагрузки войдите в систему под администратором и выполните команды

yum update -y
shutdown now -r

3.6 Установка ядра OpenVZ.

Войдите в систему и выполните команды:

 # cd /etc/yum.repos.d
 # wget http://download.openvz.org/openvz.repo
 # yum install ovzkernel

Добавьте в файл /boot/grub/grub.conf file строки:

```
title Cent OS (2.6.8-022stab029.1)
      root (hd0,0)
      kernel /vmlinuz-2.6.8-022stab029.1 ro root=/dev/sda5 quiet rhgb vga=0x31B
      initrd /initrd-2.6.8-022stab029.1.img
```

Смените заголовок на “OpenVZ”. Удалите параметры командной строки ядра, оставив только «root=...». Должно получиться что-то похожее:

```
title OpenVZ (2.6.8-022stab029.1)
      root (hd0,0)
      kernel /vmlinuz-2.6.8-022stab029.1 ro root=/dev/sda5
      initrd /initrd-2.6.8-022stab029.1.img
```

Для корректной работы ядра OpenVZ необходимо настроить некоторые парметры. Эти параметры находятся в файле /etc/sysctl.conf.

```
# Включаем перенаправление пакетов
net.ipv4.ip_forward = 1
net.ipv4.conf.default.proxy_arp = 0
# Проверка источника включена
net.ipv4.conf.all.rp_filter = 1
# разрешаем клавишу SysRq
kernel.sysrq = 1
# Явное согласование TCP
#net.ipv4.tcp_ecn = 0
# нам не нужно чтобы ВСЕ интерфейсы перенаправляли трафик
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
```

Настройка SELinux

Мы отключаем SELinux. Впишите в файл /etc/sysconfig/selinux:

```
SELINUX=disabled
```

Отслеживание соединений.

В стабильных ядрах OpenVZ (например 2.6.8-based) слежение соединений в netfilter для VE0 (аппаратный узел) по умолчанию запрещено. Если у вас установлен stateful файрволл на аппаратном узле (установка по умолчанию) вы должны или запретить его, или разрешить отслеживание соединений для VE0.

Для отслеживания соединений в VE0, добавьте в /etc/modprobe.conf file:

options ip_conntrack ip_conntrack_enable_ve0=1

Примечание: в ядрах позднее чем 2.6.8, отслеживание соединений разрешено по умолчанию.

Перезагрузка с ядром OpenVZ

Перезагрузите компьютер и выберите "OpenVZ" в загрузочном меню. Если установка ядра OpenVZ прошла успешно, мы переходим к установке утилит управления OpenVZ.

3.7 Установка утилит OpenVZ.

OpenVZ необходимо некоторое количество утилит, как то:

vzctl – управляет жизненным циклом OpenVZ VPS (создает, удаляет, запускает, останавливает, перегружает, задает параметры, и т.п.)

vzquota – управляет квотами ресурсов для VPS. Чаще вызывается неявно, через vzctl.

Для установки выполните:

 # yum install vzctl vzquota

Запуск OpenVZ

Выполните команду

 # /sbin/service vz start

Примечание: OpenVZ готово к работе на вашем компьютере. Для загрузки ядра OpenVZ по умолчанию, отредактируйте строку «default» в /boot/grub/grub.conf так, чтоб она указывала на ядро OpenVZ. Например, если описание ядра OpenVZ первое в этом файле, напишите **default 0**. Более подробно читайте **man grub.conf**.

3.8 Подготовка кэша шаблонов ОС

Установка утилит управления шаблонами.

Выполните команду

 # yum install vzpkg vzyum vzrpm43-python vzrpm44-python

Установка метаданных шаблонов

Выполните команды:

 # yum search vztmpl
 # yum install vztmpl-XXX [...]

Например

yum install vztmpl-centos-4.i386

Установка кэша репозитория

vzpkgcache

Альтернатива: использование готового шаблона

Перейдите в каталог

cd /vz/template/cache

Загрузите шаблоны с http://download.openvz.org/template/precreated/

3.9 Проверка функционала OpenVZ.

Создание VPS

[host-node]# vzctl create 112 --ostemplate centos-4-i386-default

Добавление IP-адреса

[host-node]# vzctl set 112 --ipadd 192.168.6.112 --save

Запуск VPS

[host-node]# vzctl start 112

Запуск команды с аппаратного узла

[host-node]# vzctl exec VEID ps ax

Вход в VPS

[host-node]# vzctl enter VEID
entered into VPS VEID
[ve]#

Выход из VPS

[ve]# exit
exited from VPS VEID
[host-node]#

Остановка VPS

[host-node]# vzctl stop VEID
Stopping VPS ...
VPS was stopped
VPS is unmounted

Уничтожение VPS

[host-node]# vzctl destroy VEID
Destroying VPS private area: /vz/private/VEID
VPS private area was destroyed

Убедитесь что вы устанавливаете верную версию ядра

Если вы устанавливаете ядро на многоядерный (или hyperthreading) процессор, убедитесь что вы загружаете SMP-версию ядра (ovzkernel-smp) и если вы имеете более 4Гб установленного ОЗУ, убедитесь что ядро его поддерживает (ovzkernel-entnosplit)

4. Установка OpenVZ на новый компьютер на примере Ubuntu 6.10.

В этой главе описывается установка Ubuntu Linux 6.10 Edgy Eft в качестве аппаратного узла OpenVZ. Будет рассмотрена установка и настройка следующего ПО:

- Веб-сервер Apache 2.0
- Сервер СУБД MySQL 5.0
- Ядро OpenVZ

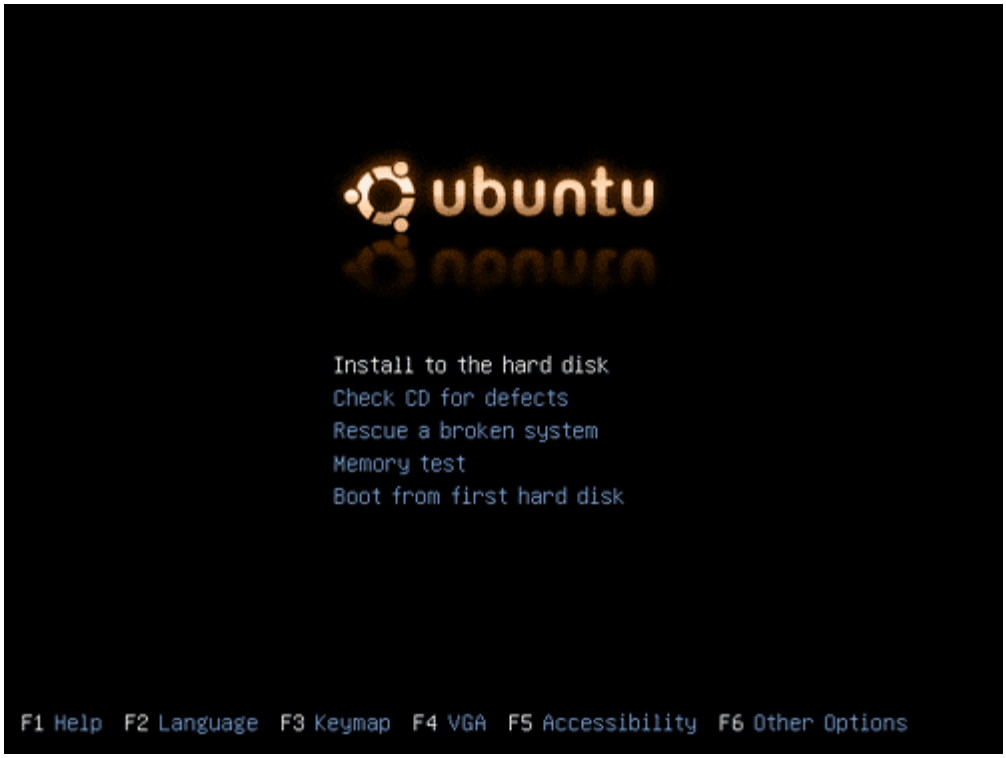
Требования

Вам понадобятся

- Установочный диск Ubuntu 6.10 Server (см. Список зеркал: http://www.ubuntu.com/download - руководство использует образ: http://ftp.cw.net/pub/linux/ftp.ubuntu.com/releases/6.10/ubuntu-6.10-server-i386.iso)
- Соединение с Интернетом.

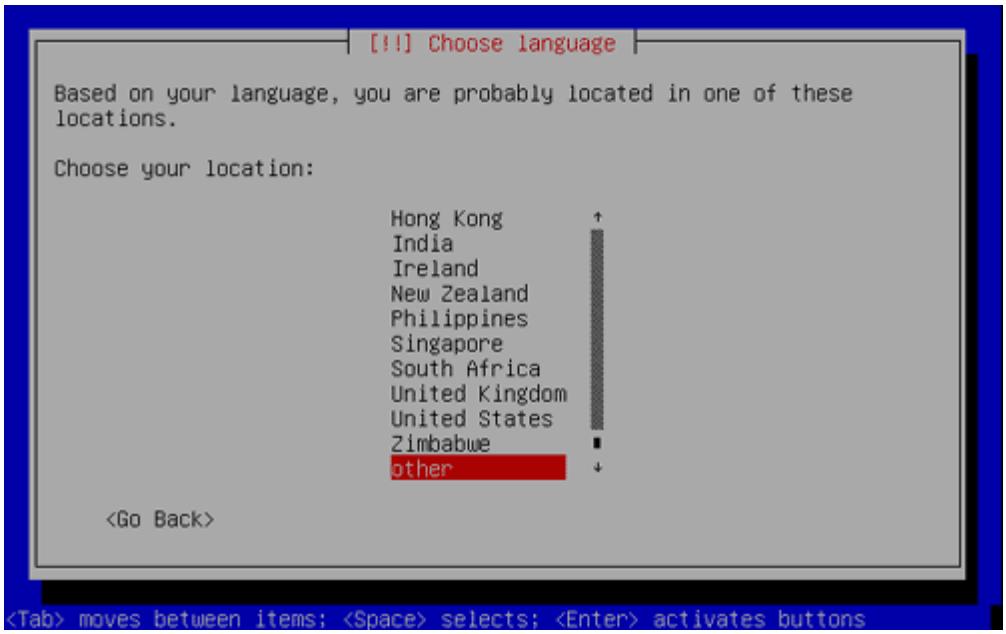
1. Базовая система

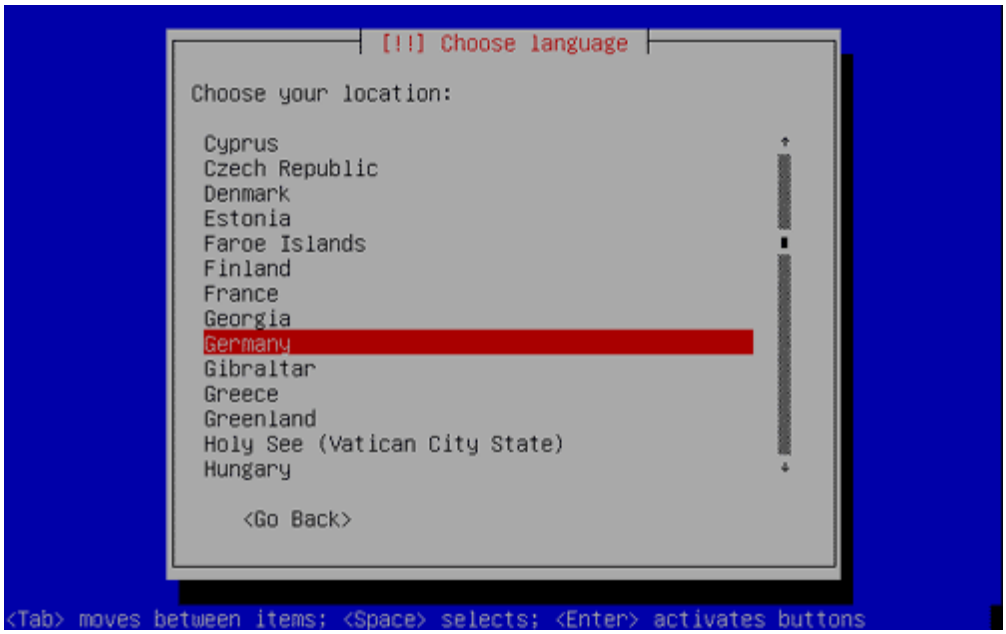
Вставьте установочный диск Ubuntu и загрузитесь с него. Выберите пункт меню «Install to the hard disk»:



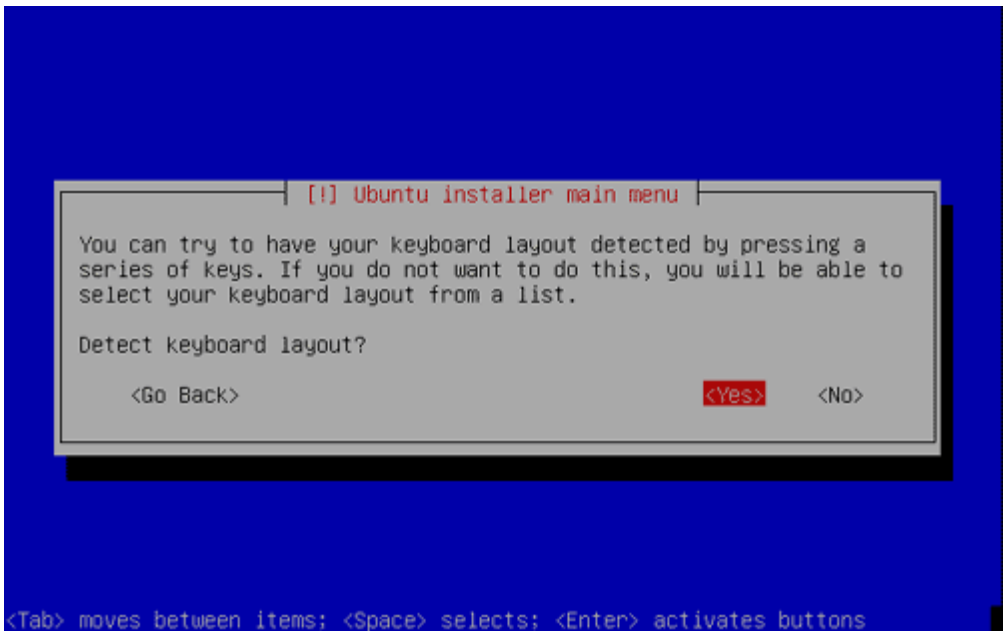
Выберите язык системы

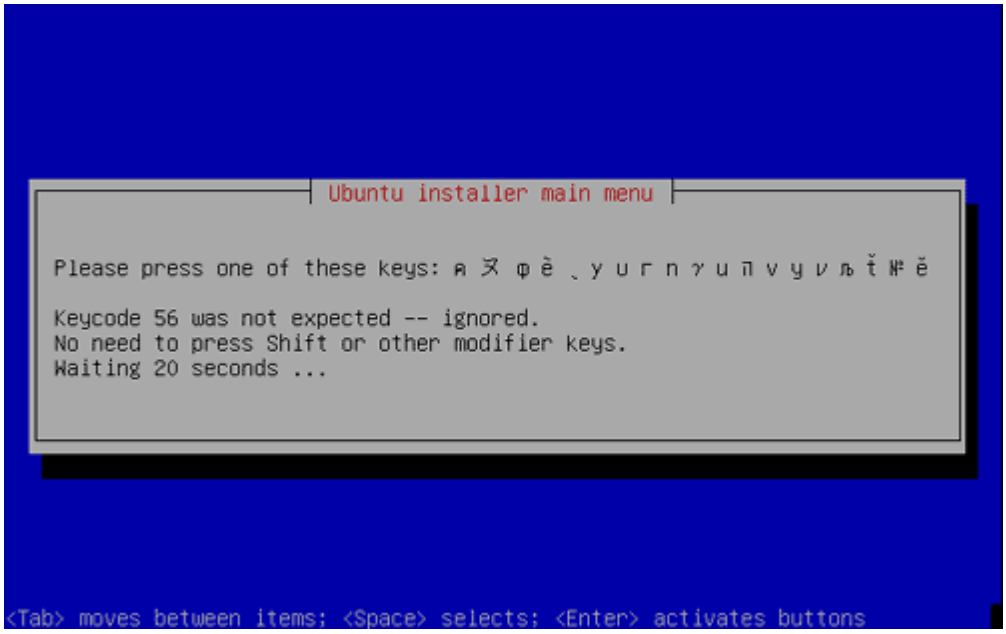
Выберите расположение:

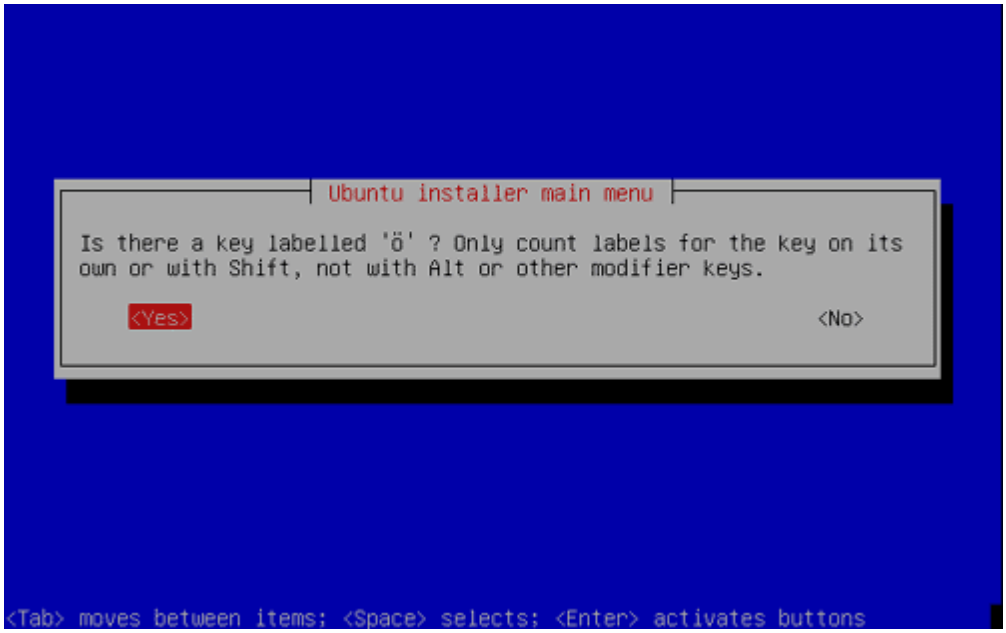


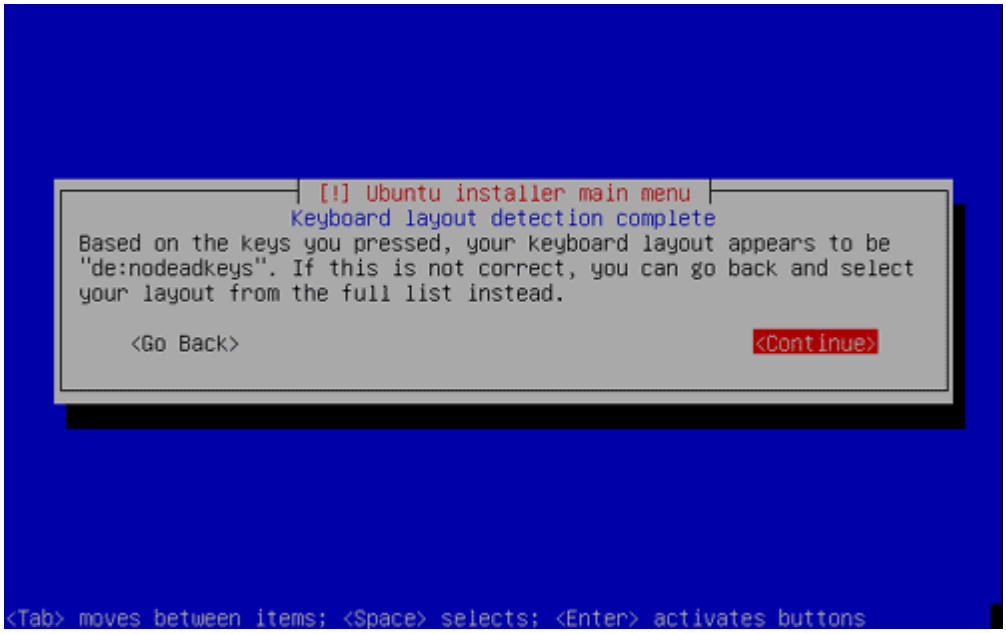


Выберите раскладку клавиатуры

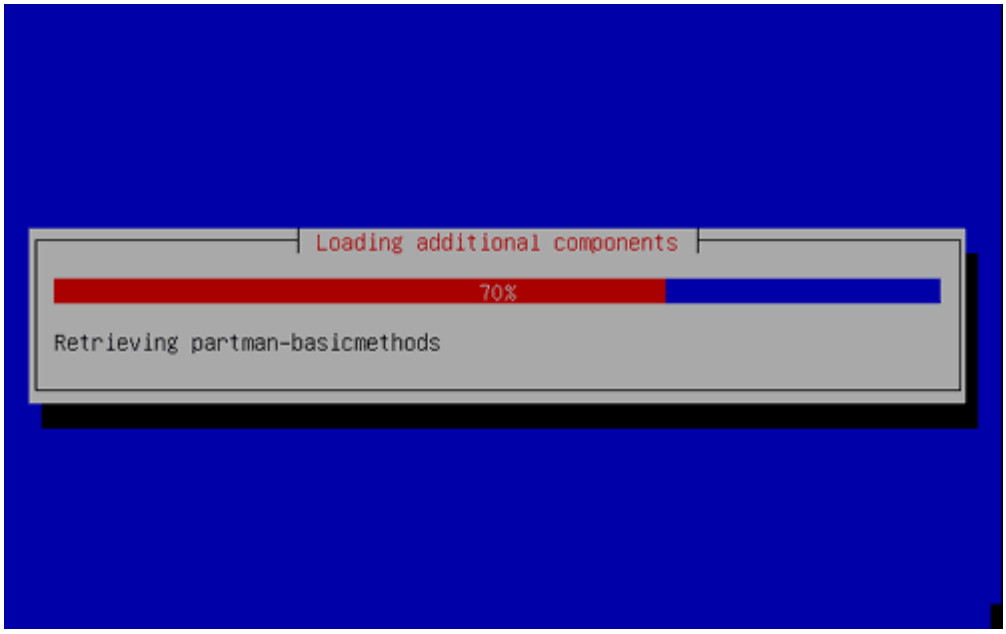


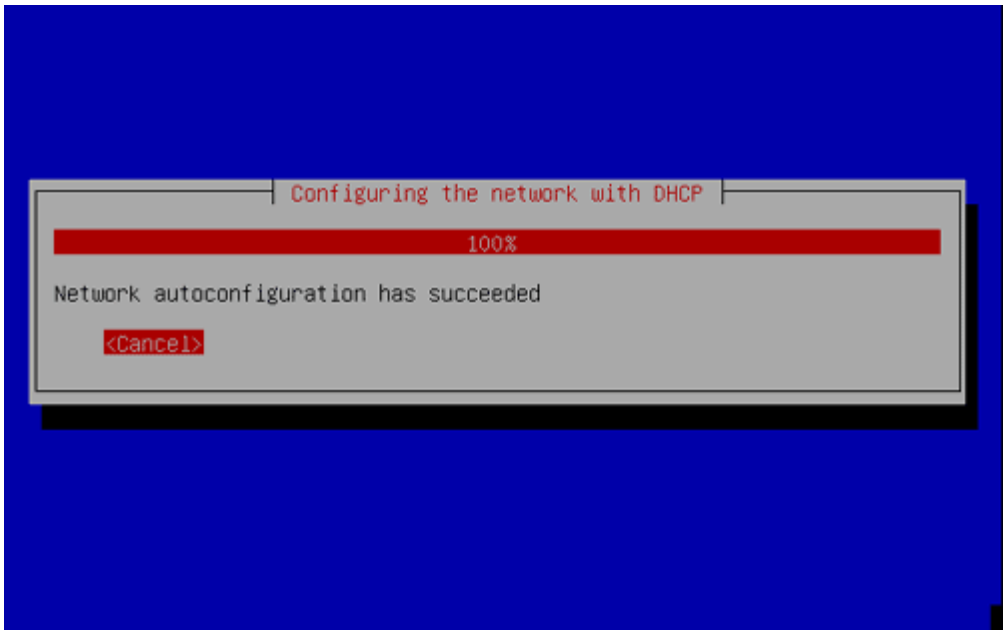



Ubuntu installer main menu
Is there a key labelled 'ö' ? Only count labels for the key on its own or with Shift, not with Alt or other modifier keys.
<Yes>
<No>
<Tab> moves between items; <Space> selects; <Enter> activates buttons

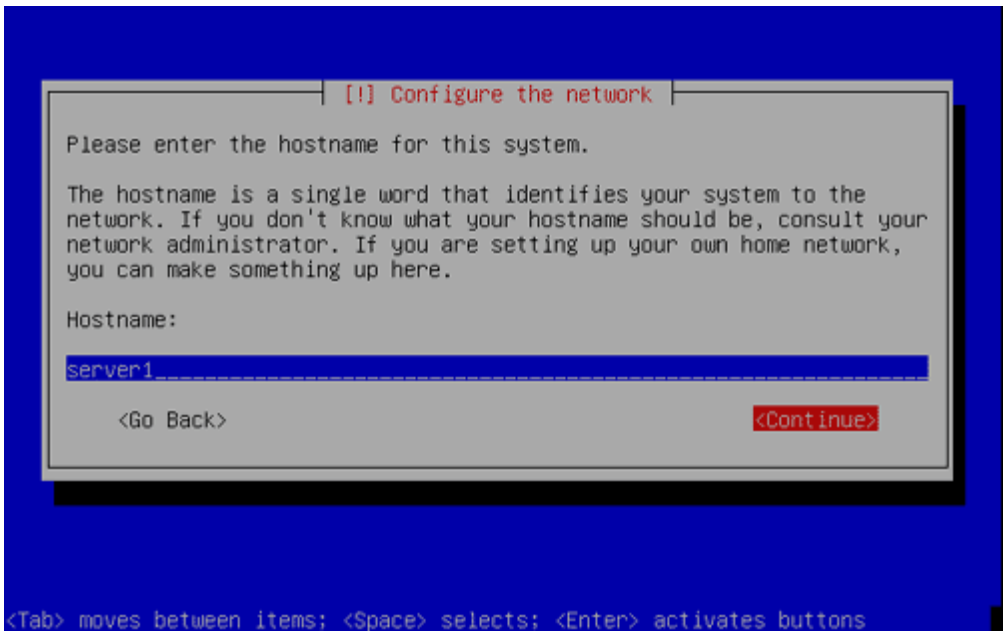

[!] Ubuntu installer main menu
Keyboard layout detection complete
Based on the keys you pressed, your keyboard layout appears to be "de:nodeadkeys". If this is not correct, you can go back and select your layout from the full list instead.
<Go Back>
<Continue>
<Tab> moves between items; <Space> selects; <Enter> activates buttons

Инсталлятор проверит диск, оборудование и настроит сеть, если в ней имеется DHCP сервер.

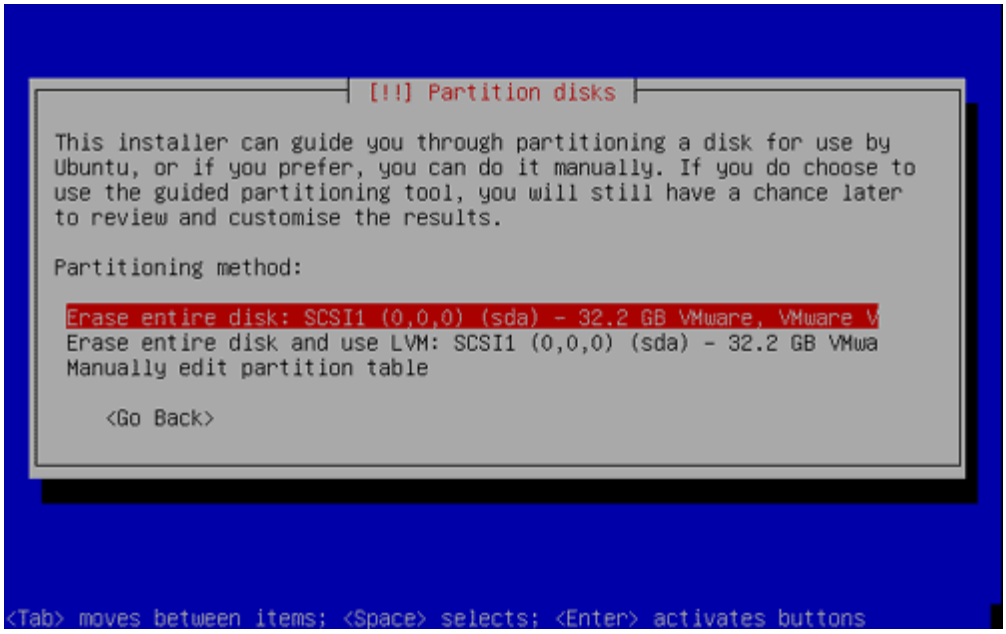


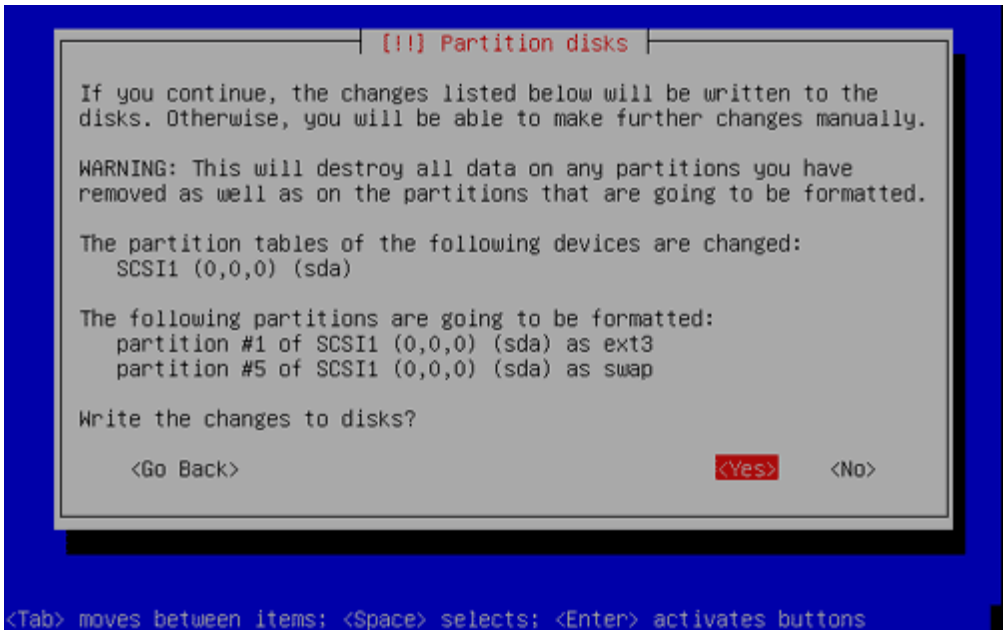


Введите сетевое имя сервера, например *server1*:

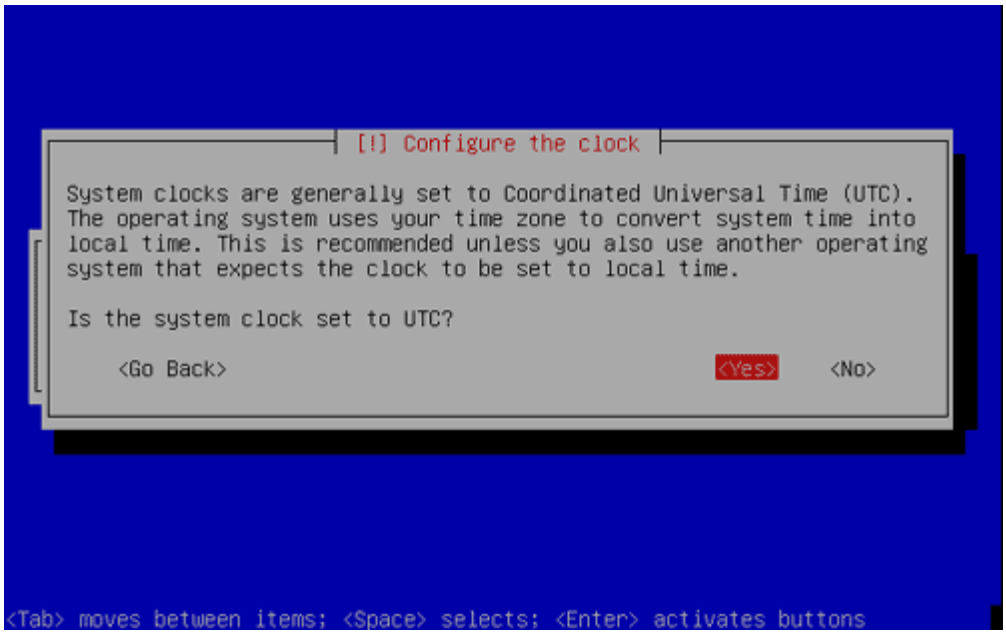


Теперь разбейте диск на большой корневой раздел / и маленький раздел подкачки, стирая весь диск (мы устанавливаем ОС на новую систему)

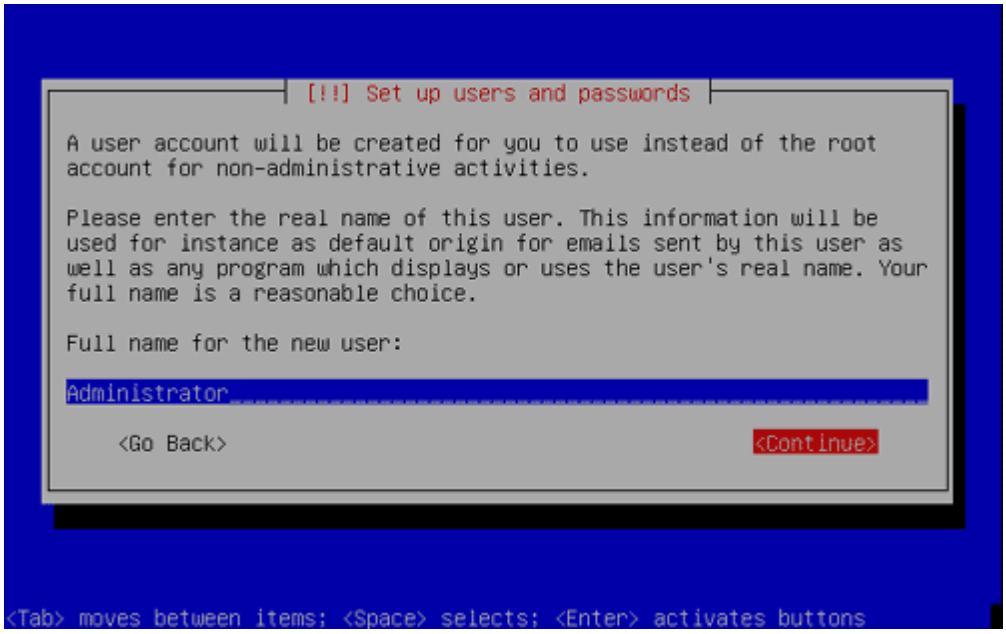


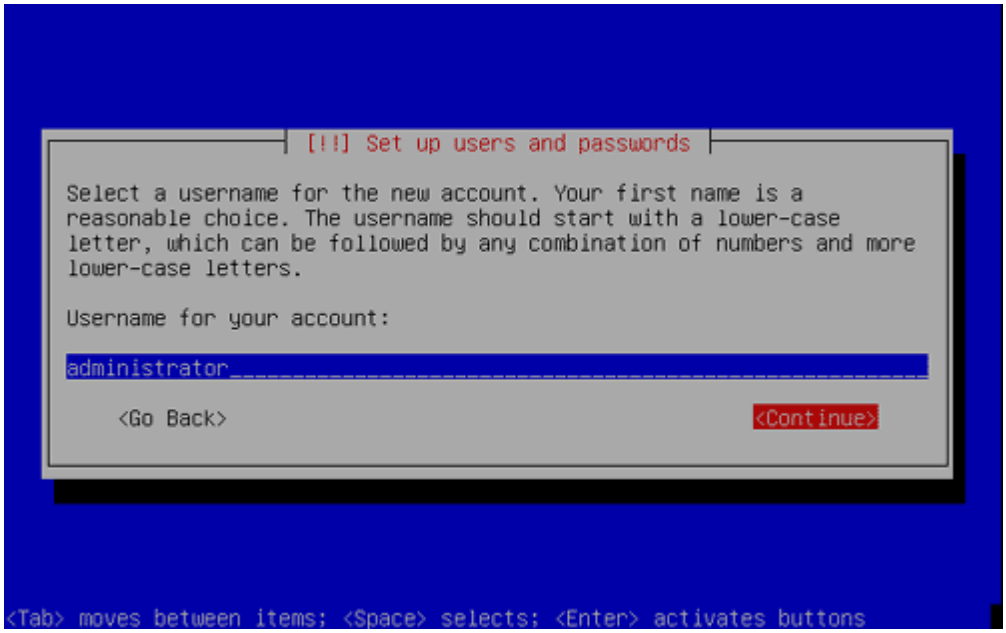


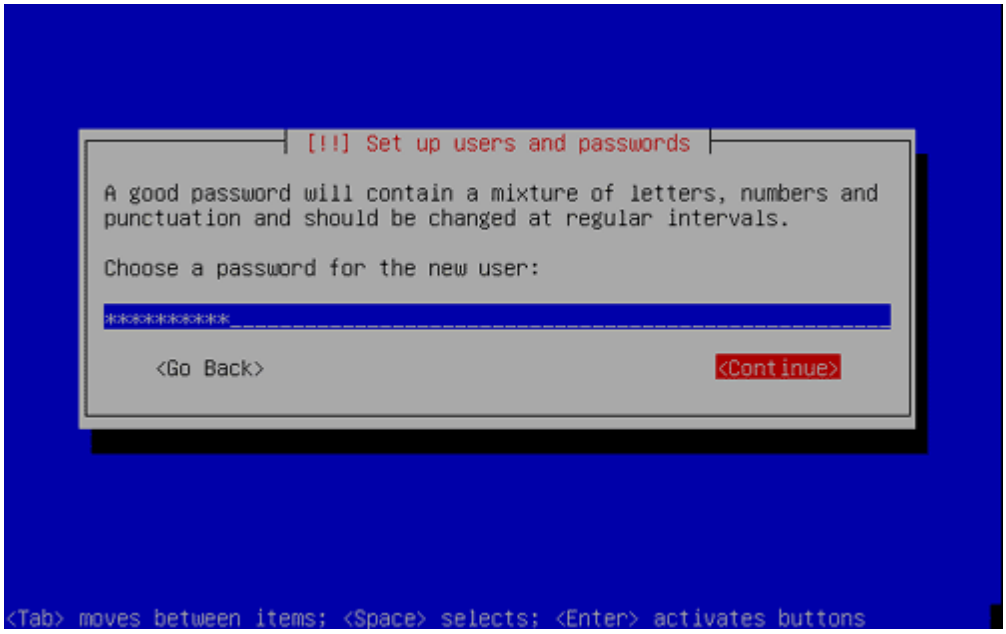
Настройте системные часы:

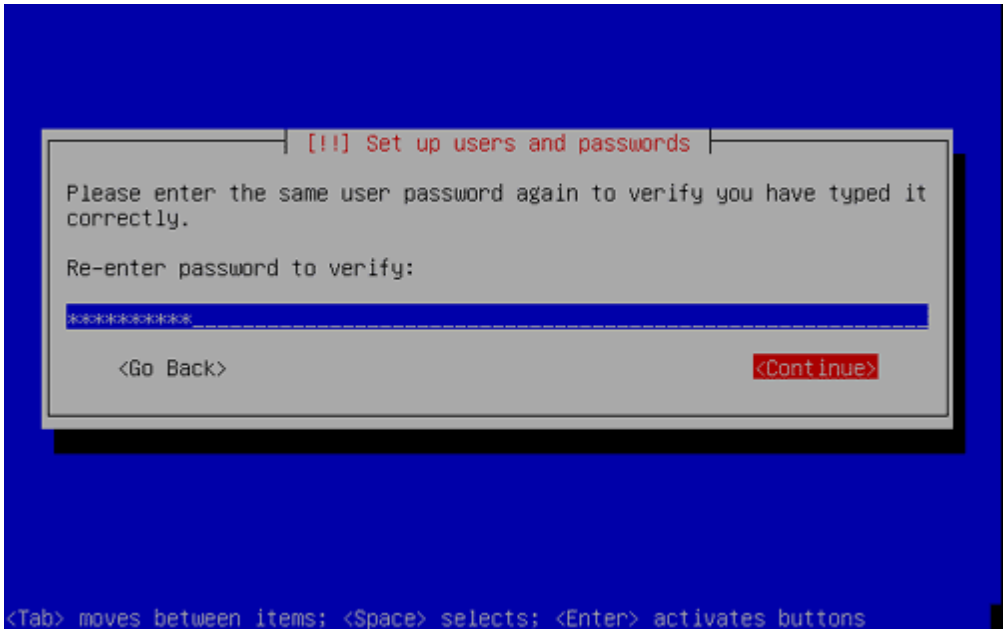


Создайте пользователя Administrator с учетной записью administrator (не используйте admin – это зарезервированное слово):

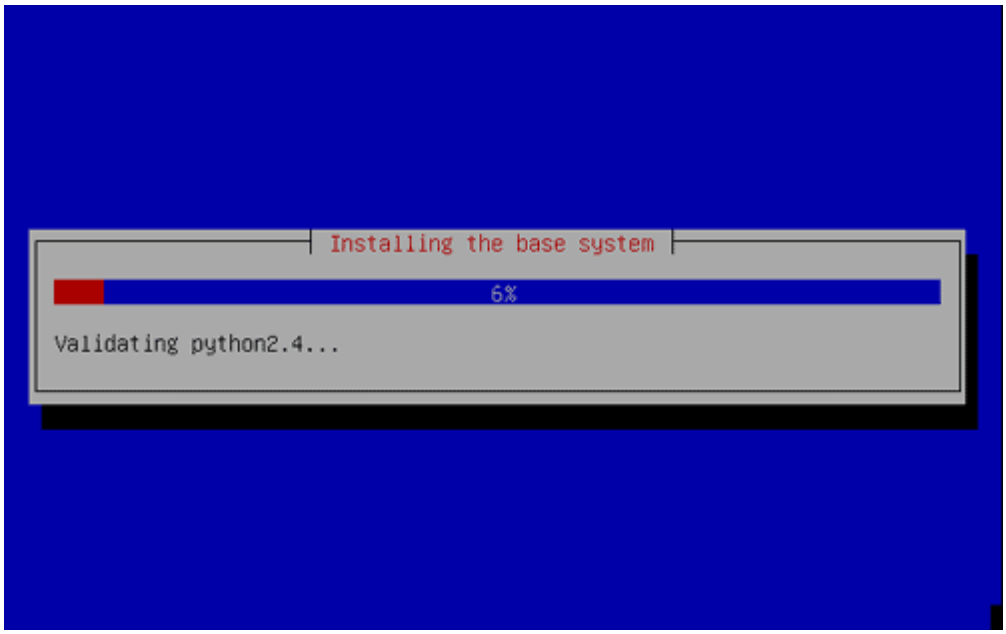



[!!] Set up users and passwords
Select a username for the new account. Your first name is a reasonable choice. The username should start with a lower-case letter, which can be followed by any combination of numbers and more lower-case letters.
Username for your account:
administrator
<Go Back>
<Continue>
<Tab> moves between items; <Space> selects; <Enter> activates buttons

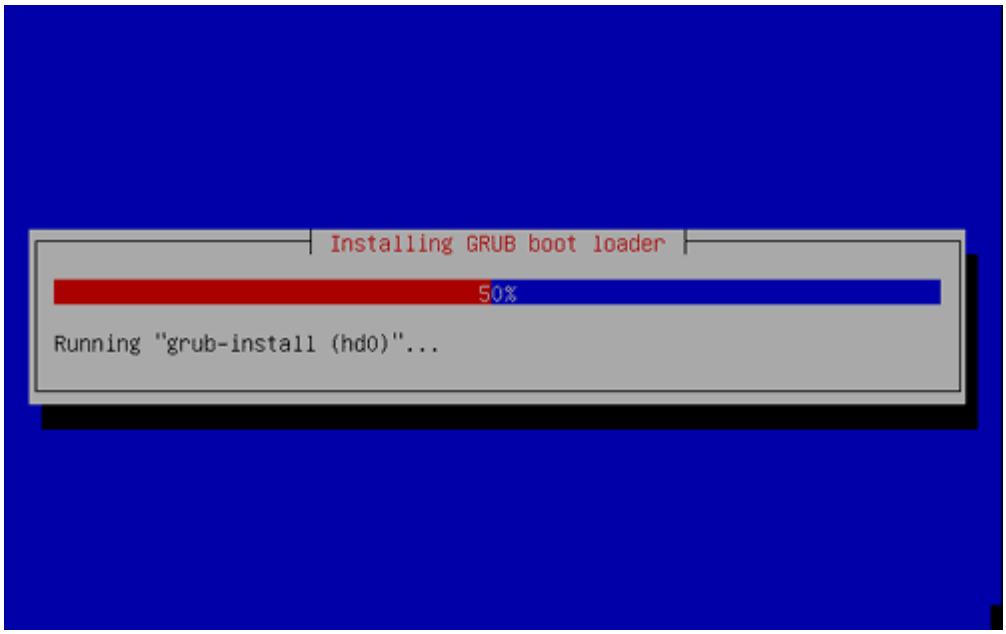

[!!] Set up users and passwords
A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.
Choose a password for the new user:
****
<Go Back>
<Continue>
<Tab> moves between items; <Space> selects; <Enter> activates buttons



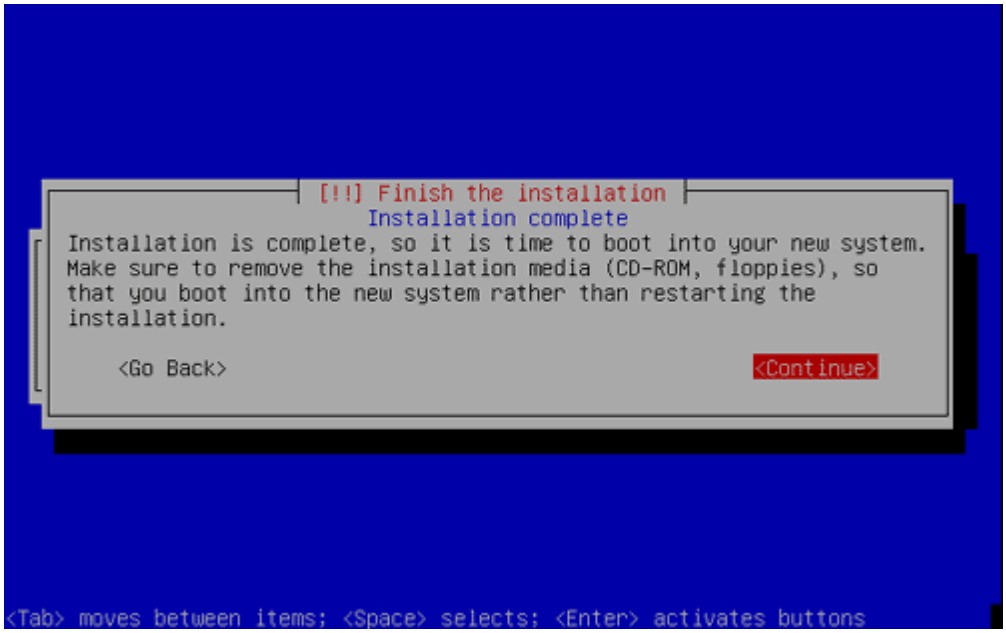
Теперь начинается собственно установка базовой системы:



Устанавливатеся загрузчик GRUB:



После завершения установки извлеките установочный диск и нажмите Continue для перезагрузки:



4.2. Включение пользователя root

Войдите в систему под записью administrator, созданной ранее. Посколько мы должны исполнять все установочные этапы под учетной записью root, в первую очередь нам придется ее разрешить.

Выполните команду

sudo passwd root

и присвойте пароль пользователю root. Перейдите в режим root командой

su

4.3. Установка сервера SSH

Ubuntu не устанавливает OpenSSH по умолчанию, поэтому сделаем это вручную. Вставьте установочный диск, затем выполните команду

apt-get install ssh openssh-server

В дальнейшем будет подразумеваться, что после этого момента вы не извлекали установочный диск.

4.4. Настройка сети

На этапе установки Ubuntu сконфигурировала систему получать динамический IP-адрес по DHCP, однако так как у нас серверная инсталляция, настроим ее использовать статический IP. Отредактируйте ***/etc/network/interfaces*** так как вы этого хотите (в примере используется IP *192.168.0.100*):

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 eth0
iface eth0 inet static
        address 192.168.0.100
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255
        gateway 192.168.0.1
```

Перезапустите сеть:

/etc/init.d/networking restart

Отредактируйте **/etc/hosts** на манер нижеуказанного:

vi /etc/hosts

```
127.0.0.1       localhost.localdomain    localhost
192.168.0.100   server1.example.com      server1
# 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
```

Выполните команды

echo server1.example.com > /etc/hostname

и перезагрузите систему

shutdown -r now

после этого выполните команды

hostname
hostname -f

теперь вы можете соединяться с вашим сервером через SSH.

4.5. Изменение списка /etc/apt/sources.list. Обновление инсталляции

Отредактируйте ***/etc/apt/sources.list***. Закомментируйте CD и разрешите некоторые дополнительные репозитории:

vi /etc/apt/sources.list

```
#
# deb cdrom:[Ubuntu-Server 6.10 _Edgy Eft_ - Release i386 (20061025.1)]/ edgy
main restricted

#deb cdrom:[Ubuntu-Server 6.10 _Edgy Eft_ - Release i386 (20061025.1)]/ edgy
main restricted
deb http://de.archive.ubuntu.com/ubuntu/ edgy main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ edgy main restricted
## Major bug fix updates produced after the final release of the
## distribution.
deb http://de.archive.ubuntu.com/ubuntu/ edgy-updates main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ edgy-updates main restricted
## Uncomment the following two lines to add software from the 'universe'
## repository.
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## universe WILL NOT receive any review or updates from the Ubuntu security
## team.
deb http://de.archive.ubuntu.com/ubuntu/ edgy universe
deb-src http://de.archive.ubuntu.com/ubuntu/ edgy universe
## Uncomment the following two lines to add software from the 'backports'
## repository.
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
# deb http://de.archive.ubuntu.com/ubuntu/ edgy-backports main restricted
universe multiverse
# deb-src http://de.archive.ubuntu.com/ubuntu/ edgy-backports main restricted
universe multiverse

deb http://security.ubuntu.com/ubuntu edgy-security main restricted
deb-src http://security.ubuntu.com/ubuntu edgy-security main restricted
deb http://security.ubuntu.com/ubuntu edgy-security universe
deb-src http://security.ubuntu.com/ubuntu edgy-security universe
```

Затем выполните команды:

apt-get update
apt-get upgrade

4.6. Изменение командной оболочки

/bin/sh указывает на /bin/dash, но нам нужен /bin/bash.

rm -f /bin/sh
ln -s /bin/bash /bin/sh

4.7. Установка приложений

Сейчас мы установим некоторые пакеты, которые пригодятся нам поздне.

apt-get install binutils cpp cpp-4.0 fetchmail flex gcc gcc-4.0 libarchive-zip-perl libc6-dev libcompress-zlib-perl libdb4.3-dev libpcre3 libpopt-dev linux-kernel-headers lynx m4 make ncftp nmap openssl perl perl-modules unzip zip zlib1g-dev autoconf automake1.9 libtool bison autotools-dev g++

(писать в одну строку)

4.8. Квотирование

Выполните команду

apt-get install quota

Исправьте **/etc/fstab**:

vi /etc/fstab

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
# /dev/sda1
UUID=02cc04f2-98cb-41db-8eb3-94de5f19b22b /               ext3
defaults,errors=remount-ro,usrquota,grpquota 0       1
# /dev/sda5
UUID=6b011d54-fb37-469d-9fa8-179b185343c1 none            swap    sw
0       0
/dev/hdc        /media/cdrom0   udf,iso9660 user,noauto     0       0
/dev/           /media/floppy0  auto    rw,user,noauto  0       0
```

Для разрешения квотирования выполните команды:
touch /quota.user /quota.group
chmod 600 /quota.*
mount -o remount /
quotacheck -avugm
quotaon -avug

4.9. MySQL

Выполните команду

apt-get install mysql-server mysql-client libmysqlclient15-dev

Мы хотим, чтобы MySQL прослушивала все сетевые интерфейсы, поэтому мы исправим **/etc/mysql/my.cnf** и закомментируем ***bind-address = 127.0.0.1***:

vi /etc/mysql/my.cnf

```
[...]
#bind-address          = 127.0.0.1
[...]
```

Перезапустим MySQL:

/etc/init.d/mysql restart

Проверим MySQL

netstat -tap

вывод должен выглядеть примерно так:

```
tcp        0      0 *:mysql                 *:*                     LISTEN      4997/mysqld
```

Выполните команды:

mysqladmin -u root password yourrootsqlpassword
mysqladmin -h server1.example.com -u root password yourrootsqlpassword

для установки доступа пользователя root (иначе ваши базы данных MySQL будут доступны всем).

4.10. Apache/PHP5

установим Apache:

apt-get install apache2 apache2-common apache2-doc apache2-mpm-prefork apache2-utils libapr0 libexpat1 ssl-cert

И PHP5:

apt-get install autoconf automake1.4 autotools-dev libapache2-mod-php5 php5 php5-common php5-curl php5-dev php5-gd php-pear php5-ldap php5-mhash php5-mysql php5-mysqli php5-snmp php5-sqlite php5-xmlrpc php5-xsl php5-imap php5-mcrypt php5-pspell

У вас будет запрошена версия libc-client:

Continue installing libc-client without Maildir support? <-- Yes

Исправим **/etc/apache2/apache2.conf**

vi /etc/apache2/apache2.conf

и поменяем ***DirectoryIndex***

```
[...]
DirectoryIndex index.html index.htm index.shtml index.cgi index.php index.php3 index.pl index.xhtml
[...]
```

исправим ***/etc/apache2/ports.conf*** и добавим ***Listen 443***:

vi /etc/apache2/ports.conf

```
Listen 80
Listen 443
```

Разрешим некоторые модули (*SSL*, *rewrite*, suexec, иinclude):

a2enmod ssl
a2enmod rewrite
a2enmod suexec
a2enmod include

перезагрузим конфигурацию

/etc/init.d/apache2 force-reload

4.11. Запретим PHP глобально (как системный скриптовый язык)

Отредактируем **/etc/mime.types** и закомментируем ***application/x-httpd-php lines***:

vi /etc/mime.types

```
[...]
#application/x-httpd-php                        phtml pht php
#application/x-httpd-php-source                   phps
#application/x-httpd-php3                  php3
#application/x-httpd-php3-preprocessed            php3p
#application/x-httpd-php4                  php4
[...]
```

Отредактируем **/etc/apache2/mods-enabled/php5.conf** следующим образом:

vi /etc/apache2/mods-enabled/php5.conf

```
<IfModule mod_php5.c>
#  AddType application/x-httpd-php .php .phtml .php3
#  AddType application/x-httpd-php-source .phps
</IfModule>
```

перезапустим Apache:

/etc/init.d/apache2 restart

4.12. Синхронизируем часы с NTP.

apt-get install ntp ntpdate

4.13 . Установка OpenVZ

Сначала мы опишем способ с компиляцией ядра Linux для Debian (Ubuntu является Debian Linux системой) с использованием патча OpenVZ. На выходе это дает готовый .deb пакет. Это необходимо, если вам необходимы драйвера, которые не вкомпилированы в ядра, доступные с systs.org. Если вам это не нужно, можете использовать скомпилированные ядра.

Подготовка основной системы

apt-get install kernel-package libncurses5-dev fakeroot wget bzip2

Компиляция ядра OpenVZ. Загрузка исходного кода ядра

Патч OpenVZ рассчитан на ядро версии 2.6.8. Мы бцдем использовать «ванильное» ядро с kernel.org and patch и конфигурировать его по нашим нуждам. Для загрузки выедите команды:

```
cd /usr/src
wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.8.tar.bz2
tar xjf linux-2.6.8.tar.bz2
cd linux-2.6.8/
```

Загрузка патча OpenVZ и модификация ядра

Загрузим патч с OpenVZ с OpenVZ.org и применим его к исходному коду.

```
wget http://download.openvz.org/kernel/stable/022stab078.14/patches/patch-022stab078-combined.gz
gzip -d patch-022stab078-combined.gz
patch -p1 < patch-022stab078-combined
```

Получение конфигурации ядра для OpenVZ

OpenVZ.org предлагает различные конфигурации ядра от типового i686 до промышленного. Мы выберем типовую i686 конфигурацию.Вы можете выбрать другую, в зависимости от вашего оборудования, типа процессора и объема памяти. Загрузить конфигурации можно с http://openvz.org/download/kernel/

```
wget http://download.openvz.org/kernel/stable/022stab078.14/configs/kernel-2.6.8-022stab078-i686.config.ovz
```

Выполните "**make menuconfig"**, выберите "***Load an alternate configuration file***" и выберите файл "***/usr/src/linux-2.6.8/kernel-2.6.8-022stab078-i686.config.ovz***".

make menuconfig

Если вы нуждаетесь в дополнительной конфигурации ядра, сделайте необходимые изменения, выберите ***Exit*** и затем ***Save*** чтобы сохранить конфигурацию ядра.

make-kpkg clean

Теперь соберем ядро.

fakeroot make-kpkg --revision=OpenVZ.2.6.8 kernel_image

Если компиляция останавливается с ошибкой, выполните

make clean

и перезапустите

make menuconfig

4.14 Установка ядра OpenVZ

Если вы не хотите компилировать ядро самостоятельно, вы можете установить скомпилированное ядро.

Добавьте репозиторий OpenVZ в ***/etc/apt/sources.list***:

echo "deb http://debian.systs.org/ stable openvz" >> /etc/apt/sources.list
apt-get update

Установите пакеты:

apt-get install kernel-image-2.6.8-stable-ovz

4.15 Установка утилит OpenVZ

apt-get install vzctl vzquota vzctl-template

перезагрузите сервер:

shutdown -r now

4.16 Установка и запуск ваших VPS

Шаблоны систем для VPS вы можете скачать с OpenVZ.org:

http://openvz.org/download/template/cache/
Мы опишем как установить и запустить шаблон Fedora 4 Minimal. Остальные шаблоны устанавливаются похожим образом. Если вам необходим собственный шаблон, смотрите далее.

Загрузим шаблон Fedora 4:

cd /var/lib/vz/template/cache
wget http://download.openvz.org/template/precreated/fedora-core-4-i386-minimal.tar.gz

Для создания VPS вы можете использовать любые ID более 100. Уникальность обязательна. Создадим VPS и настроим его стартовать при запуске аппаратного узла.

vzctl create 101 --ostemplate fedora-core-4-i386-minimal --config vps.basic
vzctl set 101 --onboot yes --save

Установим имя хоста:

vzctl set 101 --hostname test101.mytest.org --save

Установим IP и установим максимальное число сокетов Unix до 120:

vzctl set 101 --ipadd 192.168.0.167 --save
vzctl set 101 --numothersock 120 --save

Установим сервер DNS:

vzctl set 101 --nameserver 192.168.0.2 --save

Запустим VPS:

vzctl start 101

Запустим сервер SSH:

vzctl exec 101 /etc/init.d/sshd start

Установим пароль root:

vzctl exec 101 passwd

Теперь к серверу можно подключиться с помощью клиента SSH.

Получим статус сервера

vzctl status 101

Остановим VPS:

vzctl stop 101

Для просмотра состояния всех VPS на аппаратном узле выполните

vzlist –a

4.17 Генерация собственного шаблона Ubuntu.

Сначала установим утилиту ***debootstrap***. Она позволит нам загружать Debian Sarge и Ubuntu.

cd /tmp
wget
http://archive.ubuntulinux.org/ubuntu/pool/main/d/debootstrap/debootstrap_0.3.3.0ubuntu3_all.deb
dpkg -i debootstrap_0.3.3.0ubuntu3_all.deb

«Оживим» VPS с ID = 110.

debootstrap --arch i386 dapper /var/lib/vz/private/110
http://archive.ubuntulinux.org/ubuntu

применим базовую конфигурацию:

vzctl set 110 --applyconfig vps.basic --save

установим имя :

echo "OSTEMPLATE=ubuntu-6.06" >> /etc/vz/conf/110.conf

Установим IP и DNS Если вы получаете предупреждение: "*Warning: configuration file for distribution ubuntu-6.06 not found default used*", игнорируйте его.

vzctl set 110 --ipadd 192.168.0.169 --save
vzctl set 110 --nameserver 192.168.0.2 --save
vzctl set 110 --numothersock 120 --save

добавим список пакетов *sources.list*:

echo "deb http://archive.ubuntulinux.org/ubuntu dapper-updates main restricted" >> /var/lib/vz/private/110/etc/apt/sources.list
echo "deb http://archive.ubuntulinux.org/ubuntu dapper universe" >> /var/lib/vz/private/110/etc/apt/sources.list
echo "deb http://security.ubuntu.com/ubuntu dapper-security main restricted" >> /var/lib/vz/private/110/etc/apt/sources.list
echo "deb http://security.ubuntu.com/ubuntu dapper-security universe" >> /var/lib/vz/private/110/etc/apt/sources.list

Запустим VPS:

vzctl start 110

Обновим систему:

vzctl exec 110 apt-get update
vzctl exec 110 apt-get -u upgrade
vzctl exec 110 apt-get install ssh libedit2 openssh-client openssh-server quota

Запретим getty на терминалах:

vzctl exec 110 sed -i -e '/getty/d' /etc/inittab

Уберем некоторые монтировки:

vzctl exec 110 rm -f /etc/mtab
vzctl exec 110 ln -s /proc/mounts /etc/mtab

Удалим ненужные пакеты:

vzctl exec 110 apt-get clean

Очистим IP-адреса и остановим VPS:

vzctl set 110 --ipdel all --save
vzctl stop 110

Удалим ключи SSH и создадим скрипт, который их пересоздаст:

```
rm -f /var/lib/vz/private/110/etc/ssh/ssh_host_*
cat << EOF > /var/lib/vz/private/110/etc/rc2.d/S15ssh_gen_host_keys
#!/bin/bash
ssh-keygen -f /etc/ssh/ssh_host_rsa_key -t rsa -N ''
ssh-keygen -f /etc/ssh/ssh_host_dsa_key -t dsa -N ''
rm -f \$0
EOF
chmod a+x /var/lib/vz/private/110/etc/rc2.d/S15ssh_gen_host_keys
```

Запакуем VPS:

```
cd /var/lib/vz/private/110
tar czf /var/lib/vz/template/cache/ubuntu-6.06-minimal.tar.gz .
```

Примечание: Точка в конце является частью команды!
....tar.gz[пробел][точка]

Удалим VPS:

vzctl destroy 110

Для создания нового VPS на основе этого шаблона сделаем так:

vzctl create 102 --ostemplate ubuntu-6.06-minimal --config vps.basic

5. Миграция существующей системы в окружение OpenVZ.

(OpenVZ Wiki)

5.1 Подготовка нового пустого VE

Предположим, что мы присвоим ID 123 этому VE:

mkdir /vz/root/123 /vz/private/123
cat /etc/vz/conf/ve-vps.basic.conf-sample > /etc/vz/conf/123.conf

5.2 Подготовка к миграции

Остановите большинство сервисов на коспьютере. «Большинство» - означает такие как веб-сервер, СУБД – так вы не потеряете свои данные. Оставьте минимум, включая демон SSHD.

5.3 Копирование данных

Скопируйте все данные с машины в приватный раздел OpenVZ. Так как мы используем VE с ID 123, все данные нужно помещать в каталог **/vz/private/123/** (теперь она будет содержать каталоги vz/private/123/bin, etc, var и т.п.). Это можно сделать по-разному:

rsync

Пример с rsync – запускать с аппаратного узла:

rsync -arvpz --numeric-ids --exclude dev --exclude proc --exclude tmp -e "ssh -l root@a.b.c.d" root@a.b.c.d:/ /vz/private/123/

Преимущества: Ваша система остается онлайн.

Live CD

Другой путь состоит в загрузке с live cd, и использовании tar для сохранения в архиве на сетевой или USB диск.

Tar

Иной путь – использовать tar и исключить некотороые каталоги:
Создайте файл **/tmp/excludes.excl** со следующим содержанием:

.bash_history
/dev/*
/mnt/*
/tmp/*
/proc/*
/sys/*
/usr/src/*

Затем создайте архив . Запомните, если система не использует udev, вы должны проверите ветку /proc/ после создания VE , потому что некоторые устройства могут не перенестись (/dev/ptmx или другие)

 # tar cjpf /tmp/mysystem.tar.bz2 / -X /tmp/excludes.excl

Естественно, вы должны это делать только тогда, когда все критичные сервисы (MySQL, apache, ..) остановлены и временный раздел /tmp достаточен для создания архива.

Преимущества: Не нужен LiveCD, система остается онлайн.

5.4 Установка параметров VE

Шаблон ОС

Добавьте строку **OSTEMPLATE=xxx** в файл **/etc/vz/conf/123.conf**, где **xxx** равен названию дистрибутива (например **debian-3.0**) для того, чтобы vzctl могла применять изменеия, специфичные для этого дистрибутива.

IP адреса

Также вы должны назначить адреса новому VE:

vzctl set 123 --ipadd x.x.x.x --save

venet vs. veth

Вы можете использовать устройство veth вместо venet для обратной своместимости. Это может быть необходимо, если мигрируемый сервер некачественно сконфигурирован, и будет затруднительно найти все захардкоденные ссылки на интерфейсы.

5.5 «Тюнинг» VE

Так как VE слегка отличается от физического сервера, вы должны отредактировать некоторые файлы после миграции.

/etc/inittab

VE не имеет реальных терминалов, поэтому надо запретить getty в файле **/etc/inittab** (точнее в **/vz/private/123/etc/inittab**).

```
sed -i -e '/getty/d' /vz/private/123/etc/inittab
```

/etc/mtab

Сошлите /etc/mtab на /proc/mounts:

```
rm -f /vz/private/123/etc/mtab
ln -s /proc/mounts /vz/private/123/etc/mtab
```

/etc/fstab

Так как у вас больше нету реальных дисковых разделов, /etc/fstab (или его большая часть) больше не нужны. (Кроме /dev/pts):

```
cp /vz/private/123/etc/fstab /vz/private/123/etc/fstab.old
grep devpts /vz/private/123/etc/fstab.old > /vz/private/123/etc/fstab
```

Вы также можете примонтировать devpts в запущенном VE:

```
vzctl exec 123 mount -t devpts none /dev/pts
```

устройства /dev/TTY

При запуске VE необходимы некоторое количество устройств в ветке **/dev**. Это могут быть **/dev/ttyp*** и **/dev/ptyp***, или **/dev/ptmx** и примонтированные **/dev/pts.**

/dev/ptmx

Проверьте что **/dev/ptmx** существует. В противном случае - создайте:

mknod /vz/private/123/dev/ptmx c 5 2

/dev/pts/

Аналогично:

mkdir /vz/private/123/dev/pts

/dev/ttyp* and /dev/ptyp*

Проверьте на существование ветки **/dev/ttyp*** и **/dev/ptyp***. В противном случае создайте при помощи **/sbin/MAKEDEV**, или копированием с хост-системы.

копирование:

cp -a /dev/ttyp* /dev/ptyp* /vz/private/123/dev/

Пересоздание

/sbin/MAKEDEV -d /vz/private/123/dev ttyp ptyp

или

cd /vz/private/123/dev && /sbin/MAKEDEV ttyp

Другие устройства

/dev/urandom

Пересоздайте, если необходимо:

mknod /vz/private/123/dev/urandom c 1 9

/etc/init.d services

Некоторые системные службы могут быть (или будут) запрещены.
Наиболее вероятные:

- acpid, amd (не нужны)
- checkfs, checkroot (нет необходимости в проверке ФС средствами VE)
- clock (в VE не проверяются часы)
- consolefont (VE не имеет консоли)
- hdparm (VE не имеет реального жесткого диска)
- klogd (пока вы не заставили ваш iptables фиксировать какие-либо пакеты)
- keymaps (VE не имеет клавиатуры)
- kudzu (VE не оперирует физическим оборудованием)
- lm_sensors (VE не имеет доступа к аппаратным датчикам)
- microcodectl (VE не может обновлять микрокод процессора)
- netplugd (VE не имеет физической сетевой карты)

Для просмотра активизированных служб выполните

- RedHat/Fedora/SUSE: /sbin/chkconfig --list
- Gentoo: /sbin/rc-update show

Для отключения службы:

- RedHat/Fedora/SUSE: /sbin/chkconfig --del SERVICENAME
- Gentoo: /sbin/rc-update del SERVICENAME

5.6 Запуск нового VE

Попытайтесь запустить новый VE:

vzctl start 123

Если у вас возникают проблемы, см. ниже.

5.7 Устранение неисправностей

Нельзя войти в VE

Если вы не можете войти в VE (используя **vzctl enter**), сделайте следующее.

Обратитесь сначала к разделу #устройства /dev/TTY выше.

Затем проверьте что devpts примонтирован:

vzctl exec 123 mount | grep pts

Примонтируйте:

vzctl exec 123 mount -t devpts none /dev/pts

Затем, добавьте соответствующую команду в скрипты запуска. На некоторых дистрибутивах вам придется добавить соответствующую строку в ветку /etc/fstab на VE.

6. Миграция VDS между аппаратными узлами.

(OpenVZ Wiki)

«Заморозка» (checkpointing, CPT) является расширением ядра OpenVZ, позволяющим сохранять полное состояние запущенного VE и восстанавливать его позднее на другом аппаратном узле прозрачно для запущенных процессов и сетевых соединений. У этой техники есть разные применения, наиболее важное из которых непрерывная (безостановочная, zero-downtime) миграция VE.

До заморозки существовал один путь – остановка VE с последующей перезагрузкой. Эта процедура не только вызывает длительный простой, но и непрозрачна для клиентов, использующих службы, расположенные в VE. Это делает мграцию невозможной, в том случае, если клиенты неустойчивы к отказам сервера.

Для сравнения, заморозка позволяет мигрировать VE полностью прозрачно как для пользователей VE, так и для внешних клиентов служб VE. Она немного задерживает обслуживание на период миграции, не более.

6.1 Online-миграция

В состав OpenVZ входит утилита vzmigrate. С ее помощью мы можем осуществлять «горячую» (a.k.a. online) миграцию, т.е. как будто с VE ничего и не случалось. Это выполняется командой

vzmigrate --online <host> VEID

В процессе миграции все приватные данные VE сохраняются в файл, который переносится на целевой узел.

Для того, чтоб vzmigrate не запрашивала пароли, открытые ключи ssh public с сервера-источника должны быть доступны на сервере-получателе, в списке авторизованных ключей. Другими словами, команда **ssh root@host** не должна запрашивать пароль.

6.2 Функции ручной заморозки и восстановления

vzmigrate не единственное средство для миграции. Утилита vzctl предоставляет все необходимые возможности для этого.

заморозка:

vzctl chkpnt VEID --dumpfile <path>

Эта команда «замораживает» VE в файл дампа. Если ключ --dumpfile опущен, , vzctl использует умолчательный путь /vz/dump/Dump.VEID.

После этого возможно восстановление:

vzctl restore VEID --dumpfile <path>

Если дамп был перенесен на другой аппаратный узел, он точно так же может быть восстановлен.

Критическое требование состоит в том чтобы файловая система на момент восстановления была идентична моменту заморозки. Если это условие не выполняется, процесс восстановления может быть провален, или представляться процессам внутри VE как недоступность каких-либо файлов. Когда VE восстанавливается на том же узле, на которм был заморожен, это соблюдается автоматически. При переносе на другой узел требуется синхронизация файловых систем. vzctl не обеспеивает этого, поэтому требуется использование сторонних утилит (таких как rsync).

6.3 Пошаговая заморозка и восстановление.

Процесс заморозки может быть воспроизведен пошагово. Он состоит из трех этапов.

Первый этап– приостановка VE. На этом этапе CPT помещает процессы в особое состояние, и останавливает все сетевые интерфейсы VE. Это производится командой

vzctl chkpnt VEID --suspend

Второй этап – сериализация VE. В этой фазе CPT сохраняет состояние процессов и глобальное состояние VE в файл образа.

vzctl chkpnt VEID --dump --dumpfile <path>

Третий этап – остановка или восстановление процессов. Если миграция удалась, VE может быть остановлен командой:

vzctl chkpnt VEID --kill

Если миграция провалилась, восстановление может быть произведено командой:

vzctl chkpnt VEID --resume

Процесс восстановления состоит из 2 этапов. Первый – восстановление процессов, и их разморозка. После этого, они готовы к возобновлению исполнения.

vzctl restore VEID --undump --dumpfile <path>

Второй этап – восстановление работы VE, или его останов, если восстановление провалилось.

vzctl restore VEID --resume
vzctl restore VEID --kill

7. Резервное копирование и откат VDS.

(OpenVZ Wiki)

7.1 Vzdump

Vzdump – утилита, создающая копии исполняющихся VE. Она создает tar-архивы приватного раздела VE, включая коняигуарционные файлы. Существует несколько путей использования утилиты:

- Остановка VE перед резервированием (существенный простой)
- Использование rsync и приостановка/восстановление (минимальный простой)
- Использование LVM2 (горячее резервирование)

Vzdump сохраняет копию на диск в одном файле.

7.2 Загрузка vzdump

Вы можете загрузить vzdump rpm- или deb-пакеты с http://download.openvz.org/contrib/utils/vzdump/ или самую новую версию с http://www.proxmox.com/cms_proxmox/en/technology/oss-software/openvz/

Для rpm систем:

wget http://www.proxmox.com/cms_proxmox/cms/upload/vzdump/vzdump-0.4-1.noarch.rpm

Для Debian-систем:

wget http://www.proxmox.com/cms_proxmox/cms/upload/vzdump/vzdump_0.4-1_all.deb

7.3 Установка vzdump

Для rpm систем:

rpm -i vzdump-0.4-1.noarch.rpm

Для Debian-систем:

dpkg -i vzdump_0.4-1_all.deb

7.4 Сводные данные

vzdump OPTIONS [--all | <VEID>]

--compress	сжимает архив (gzip)
--dumpdir DIR	сохраняет архив в каталог DIR
--xdelta	создает дифференциальную копию
--mailto EMAIL	посылает уведомление на EMAIL
--stop	останавдивает/запускает VPS
--suspend	приостанавливает/возобновляет VPS
--snapshot	использует снимок LVM
--restore FILENAME	восстанавливает из резервной копии FILENAME

7.5 Примеры

Резервирование

Сохраним VE 777 – просто создадим архив приватного раздела VE и файлов конфигурации в каталог по умолчанию (обычно /vz/dump/).

vzdump 777

Используем rsync и приостановку сервера.

vzdump --suspend 777

Заархивируем все VE и пошлем письмо администратору.

vzdump --suspend --all --mailto root

Используем LVM2 для обеспечения безостановочного копирования в произвольный каталог.

vzdump --dumpdir /space/backup --snapshot 777

Восстановление

Восстановим предыдущий архив в VE с номером 600

vzdump --restore /space/backup/vzdump-777.tar 600

8. Использование трансляции адресов.

(OpenVZ Wiki)

Обычно. Вы назначаете открытые IP адреса для ваших VE. Иногда вы не захотите этого делать – нехватка адресов, и т.п..

Предварительные требования

IP перенаправление

IP перенаправление должно быть включено на аппаратном узле. Убедитесь что это так:

```
$ cat /proc/sys/net/ipv4/ip_forward
1
```

Команда должна вывести '1'. Если выводится '0', разрешите перенаправление IP как это описано в статье Quick installation#sysctl.

Отслеживание соединений IP

Необходимо разрешить остлеживание соединений на аппаратном узле. **Для ядер OpenVZ версии 2.6.8**, поместите в файл /etc/modprobe.conf строку:

```
modprobe ip_conntrack ip_conntrack_enable_ve0=1
```

и перезагрузите компьютер.

Для ядер OpenVZ позднее чем 2.6.8, разрешено по умолчанию, убедитесь что эта строка отсутствует в файлах /etc/modules.conf или /etc/modprobe.conf.

```
options ip_conntrack ip_conntrack_disable_ve0=1
```

Если она присутствует – удалите, или закомментируйте ее и перезагрузите компьютер.

Как предоставить доступ к Интернету из VE.

Для предоставления доступа для VE, имеющих исключительно внутренние адреса, на аппаратном узле должен быть настроен SNAT (Source Network Address Translation, или по другому - IP masquerading, IP-маскарадинг). Она доступна со стандартной утилитой Linux iptables. Для простейшей настройки SNAT выполните на аппаратном узле следующую команду:

 # iptables -t nat -A POSTROUTING -s src_net -o eth0 -j SNAT --to ip_address

где src_net – диапазон приватных адресов IP, принадлежащих VE, которые нужно обрабатывать при помощи SNAT, ip_address – внешний IP адрес вашего аппаратного узла. Правила могут быть множественными. Если вы используете несколько интерфейсов, вы можете их указывать явно: -o eth2. Для того, чтобы транслировать с помощью SNAT все адреса, выполните команду:

 # iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to ip_address

> **Примечание**: Если это не работает, попробуйте что-нибудь из нижеописанного.

1. Если вы используете стабильное 2.6.8 ядро, убедитесь что слежение соединений явно разрешено: В файле /etc/modprobe.conf должна присутствовать строка:

options ip_conntrack ip_conntrack_enable_ve0=1

> **Примечание**: в ядрах более поздних, чем 2.6.8, слежение соединений разрешено по умолчанию.

2. По неизвестным причинам, это не работает на хостах с Debian. Там это можно сделать так:

modprobe ip_conntrack ip_conntrack_enable_ve0=1

> Убедитесь, что этот модуль загружается перед любыми модулями iptables! Этот модуль загружается без опций, поэтому, после изменений необходимо перезагрузить компьютер.

Как сделать VE доступными из интернета.

В то же время, чтобы сделать некоторые службы в VE с приватными адресами, доступными из Интернета, нужно сконфигурировать DNAT (Destination Network Address Translation) на аппаратном узле. Для просто настройки DNAT выполните на аппаратном узле следующую команду:

```
# iptables -t nat -A PREROUTING -p tcp -d ip_address --dport port_num \
  -i eth0 -j DNAT --to-destination ve_address:dst_port_num
```

где, ve_address - это IP адрес VE, dst_port_num - это номер порта TCP, используемого маршрутизируемой службой, ip_address - внешний (публичный) IP адрес вашего аппаратного узла, и port_num - TCP порт аппаратного узла, на который отображается сервис из VE. Обратите внимание, что после занятия порта port_num в целях DNAT, он становится недоступен для использования любым другим службам на аппаратном узле. Кроме того, в этом случае включение SNAT обязательно.

Например, мы хотим предоставить доступ к веб-серверу на VE и оставить в живых веб-сервер на аппаратном узле:

```
# iptables -t nat -A PREROUTING -p tcp -d ip_address --dport 8080 \
  -i eth0 -j DNAT --to-destination ve_address:80
# iptables -t nat -A POSTROUTING -s ve_address -o eth0 -j SNAT --to ip_address
```

После этого вы сможете получать доступ к веб-серверу VE по адресу http://ip_address:8080/.

9. Маршрутизация по источнику.

(OpenVZ Wiki)

Иногда, когда в вашей сети имеется более одного маршрутизатора, возникает потребность, чтобы разные VE использовали разные маршрутизаторы. Для этого вы должны настроить маршрутизацию к источнику на аппаратном узле.

 # /sbin/ip rule add from $IP table $TBL
 # /sbin/ip route add default dev eth0 via $GW table $TBL

Где:

- $IP равен IP-адресу, который будет направляться на неумолчательный шлюз. Это может также быть маска, например 10.9.8.0/24, что означает, что все VE с адресами 10.9.8.x будут маршрутизироваться через этот шлюз.
- $GW – шлюз для этого адреса.
- $TBL – любой номер свободной таблицы. Обратите внимание, что номера 250-255 обычно зарезеривированы. (см /etc/iproute2/rt_tables).

Примечание: каждый неумолчательный шлюз требует новый номер таблицы.

Подробнее см. man ip.

10. Развертывание виртуальной частной сети

(OpenVZ Wiki)

Поддержка TUN/TAP ядром

OpenVZ поддерживает VPN внутри VE vчерез модули и устройства ядра TUN/TAP. Для того чтоб разрешить VE #101 использовать устройства TUN/TAP, нужно сделать следующее:

Убедитесь, что модуль **tun** загружен на аппаратном узле:

 # lsmod | grep tun

И если необходимо – загрузите:

 # modprobe tun

Вы также можете добавить его в файл /etc/modules.conf, чтобы при перезагрузке он загружался автоматически.

Разрешение доступа к TUN/TAP из VE

Разрешите VE использовать tun/tap на аппаратном узле:

 # vzctl set 101 --devices c:10:200:rw --save

И создайте символьное устройство внутри VE:

 # vzctl exec 101 mkdir -p /dev/net
 # vzctl exec 101 mknod /dev/net/tun c 10 200
 # vzctl exec 101 chmod 600 /dev/net/tun

Настройка VPN внутри VE

После этого настройте VPN так как вам необходимо.

Вы можете использовать следующее ПО для работы с VPN через TUN/TAP:

- Virtual TUNnel (http://vtun.sourceforge.net/)
- OpenVPN (http://openvpn.sourceforge.net/)

11. Виртуальный Ethernet-адаптер.

(OpenVZ Wiki)

Виртуальная сетевая карта - это ethernet-совместимое устройство, которое может быть использовано внутри VE. В отличие от venet-устройства, veth-устройство имеет MAC адрес. Поэтому, оно может быть использвано в различных настройках, и пользователь VE может полностью его настраивать, включая IP, шлюзы, и т.п.

> **Виртуальная ethernet-карта состоит из двух устройств - одно в VE0 и другое - в VE. Эти устройства соединены между собой, поэтому, если пакет идет в одно устройство, выйдет он из другого.**

Использование виртуальной сетевой карты

Модуль ядра

Сперва убедитесь, что загружен модуль vzethdev:

```
# lsmod | grep vzeth
vzethdev            8224  0
vzmon              35164  5 vzethdev,vznetdev,vzrst,vzcpt
vzdev               3080  4 vzethdev,vznetdev,vzmon,vzdquota
```

Если он не загружен, загрузите его:

```
# modprobe vzethdev
```

Вы можете добавить его запуск в файл /etc/init.d/vz, тогда он будет загружен при начальной загрузке системы.

> **Примечание**: начиная с vzctl версии 3.0.11, vzethdev загружается через /etc/init.d/vz

Добавление veth в VE

> **Примечание**: Используйте случайный адрес MAC. Не используйте адреса MAC реальных устройств, потому что это может вызывать конфликты адресов MAC. Адреса вводятся в формате XX:XX:XX:XX:XX:XX.

синтаксис для vzctl версии < 3.0.14

vzctl set <VEID> --veth_add <dev_name>,<dev_addr>,<ve_dev_name>,<ve_dev_addr>

Где

- dev_name имя, устройства, создаваемого на аппаратном узле.
- dev_addr - его MAC адрес
- ve_dev_name - соответственнное имя устройства в VE
- ve_dev_addr - соответственно MAC этого устройства.

> **Примечание**: этот механизм действует инкрементально, так что новые устройства будут добавляться к существующим.

Не оставляйте пробелов после запятых.

Пример:

vzctl set 101 --veth_add veth101.0,00:12:34:56:78:9A,eth0,00:12:34:56:78:9B --save

После выполнения этой команды veth-устройство для VE 101 будет добавлено, и его настройки будут сохранены в файле конфигурации VE. Хост-адаптер будет иметь имя veth101.0 и MAC 00:12:34:56:78:9A. VE-адаптер будет иметь имя eth0 и MAC 00:12:34:56:78:9B.

синтакисис для vzctl версии >= 3.0.14

vzctl set <VEID> --netif_add <ifname>[,<mac>,<host_ifname>,<host_mac]

Где

- ifname - имя устройства в VE
- mac - MAC адрес устройства в VE
- host_ifname – имя устройства на хосте (VE0)
- host_mac - MAC адрес устройства на хосте (VE0)

> **Примечание**: Все парметры кроме ifname необязательны и генерируются автоматически при умолчании.

Пример:

vzctl set 101 --netif_add eth0,00:12:34:56:78:9A,veth101.0,00:12:34:56:78:9B --save

Удаление виртуальной Ethernet-карты из VE

синтаксис для vzctl версии < 3.0.14

vzctl set <VEID> --veth_del <dev_name>

Где dev_name имя устройства на хосте.

Пример:

vzctl set 101 --veth_del veth101.0 --save

синтаксис для vzctl версии >= 3.0.14

vzctl set <VEID> --netif_del <dev_name>|<all>

Где

- dev_name - имя устройства в VE.

> **примечание**: для удаления всех адаптеров в VE, используйте all.

Пример:

vzctl set 101 --netif_del eth0 --save

Общие настройки для виртуальных устройств ethernet

Необходимо иметь загруженным модуль vzethdev.

Простая настройка

Запуск VE

[host-node]# vzctl start 101

добавление в VE

[host-node]# vzctl set 101 --veth_add
veth101.0,00:12:34:56:78:9A,eth0,00:12:34:56:78:9B --save

настройка на хосте

[host-node]# ifconfig veth101.0 0
[host-node]# echo 1 > /proc/sys/net/ipv4/conf/veth101.0/forwarding
[host-node]# echo 1 > /proc/sys/net/ipv4/conf/veth101.0/proxy_arp
[host-node]# echo 1 > /proc/sys/net/ipv4/conf/eth0/forwarding
[host-node]# echo 1 > /proc/sys/net/ipv4/conf/eth0/proxy_arp

настройка в VE

```
[host-node]# vzctl enter 101
[ve-101]# /sbin/ifconfig eth0 0
[ve-101]# /sbin/ip addr add 192.168.0.101 dev eth0
[ve-101]# /sbin/ip route add default dev eth0
```

Добавление маршрута на хосте

```
[host-node]# ip route add 192.168.0.101 dev veth101.0
```

Работа с IPv6

Запуск VE

```
[host-node]# vzctl start 101
```

добавление в VE

```
[host-node]# vzctl set 101 --veth_add
veth101.0,00:12:34:56:78:9A,eth0,00:12:34:56:78:9B --save
```

Настройка на хосте

```
[host-node]# ifconfig veth101.0 0
[host-node]# echo 1 > /proc/sys/net/ipv6/conf/veth101.0/forwarding
[host-node]# echo 1 > /proc/sys/net/ipv6/conf/eth0/forwarding
[host-node]# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

Настройка в VE

```
[host-node]# vzctl enter 101
[ve-101]# /sbin/ifconfig eth0 0
```

Запуск оповещения маршрутизатора (router advertisement daemon (radvd)) для IPv6 на хосте

Ниже приведен пример конфигурационного файла /etc/radv.conf:

```
interface veth101.0
{
	AdvSendAdvert on;
	MinRtrAdvInterval 3;
	MaxRtrAdvInterval 10;
	AdvHomeAgentFlag off;

	prefix 3ffe:2400:0:0::/64
	{
		AdvOnLink on;
		AdvAutonomous on;
		AdvRouterAddr off;
	};
};

interface eth0
{
	AdvSendAdvert on;
	MinRtrAdvInterval 3;
	MaxRtrAdvInterval 10;
	AdvHomeAgentFlag off;

	prefix 3ffe:0302:0011:0002::/64
	{
		AdvOnLink on;
		AdvAutonomous on;
		AdvRouterAddr off;
	};
};
```

Теперь запустите radvd:

[host-node]# /etc/init.d/radvd start

Добавьте адреса IPv6 к устройствам в VE0

[host-node]# ip addr add dev veth101.0 3ffe:2400::212:34ff:fe56:789a/64
[host-node]# ip addr add dev eth0 3ffe:0302:0011:0002:211:22ff:fe33:4455/64

Виртуальные сетевые карты могут быть объединены одним мостом.

Создание моста

```
[host-node]# brctl addbr vzbr0
```

Добавление сетевых карт к мосту

```
[host-node]# brctl addif vzbr0 veth101.0
...
[host-node]# brctl addif vzbr0 veth101.n
[host-node]# brctl addif vzbr0 veth102.0
...
...
[host-node]# brctl addif vzbr0 vethXXX.N
```

Настройка моста.

```
[host-node]# ifconfig vzbr0 0
[host-node]# echo 1 > /proc/sys/net/ipv4/conf/vzbr0/forwarding
[host-node]# echo 1 > /proc/sys/net/ipv4/conf/vzbr0/proxy_arp
```

добавление маршрутов в хосте VE0

```
[host-node]# ip route add 192.168.101.1 dev vzbr0
...
[host-node]# ip route add 192.168.101.n dev vzbr0
[host-node]# ip route add 192.168.102.1 dev vzbr0
...
...
[host-node]# ip route add 192.168.XXX.N dev vzbr0
```

Обеспечение постоянства виртуальной сетевой карты.

Вышеописанных команд недостаточно, чтобы сетевая карта постоянно присутствовала в VE. Ниже описывается способ обеспечить постоянное присутствие сетевой карты в VE.

Очистка ${VEID}.conf

A)

Откройте файл /etc/vz/conf/VEID.conf и закомментируйте все вхождения IP_ADDRESS чтобы предупредить создание VENET-устройства в VE. Добавьте или измените значение CONFIG_CUSTOMIZED="yes".

B)

Выполните пункт A и дополнительно допишите VETH_IP_ADDRESS="<your VE IP>" в /etc/vz/conf/VEID.conf

Добавление внешнего скрипта в хост

Скопируйте и вставьте соответствующий код (A или B) в /usr/sbin/vznetaddroute:

A)

```
#!/bin/bash
#
# This script adds the appropriate VE0-route for veth-enabled VEs.
# See http://wiki.openvz.org/Virtual_Ethernet_device for more information.
#

# check the VEID
if [ "${VEID}" == 101 ]; then
  echo "Adding interface veth101.0 and route 192.168.0.101 for VE101 to VE0"
  /sbin/ifconfig veth101.0 0
  echo 1 > /proc/sys/net/ipv4/conf/veth101.0/forwarding
  echo 1 > /proc/sys/net/ipv4/conf/veth101.0/proxy_arp
  echo 1 > /proc/sys/net/ipv4/conf/eth0/forwarding
  echo 1 > /proc/sys/net/ipv4/conf/eth0/proxy_arp
  /sbin/ip route add 192.168.0.101 dev veth101.0
elsif [ "${VEID}" == 102 ]; then
  echo "Adding interface veth102.0 and route 192.168.0.102 for VE101 to VE0"
  /sbin/ifconfig veth101.0 0
  echo 1 > /proc/sys/net/ipv4/conf/veth102.0/forwarding
  echo 1 > /proc/sys/net/ipv4/conf/veth102.0/proxy_arp
  echo 1 > /proc/sys/net/ipv4/conf/eth0/forwarding
  echo 1 > /proc/sys/net/ipv4/conf/eth0/proxy_arp
  /sbin/ip route add 192.168.0.102 dev veth102.0
elsif [ "${VEID}" == YOUR_VE ]; then
  # same as above with the vethYOUR_VE.0 device and the appropriate ip
fi
exit
```

Добавьте по одной секции elsif для каждого VE с виртуальной сетевой картой. Сделайте

chmod +x /usr/sbin/vznetaddroute
Чтобы разрешить скрипту исполняться.

B)

```
#!/bin/bash

VZCONFDIR=/etc/vz

. $VZCONFDIR/conf/$VEID.conf

VZHOSTIF=`echo $NETIF |sed 's/^.*host_ifname=\(.*\),.*$/\1/g'`

if [ -n $VETH_IP_ADDRESS ]; then
      echo "Adding interface $VZHOSTIF and route $VETH_IP_ADDRESS for VE$VEID
to VE0"
      /sbin/ifconfig $VZHOSTIF 0
      echo 1 > /proc/sys/net/ipv4/conf/$VZHOSTIF/proxy_arp
      echo 1 > /proc/sys/net/ipv4/conf/$VZHOSTIF/forwarding
      /sbin/ip route add $VETH_IP_ADDRESS dev $VZHOSTIF
else
      echo "found no VETH_IP_ADDRESS in  $VZCONFDIR/conf/$VEID.conf!"
      exit 1;
fi
exit
```

Сделайте скрипт исполняемым

chmod +x /usr/sbin/vznetaddroute
to make the script executable.

Дополнительно вам потребуется отредактировать /etc/network/interfaces:

```
auto eth0
iface eth0 inet static
      address 10.1.1.1
      netmask 255.255.255.0
      network 10.1.1.0
      broadcast 10.1.1.255
      gateway 10.215.1.254
      dns-nameservers 10.215.1.20
      dns-search prod.your.domain
      up sysctl -w net.ipv4.conf.eth0.proxy_arp=1
      up sysctl -w net.ipv4.conf.eth0.forwarding=1
```

после этого вручную выполните:

```
sysctl -w net.ipv4.conf.eth0.proxy_arp=1
sysctl -w net.ipv4.conf.eth0.forwarding=1
```

Разрешите vzctl запускать скрипт

Скопируйте и вставьте строки в /etc/vz/vznet.conf:

```
#!/bin/bash
EXTERNAL_SCRIPT="/usr/sbin/vznetaddroute"
```

Этот скрипт будет исполняться при каждом старте VE.

добавление скрипта в VE

Запустим VE
vzctl start 101

, войдем в него
vzctl enter 101

создадим новый файл /etc/init.d/route-up следующего содержания:

```
#!/bin/bash
/sbin/ip route add default dev eth0
```

Сделаем скрипт исполняемым
chmod +x /etc/init.d/route-up

и добавим к runlevels:

```
ve101:/# update-rc.d route-up defaults
 Adding system startup for /etc/init.d/route-up ...
  /etc/rc0.d/K20route-up -> ../init.d/route-up
  [...]
```

Проверка

покинем VE
exit

, остановим
vzctl stop 101

и перезапустим
vzctl start 101
.
Оставаясь на хосте, проверим маршрут в VE:
ve0:/# ip route ls
192.168.0.101 dev veth101.0 scope link
[...]
ve0:/# ping 192.168.0.101 -c 4 -q
[...]
--- 192.168.0.101 ping statistics ---
4 packets transmitted, 4 recieved, 0% packet loss, time 0ms
Если что-то сломалось, проверим те файлы которые мы меняли.
vzctl enter 101
и проверим маршрутизацию:

```
ve101:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:12:34:56:78:9B
          inet addr:192.168.0.101  Bcast:0.0.0.0  Mask:255.255.255.255
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6757 (6.5 KiB)  TX bytes:10396 (10.1 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
ve101:/# ip route ls
default dev eth0  scope link
ve101:/# ping 192.168.0.101 -c 4 -q
[...]
--- 192.168.0.101 ping statistics ---
4 packets transmitted, 4 recieved, 0% packet loss, time 0ms
```

Дополнительно

- Virtual network device
- Differences between venet and veth

Ссылки

- Linux IPv6 HOWTO, a chapter about radvd

12. Виртуальное сетевое устройство.

(OpenVZ Wiki)

Использование

Модуль ядра

Убедитесь что модуль venetdev загружен:

 # lsmod | grep vznetdev

При необходимости загрузите:

 # modprobe vznetdev

Проверьте /etc/init.d/vz чтобы убедиться, что модуль загружается при загрузке ядра.

Добавление адреса IP в VE

vzctl set <VEID> --ipadd <IP1>[,<IP2>,...] [--save]

> **Примечание**: Имеет инкрементальный характер, все IP адреса будут добавляться к существующим.

Пример

vzctl set 101 --ipadd 10.0.0.1 --save

Удаление IP адресов из VE

vzctl set <VEID> --ipdel <IP1>[,<IP2>,...] [--save]
vzctl set <VEID> --ipdel all [--save]

Пример

vzctl set 101 --ipdel 10.0.0.1

Дополнительно

- Veth
- Differences between venet and veth

13. Использование X-приложений в VDS.

(OpenVZ Wiki)

X перенаправление

Для запуска приложений X Window внутри VE, вам всего лишь надо соединиться с VE через **ssh с ключом –X**:

host# ssh –X user@address

После соединения с VE проверьте что переменная $DISPLAY определена и X11 перенаправление разрешено:

ve# echo $DISPLAY
localhost:10.0

В случае, когда $DISPLAY не определена, убедитесь, что X перенаправление разрешено в конфигурации sshd config внутри VE. В большинстве дистрибутивов Linux SSH хранит конфигурацию в /etc/ssh/sshd_config. Вы должны установить параметр X11Forwarding в yes. Также, VE должен иметь установленный пакет xauth, установите его если он отсутствует. После этого перезапустите SSHD:

ve# /etc/init.d/sshd restart

> **Примечание**: Не забудьте пересоединиться с VE ☐

Теперь вы можете запускать графические приложения в VE :

ve# firefox

Если вы хотите запускать полное X-окружение, (в т.ч оконный менеджер), вы должны остановтиь локальный оконный менеджер, и запустить только чистый X-сервер. Кроме того, вы должны запускать SSH с ключом **-Y**. И если вы хотите использовать gnome/kde/..., не забудьте увеличить ограничения UBC, т.к. значения по умолчанию не рассчитаны на запуск таких монстров. ;)

VNC для X

Сначала вы должны запустить **Xvnc** сервер внутри VE. Самый легкий путь - сделать это с помощью скрипта **vncserver**. Он запускает все необходимые службы, и небольшой http демон который предоставляет графический доступ через веб (с использованием апплета Java).

```
ve# vncserver -name mydesktop
New 'mydekstop' desktop is ve:1

Starting applications specified in ~/.vnc/xstartup
Log file is ~/.vnc/ve:1.log
```

Теперь вы может присоединиться к своему рабочему столу при помощи команды **vncviewer**:

```
host# vncviewer <VE_IP>:1
```

Если рабочий стол больше, чем разрешение вашего X терминала, то вы увидите линейки прокрутки. Чтобы этого не происходило, уменьшите размер стола:

```
vncserver –geometry 1000x650
```

Это вполне подходит для разрешения 1024 на 768.

Запуск KDE внутри VNC

Для запуска KDE внутри twm замените строку `twm &` на `startkde &` в файле `~/.vnc/xstartup` внутри VE.

Соединение с VNC из-за сетевого экрана.

VNC использует сетевые порты от 5900/TCP. Эти порты могут быть закрыты внутри разных сетей. В этом случае вам потребуется туннелирование VNC. Обычно для этого используется **ssh**.

```
localhost# ssh -L 5900:localhost:5900 <remote host>
```

где <remote host> - имя удаленного компьютера с VNC. Затем вы можете запускать VNC Viewer.

```
localhost# vncviewer localhost
```

Использование xdm

Вы также можете использовать **xdm** внутри VE и X -query <remote IP> :1
Но VNC использовать легче.

Ссылки

- http://forum.openvz.org/index.php?t=tree&th=235&mid=1115&&rev=&reveal=
- http://ait.web.psi.ch/services/linux/kde-desktop-sharing.htm
- http://ait.web.psi.ch/services/ssh/vnc-ssh.html
- http://www.vnc.com/pipermail/vnc-list/2002-July/031831.html
- http://www.linuxjournal.com/article/5499
- http://www.realvnc.com/pipermail/vnc-list/2002-March/029502.html
- http://www.redhat.com/archives/rhl-list/2003-December/msg01859.html
- http://faq.gotomyvnc.com/fom-serve/cache/56.html

14. Особенности работы с NFS

(OpenVZ Wiki)

Сервер NFS

В настоящее время ядро OpenVZ k не включает поддержку сервера NFS уровня ядра. Однако вы можете использовать сервер NFS пользовательского режима в VE.

Сервер NFS внутри VE

Существуют два пути создать сервер NFS на общем аппаратном узле: использовать демон NFS-сервера режима пользователя, или использовать внутриядерную реализацию сервера NFS.

Сервер NFS режима ядра

Бинарные пакеты RPM, предоставляемые командой OpenVZ содержат ядра без поддержки NFS. Поэтому вам придется перекомпилировать ядро с опцией CONFIG_NFSD=m. После загрузки с таким ядром, вы можете использовать на аппаратном узле сервер NFS. Ядерный сервер NFS для обслуживания запросов клиентов запускает потоки команд ядра. Но по причинам безопасности, потоки ядра запрещены в VE! Поэтому вы не можете использовать ядерный сервер NFS внутри VE без модификации ядра.

Сервер NFS пользовательского режима

Плюс такого сервера в том, что он не требует поддержки ядром. Если он откажет – это не вызовет крах системы. Минусы – производительность, сервер режима ядра всегда быстрее.

Одна хорошо известная реализация сервера NFS - "The LINUX User-Space NFS Server" от Олафа Кирха (Olaf Kirch). Некоторые дистрибутивы Linux включают этот пакет: Debian Sarge (nfs-user-server), OpenSUSE 10.0 (nfs-server). Для других дистрибутивов, вы можете загрузить исходные коды и компилировать их. Есть небольшой трюк касающийся mountd и nfsd (оба этих демона и portmap составляют сервер пользовательского режима). Вы можете запускать их с ключом -r:

```
# portmap
# rpc.mountd -r
# rpc.nfsd -r
```

Причина этого состоит в том, что эти демоны проверяют главный нмоер устройства перед экспортом. Если он равен 0 тогда демоны считают, что это NFS и отказываются реэкспортировать их. Показателем этого является то что клиенты всегда получают ошибку "permission denied". Simfs (файловая система на которой расположен VE) ассоциирована с неименованным устройством, главный номер которого также равен 0. Поэтому мы запускаем эти демоны с ключом –r дабы подавить ошибки из-за отказа реэкспортирования.

“The LINUX User-Space NFS Server” Олафа Кирха реализует NFSv2. Поэтому он обрабатывает файлы размером менее 2 Гб. Если вам требуется поддержка обработки более массивных файлов, используйте другой сервер режима пользователя, unfs3. Он реализует версию 3 стандарта NFS.

Клиент NFS

Приготовления

ПО

Вам потребуется:

- ядро версии 028test006 или выше
- vzctl версии 3.0.13 и выше

Подготовка аппаратного узла

```
# modprobe nfs
```

Подготовка VE

Для того чтобы VE мог использовать NFS, вы должны запустить его с разрешенной опцией "nfs". Если VE запущен, когда вы устанавливаете опцию --features "nfs:on", вы должны перезагрузить его.

```
# vzctl set 101 --features "nfs:on" --save
# vzctl start 101
```

Посел этого вы можете видеть nfs в /proc/filesystems

```
# vzctl exec 101 cat /proc/filesystems
      ext3
      ext2
nodev   rpc_pipefs
nodev   proc
nodev   nfs
nodev   sysfs
nodev   tmpfs
nodev   devpts
```

Монтирование NFS

Предположим ваш NFS сервер установлен на узле 192.168.0.1:/nfs_pub

```
# vzctl enter 100
# mkdir /nfs
# mount -t nfs 192.168.0.1:/nfs_pub /nfs
# cat /proc/mounts
simfs / simfs rw 0 0
proc /proc proc rw 0 0
sysfs /sys sysfs rw 0 0
devpts /dev/pts devpts rw 0 0
nfs /nfs nfs
rw,vers=3,rsize=32768,wsize=32768,hard,proto=tcp,timeo=600,retrans=2,sec=sys,
addr=192.168.0.1 0 0
```

Дополнительно читайте NFS-client HOWTO

Ссылки

- Linux NFS Overview, FAQ and HOWTO Documents
- The Network Administrators' Guide by Olaf Kirch
- unfs3 homepage
- Overview of nfs-user-server source package

15. Yum

(OpenVZ Wiki)

Установка

Скачайте файл openvz.repo и поместите его в каталог /etc/yum.repos.d/

Использование

Обновление

Если у вас уже установлена OpenVZ выполняйте

 # yum update

Периодически для своевременного обновления вашей системы.

Новая установка

Выполните

 # yum install ovzkernel vzctl

Для установки OpenVZ на вашу систему.

> **Примечание**: простой установки этих пакетов недостаточно для установки OpenVZ и запуска системы. См. Quick installation.

Дополнительные ядра

В файле openvz.repo по умолчанию разрешен только один репозиторий, — содержащий стабильные ядра и инструментарий. Если вы хотите запускать другие ядра OpenVZ (например, в текущей разработке), отредактируйте /etc/yum.repos.d/openvz.repo вручную, изменив строки enable=0 на enable=1 для необходимых вам ядер.

Решение проблем

Fedora Core 6: извещение о более новом ядре

В Fedora 6 вы можете получить ошибку, устанавливая ядро OpenVZ с использованием yum, наподобие этой:

Transaction Check Error: package kernel-2.6.18-1.2869.fc6 (which is newer than kernel-2.6.18-ovz028test010.1) is already installed
package kernel-2.6.19-1.2895.fc6 (which is newer than kernel-2.6.18-ovz028test010.1) is already installed

Для того, чтобы избежать этого, запретите пакеты установки ядра в репозиториях fedora-core и fedora-updates, добавив строку:

exclude=kernel kernel-smp kernel-enterprise

в файлы

- Секция [core] в /etc/yum.repos.d/fedora-core.repo
- Секция [updates] в /etc/yum.repos.d/fedora-updates.repo

16. OpenVZ Live CD

(OpenVZ Wiki)

Содержимое компакт-диска

CD содержит следующие пакеты OpenVZ

- kernel 2.6.18-028stab027
- vzctl 3.0.16
- vzquota 3.0.9

плюс шаблоны:

- Debian 3.1 minimal
- CentOS 4 minimal
- Fedora Core 5 minimal

Кроме того, он включает в себя ПО с дистрибутива Knoppix 5.1.1 CD, кроме:

- OpenOffice
- GIMP
- Frozen Bubble

Эти пакеты были изъяты для того, чтобы включить OpenVZ в состав.

Закачка

ISO-образ доступен напрямую со страницы загрузки download.openvz.org/livecd/ и с любого из его зеркал download mirrors.

Файл	Объем
KNOPPIX_V5.1.1-OPENVZ-CD-2007-01-04-EN.iso	683 MB
KNOPPIX_V5.1.1-OPENVZ-CD-2007-01-04-EN.iso.md5	77 B
KNOPPIX_V5.1.1-OPENVZ-CD-2007-01-04-EN.iso.asc	189 B

Запись

После скачивания вы должны записать образ на компакт-диск. Вы можете это сделать предпочитаемой Вами программой записи компакт-дисков (cdrecord, Small CD-Writer, Nero). Инструкции по записи выходят за рамки данной инструкции.

Загрузка

Вставьте CD в CD-ROM и перезагрузите компьютер. В некоторых случаях вам придется поменять настройки BIOS чтобы разрешить загрузку с CD.

Ограничения

Так как это Live CD, все изменения создаются в ОЗУ, т.е. не сохраняются при перезагрузке.

Количество VE, которые вы можете создать, зависит от объем памяти. Имея около 1 Гб памяти вы можете создать около 5 VE. Дисковая квота OpenVZ disk quota не работает на LiveCD по причине OpenVZ Bug #558.

Изменения

- 7 May 2007, первая версия, базируемая на Knoppix 5.1.1

Ссылки

- knoppix.com, домашний сайт Knoppix

Использование OpenVZ live CD

(OpenVZ Wiki)

Создание VE

Мы создадим VE с ОС Debian и VEID = 101. Откройте терминал и выполняйте следующие команды под учетной записью root:

```
root@Knoppix:~# vzctl create 101 --ostemplate debian-3.1-i386-minimal
Creating VE private area (debian-3.1-i386-minimal)
Performing postcreate actions
VE private area was created
```

vzctl – это утилита управления VE. Посмотрите в каталог **/var/lib/vz/template/cache/** чтобы узнать какие еще шаблоны ОС поставляются с LiveCD:

```
root@Knoppix:~# ls -1 /var/lib/vz/template/cache/
centos-4-i386-minimal.tar.gz
debian-3.1-i386-minimal.tar.gz
fedora-core-5-i386-minimal.tar.gz
```

Перечисление VE

Вы можете получить список созданных на аппаратном узле VE с помощью команды **vzlist**:

```
root@Knoppix:~#  vzlist -a
      VEID      NPROC STATUS  IP_ADDR         HOSTNAME
       101          - stopped -              -
```

Запуск VE

```
root@Knoppix:~# vzctl start 101
Starting VE ...
VE is mounted
Setting CPU units: 1000
VE start in progress...
root@Knoppix:~# vzlist -a
      VEID      NPROC STATUS  IP_ADDR         HOSTNAME
       101          5 running              -
```

Выполнение команд в VE

Из результатов предыдущей команды мы видим, что в VE 101 выполняется 5 процессов. На аппаратном узле вы можете использовать команду **ps** для того чтоб узнать какие именно процессы были запущены, похожий подход применяется и в отношении VE, с той разницей что делается это при помощи команды **vzctl exec**:

```
root@Knoppix:~# vzctl exec 101 ps
  PID TTY        TIME CMD
    1 ?      00:00:00 init
 7672 ?        00:00:00 rc
 7674 ?        00:00:00 S10sysklogd
 7677 ?        00:00:00 syslogd
 7678 ?        00:00:00 syslogd
 7683 ?        00:00:00 ps
```

Вход в VE

Вы можете получить командную оболочку VE:

```
root@Knoppix:~# vzctl enter 101
entered into VE 101
Knoppix:/#
```

В оболочке вы можете делать все то же, что и на аппаратном узле:

```
Knoppix:/# useradd new-user
Knoppix:/# passwd new-user
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Knoppix:/# mkdir /home/new-user
Knoppix:/# chown new-user /home/new-user/
Knoppix:/# su new-user
Knoppix:/$ cd ~
Knoppix:~$ pwd
/home/new-user
exit
Knoppix:/#
```

Для выхода из VE наберите **exit**:

```
Knoppix:/# exit
logout
exited from VE 101
root@Knoppix:~#
```

Настройка сети в VE

```
root@Knoppix:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@Knoppix:~# ifconfig venet0 up
root@Knoppix:~# vzctl set 101 --ipadd 10.1.1.1 --save
Adding IP address(es): 10.1.1.1
Saved parameters for VE 1
root@Knoppix:~# vzlist -a
     VEID      NPROC STATUS  IP_ADDR         HOSTNAME
      101          4 running 10.1.1.1        -
```

Теперь ваш аппаратный узел и VE связаны и могут например обнаружить друг друга командой **ping**:

```
root@Knoppix:~# ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=64 time=3.80 ms

--- 10.1.1.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.804/3.804/3.804/0.000 ms
root@Knoppix:~#
root@Knoppix:~# vzctl exec 101 ping 192.168.0.244
PING 192.168.0.244 (192.168.0.244) 56(84) bytes of data.
64 bytes from 192.168.0.244: icmp_seq=1 ttl=64 time=0.508 ms

root@Knoppix:~#
```

Кроме того, вы можете пинговать и другие компьютеры в сети, но для этого необходимо настроить NAT (Network Address Translation) и сервер имен.

В данном примере подразумевается что адрес вашего аппаратного узла 192.168.0.244 и адрес сервера имен 192.168.1.1.

```
root@Knoppix:~# iptables -t nat -A POSTROUTING -s 10.1.1.1 -o eth0 -j SNAT --to
192.168.0.244
root@Knoppix:~# vzctl set 101 --nameserver 192.168.1.1 --save
File resolv.conf was modified
Saved parameters for VE 101
root@Knoppix:~# vzctl exec 101 ping google.com
PING google.com (64.233.167.99) 56(84) bytes of data.
64 bytes from py-in-f99.google.com (64.233.167.99): icmp_seq=1 ttl=241
time=23.0 ms
```

Установка ПО внутри VE

По умолчанию, комплектация VE минимальна. Но вы можете устанавливать необходимые вам пакеты.

Для примера установим gcc внутри VE 101:

```
root@Knoppix:~# vzctl enter 101
entered into VE 101
Knoppix:/#
Knoppix:/# apt-get install gcc
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  binutils cpp cpp-3.3 gcc-3.3
Suggested packages:
  binutils-doc cpp-doc make manpages-dev autoconf automake libtool flex bison gdb
gcc-doc gcc-3.3-doc
Recommended packages:
  libc-dev libc6-dev
The following NEW packages will be installed:
  binutils cpp cpp-3.3 gcc gcc-3.3
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 5220kB of archives.
After unpacking 13.6MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.freenet.de stable/main binutils 2.15-6 [2221kB]
Get:2 http://ftp.freenet.de stable/main cpp-3.3 1:3.3.5-13 [1393kB]
Get:3 http://ftp.freenet.de stable/main cpp 4:3.3.5-3 [29.6kB]
Get:4 http://ftp.freenet.de stable/main gcc-3.3 1:3.3.5-13 [1570kB]
Get:5 http://ftp.freenet.de stable/main gcc 4:3.3.5-3 [4906B]
Fetched 5220kB in 10s (507kB/s)
Selecting previously deselected package binutils.
(Reading database ... 7436 files and directories currently installed.)
Unpacking binutils (from .../binutils_2.15-6_i386.deb) ...
Selecting previously deselected package cpp-3.3.
Unpacking cpp-3.3 (from .../cpp-3.3_1%3a3.3.5-13_i386.deb) ...
Selecting previously deselected package cpp.
Unpacking cpp (from .../cpp_4%3a3.3.5-3_i386.deb) ...
Selecting previously deselected package gcc-3.3.
Unpacking gcc-3.3 (from .../gcc-3.3_1%3a3.3.5-13_i386.deb) ...
Selecting previously deselected package gcc.
Unpacking gcc (from .../gcc_4%3a3.3.5-3_i386.deb) ...
Setting up binutils (2.15-6) ...

Setting up cpp-3.3 (3.3.5-13) ...
Setting up cpp (3.3.5-3) ...
Setting up gcc-3.3 (3.3.5-13) ...
Setting up gcc (3.3.5-3) ...
```

Knoppix:/# exit
logout
exited from VE 101
root@Knoppix:~#

Ограничение ресурсов

Важным свойством VE является возможность ограничить потребляемые ресурсы: CPU, память, дисковое пространство. Это также делается с помощью **vzctl**. Текущий ограничения и значения использования ресурсов могут быть просмотрены в **/proc/bc/VEID/resources**:

```
root@Knoppix:~# cat /proc/bc/101/resources
        kmemsize        628209    976969   2752512   2936012         0
        lockedpages          0         0        32        32         0
        privvmpages       5238      6885     49152     53575         0
        shmpages          5012      5014      8192      8192         0
        numproc              3        11        65        65         0
        physspages        5084      6020         0 2147483647         0
        vmguarpages          0         0      6144 2147483647         0
        oomguarpages      5084      6020      6144 2147483647         0
        numtcpsock           0         2        80        80         0
        numflock             1         5       100       110         0
        numpty               0         1        16        16         0
        numsiginfo           0         6       256       256         0
        tcpsndbuf            0      4440    319488    524288         0
        tcprcvbuf            0     42180    319488    524288         0
        othersockbuf      2220      6660    132096    336896         0
        dgramrcvbuf          0      2220    132096    132096         0
        numothersock         1         6        80        80         0
        dcachesize           0         0   1048576   1097728         0
        numfile            106       339      2048      2048         0
        numiptent           10        10       128       128         0
root@Knoppix:~#
```

Первая колонка – наименование ресурса, вторая – текущее значение, третья – пиковое значение, 4 и 5 – барьер и ограничение. Последняя – счетчик ошибок.

Если вы видите ненулевые значения в последней колонке, значит ваш VE «голодает». Это основная причина отказа некоторых приложений в VE. В этом случае вы должны увеличить ограничения.

Остановка и уничтожение VE

Остановим и уничтожим VE:

```
root@Knoppix:~# vzctl stop 101
Stopping VE ...
VE was stopped
VE is unmounted
root@Knoppix:~# vzctl destroy 101
Destroying VE private area: /var/lib/vz/private/101
VE private area was destroyed

root@Knoppix:~#
```