

Руководство по созданию и управлению контейнерами на базе OpenVZ

Амет Умеров¹
27 декабря 2014 г.²

¹admin@amet13.name

²Дата последней правки документа

Содержание

1	Введение в виртуализацию	3
1.1	Эмуляция оборудования	4
1.2	Полная виртуализация	5
1.3	Паравиртуализация	6
1.4	Виртуализация уровня операционной системы	6
1.5	OpenVZ — технология виртуализации уровня ОС	7
2	Подготовительные действия	9
2.1	Установка и настройка CentOS	9
2.2	Обновление пакетной базы программ	11
2.3	Настройка времени системы	11
2.4	Установка ядра и утилит OpenVZ	12
2.5	Загрузка шаблонов для гостевых ОС	13
3	Создание и настройка нового сервера VPS	14
3.1	Проверка сети и дискового пространства	14
3.2	Идентификаторы контейнеров	14
3.3	Просмотр списка контейнеров и шаблоны	15
3.4	Конфигурационные файлы	15
3.5	Создание и настройка контейнера	16
3.6	Запуск и вход	18
3.7	Статус VPS	20
3.8	Остановка, перезапуск и удаление контейнера	20
3.9	Запуск команд	21
4	Управление ресурсами	23
4.1	Дисковые параметры	23
4.2	Параметры процессора	24
4.3	Системные параметры	25
5	VPN в контейнере	26
6	Резервное копирование и восстановление	28
6.1	Резервная копия контейнера	28
6.2	Восстановление контейнера из резервной копии	30
7	ploор	31
8	Управление VPS через Web-браузер	33

9 Рекомендации по работе с системой	35
Список литературы	37
A Установка CentOS 6	38

1 Введение в виртуализацию

В данном руководстве описывается установка и настройка системы виртуализации OpenVZ¹ на базе Linux-дистрибутива CentOS 6.

На момент последней правки руководства, последняя версия CentOS, поддерживающая OpenVZ — 6.6.

Для полного понимания информации руководства, необходимо иметь базовые знания о виртуализации, а также иметь навыки работы в командной строке GNU/Linux.

Виртуализация — предоставление наборов вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее изоляцию вычислительных процессов.

Виртуализацию можно использовать в [1]:

- Консолидации серверов (позволяет мигрировать с физических серверов на виртуальные, тем самым увеличивается коэффициент использования аппаратуры, что позволяет существенно сэкономить на аппаратуре, электроэнергии и обслуживании);
- Разработке и тестировании приложений (возможность одновременно запускать несколько различных ОС, это удобно при разработке кроссплатформенного ПО, тем самым значительно повышается качество, скорость разработки и тестирования приложений);
- Бизнесе (использование виртуализации в бизнесе растет с каждым днем и постоянно находятся новые способы применения этой технологии, например, возможность безболезненно сделать снапшот² и быстро восстановить систему в случае сбоя);
- Организации виртуальных рабочих станций (так называемых «тонких клиентов»³).

¹OpenVZ распространяется под лицензией GNU GPL v2.0

²Снапшот (англ. snapshot) — снимок состояния виртуальной машины (ВМ) в определенный момент времени. Сюда входят настройки ВМ, содержимое памяти и дисков

³Тонкий клиент (англ. thin client) — бездисковый компьютер-клиент в сетях с клиент-серверной или терминальной архитектурой, который переносит все или большую часть задач по обработке информации на сервер

Общая схема взаимодействия виртуализации с аппаратурой и программным обеспечением (ПО) представлена на рис. 1.

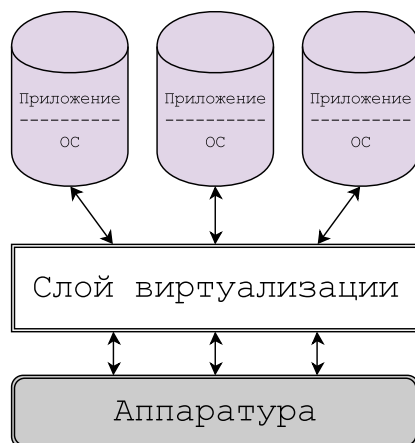


Рис. 1: Схема взаимодействия виртуализации с аппаратурой и ПО

Взаимодействие приложений и операционной системы (ОС) с аппаратным обеспечением осуществляется через абстрагированный слой виртуализации.

Существует несколько подходов организации виртуализации:

- Эмуляция оборудования (QEMU, Bochs, Dynamips);
- Полная виртуализация (KVM, HyperV, VirtualBox);
- Паравиртуализация (Xen, L4, Trango);
- Виртуализация уровня ОС (LXC, OpenVZ, Jails, Solaris Zones).

1.1 Эмуляция оборудования

Является самым сложным и трудоемким методом виртуализации (рис. 2). Однако, большим преимуществом для данного метода является, к примеру, управление неизменной ОС, предназначенной для PowerPC на системе с ARM процессором.

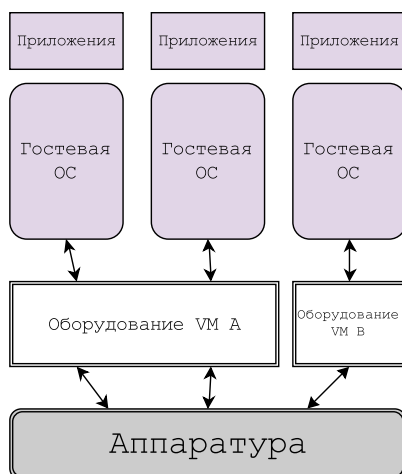


Рис. 2: Эмуляция оборудования моделирует аппаратные средства

1.2 Полная виртуализация

Поверх уже установленной ОС, устанавливается программа-гипервизор¹, которая осуществляет взаимосвязь между гостевыми ОС и хост-компьютером (рис. 3).

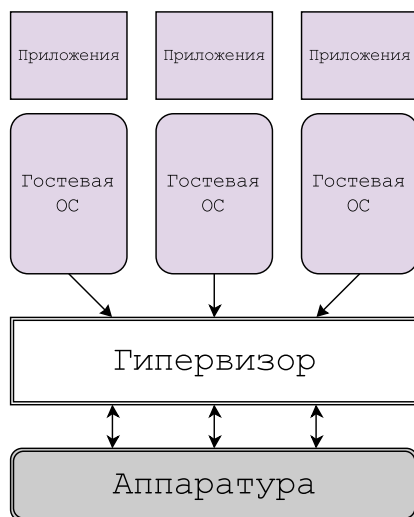


Рис. 3: Полная виртуализация использует гипервизор

¹Гипервизор (англ. hypervisor)— программа или аппаратная схема, позволяющая одновременное, параллельное выполнение нескольких ОС на одном и том же компьютере, обеспечивает изоляцию операционных систем друг от друга

Преимуществом технологии полной виртуализации является установка различных ОС, а недостатком — меньшая производительность, за счет накладных расходов на гипервизор, а также понижение скорости работы с подсистемой ввода/вывода из-за необходимости изоляции.

1.3 Паравиртуализация

Имеет некоторые сходства с полной виртуализацией. В данном методе также используется гипервизор для разделения доступа к аппаратуре, но объединяется код, касающийся виртуализации, в ОС [2] (рис. 4).

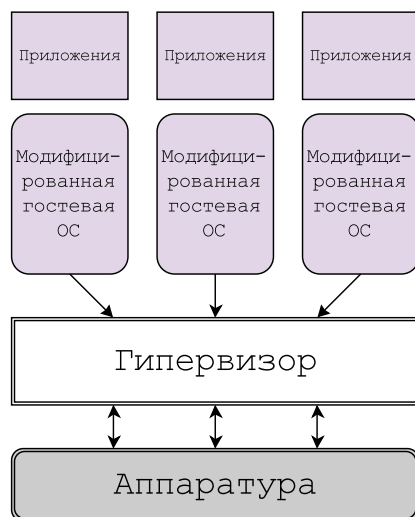


Рис. 4: Паравиртуализация разделяет процесс с гостевой ОС

Недостатком паравиртуализации является необходимость изменения гостевой ОС для гипервизора, однако таким образом гораздо увеличивается производительность.

1.4 Виртуализация уровня операционной системы

Не нуждается в гипервизоре. Для ее работы необходимо модифицированное ядро на хост-системе с набором патчей и утилит для управления контейнерами¹ (рис. 5).

За счет того, что контейнер напрямую взаимодействует с ядром, а не через гипервизор, обеспечивается максимальное быстродействие. Но,

¹Контейнер или VPS/VDS (англ. Virtual Private/Dedicated Server) — виртуальный выделенный сервер, эмулирует работу физического сервера

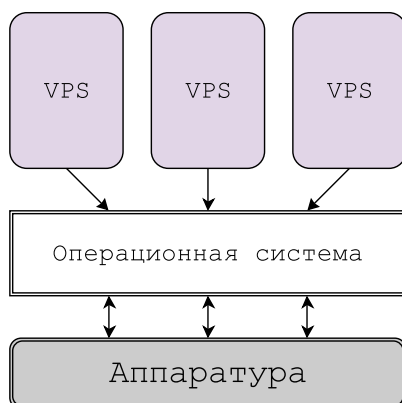


Рис. 5: Виртуализация уровня ОС изолирует серверы

так как для всех контейнеров используется общее ядро, то нет возможности использовать разные ОС в контейнерах.

1.5 OpenVZ — технология виртуализации уровня ОС

Позволяет создавать множество защищенных, изолированных друг от друга виртуальных сред (VE) на одном узле.

Каждый контейнер ведет себя так же, как автономный сервер и имеет собственные файлы, процессы, сеть (IP адреса, правила маршрутизации и т. д.). В отличие от KVM или Xen, OpenVZ использует одно ядро, которое является общим для всех виртуальных сред.

Контейнеры можно разделить на две составляющие:

- Ядро (namespaces, cgroups);
- Пользовательские утилиты (vzctl, vzquota, vldump, ...).

namespaces — пространства полностью изолированных имен, позволяющие безопасно изолировать дисковое пространство, процессы, пользователей, имя компьютера (hostname) и т. д.

cgroups — механизм контроля за ресурсами, позволяющий управлять памятью и распределять процессорное время.

Для удобства управления контейнерами можно использовать консольные утилиты (vzctl), либо управлять ими через Web-браузер (OpenVZ Web Panel, ISP VEmanager).

Проведенные тестирования показывают, что OpenVZ является одним из наиболее актуальных решений на рынке виртуализации, так как показывает внушительные результаты в различных тестированиях [3]. Один из таких тестов [4] приведен на рис. 6.

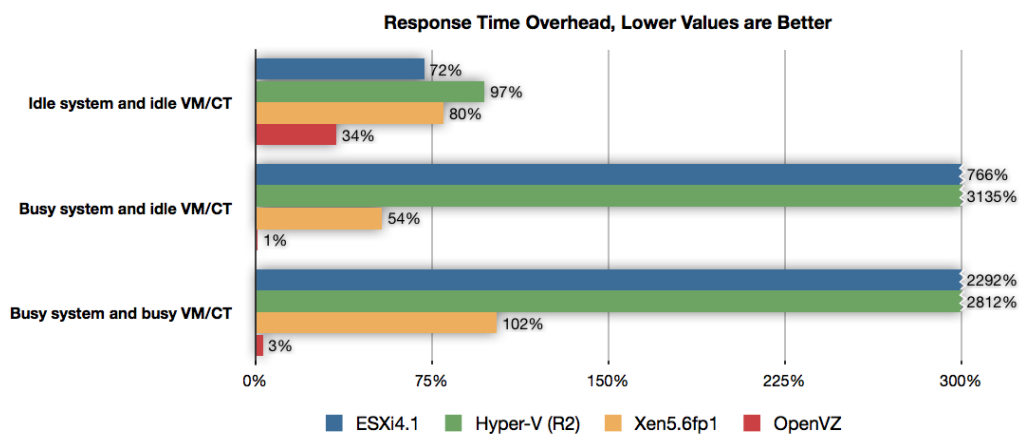


Рис. 6: Время отклика системы (меньше — лучше)

На графике можно наблюдать, что проводилось три теста с нагрузкой на систему и виртуальную машину (ВМ), без нагрузки на систему и ВМ, с нагрузкой на ВМ и без нагрузки на систему. Во всех трех тестах OpenVZ показал результаты наименьшего времени отклика. В этих тестах ESXi и Нурег-V показывают оверхед¹ 700—3000%, когда у OpenVZ всего 1—3%.

¹Оверхед (англ. overhead) — неизбежные накладные расходы памяти или процессорного времени

2 Подготовительные действия

Для организации виртуального пространства, в первую очередь необходимо установить операционную систему на хост-компьютер. В качестве хост-компьютера может выступать как сервер, так и настольный компьютер.

В данном руководстве описывается установка и настройка ОС CentOS 6¹. CentOS основан на коммерческом дистрибутиве Red Hat Enterprise Linux (RHEL). RHEL состоит из свободного ПО с открытым кодом, однако доступен в виде дисков с бинарными пакетами только для платных подписчиков. Red Hat предоставляют исходные коды системы, так как это предполагается лицензией GNU/GPL. Разработчики CentOS используют данный исходный код для создания окончательного продукта, очень близкого к RHEL и доступного для скачивания. Срок поддержки CentOS — десять лет.

2.1 Установка и настройка CentOS

Программа установки CentOS проста в использовании и локализована почти на все языки мира. В качестве системного, рекомендуется использовать язык по умолчанию (english).

Во время установки, помимо выбора языка необходимо указать пароль суперпользователя (root), например `p@ssw0rd` и имя узла, например `centos.openvz`.

На жестком диске нужно создать как минимум 3 раздела:

- `/` — не менее 6G (ext4) — корневой раздел;
- `swap` — 1G² — раздел подкачки;
- `/vz` — все остальное дисковое пространство (ext4).

Пошаговая установка CentOS 6 описана в прил. А.

После установки ОС, необходимо перезагрузиться.

Первый вход в систему осуществляется с учетной записи суперпользователя. Логин — `root`, пароль³ — `p@ssw0rd`.

После входа в систему, пользователю будет доступен консольный интерфейс интерпретатора командной строки (bash). С помощью ввода команд в интерпретаторе, осуществляется работа в ОС GNU/Linux.

¹Образы для скачивания: <http://mirror.yandex.ru/centos/6.6/isos/>

²Размер swap должен быть равен примерно половине объема оперативной памяти

³Во время ввода пароля, число его символов не отображается, это сделано в целях безопасности

Строка ввода команд выглядит примерно так:

```
[root@centos ~]#
```

где:

- `root` — имя текущего пользователя;
- `centos` — имя компьютера (задавали при установке);
- `~` — текущий каталог¹, в котором находится пользователь.

Символ `#` означает, что в данный момент вход осуществлен пользователем `root`.

Для безопасной работы с GNU/Linux нужно иметь как минимум одного локального пользователя отличного от `root`. Пусть этим пользователем будет `stud` с паролем `$tud`:

```
# adduser stud
# passwd stud
New password: $tud
```

Теперь, во время входа в систему, можно указать пользователя `stud` и пароль `$tud`:

```
# exit
centos login: stud
Password: $tud
Last login: Mon Jan 01 12:12:12 on tty1
[stud@centos ~]$
```

Чтобы переключиться в системе с пользователя `stud` на `root` можно выполнить команду:

```
$ su -
Password: p@ssw0rd
[root@centos ~]#
```

По умолчанию в CentOS после перезагрузки сетевые интерфейсы не включаются. Нужно сделать так, чтобы при загрузке системы интерфейсы поднимались сами:

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
ONBOOT="yes"
```

```
# ifup eth0
```

¹Для просмотра имени текущего каталога используется команда `pwd`

2.2 Обновление пакетной базы программ

Для установки, удаления и обновления программ, в CentOS существует утилита `yum`. На сервере важно всегда обновлять программное обеспечение, так как в новых версиях не только могут добавлять новые возможности, но и исправлять уязвимости. Указанная ниже команда обновляет все существующие в системе пакеты:

```
# yum update
```

Для удобства работы в ОС рекомендуется установить следующее ПО:

- `vim` (текстовый редактор);
- `wget` (утилита для скачивания файлов);
- `nslookup` (утилита для работы с DNS);
- `man`¹ (руководства к программам).

```
# yum install vim wget bind-utils man
```

Чтобы установить `htop` (утилиту системного мониторинга), нужно подключить дополнительный репозиторий `rpmforge`:

```
# cd /tmp
# wget http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el6.rf.x86_64.rpm
# rpm -Uvh rpmforge-release*.rf.x86_64.rpm
# yum install htop
```

2.3 Настройка времени системы

Для сервера очень важно, чтобы было установлено правильное время.

Чтобы синхронизировать время с интернетом необходимо установить пакет `ntp`:

```
# yum install ntp
# service ntpd start
Starting ntpd: [ OK ]
# ntpdate -bs pool.ntp.org
# date
Mon Jan 01 12:12:12 MSK 2014
```

¹Для того, чтобы узнать как правильно пользоваться командой `man` можно выполнить команду `man man`

2.4 Установка ядра и утилит OpenVZ

Для установки ядра и утилит OpenVZ нужно добавить репозиторий и импортировать ключ:

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

Установка `vzkernel` (ядра OpenVZ):

```
# yum install vzkernel
```

Настраиваем параметры ядра¹, необходимые для работы OpenVZ:

```
# vim /etc/sysctl.conf
net.ipv4.ip_forward = 1
net.ipv4.conf.default.proxy_arp = 0
net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.rp_filter = 1
kernel.sysrq = 1
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
```

Отключаем SELinux²:

```
# echo "SELINUX=disabled" > /etc/sysconfig/selinux
```

Устанавливаем инструменты, необходимые для работы с OpenVZ:

```
# yum install vzctl vzquota ploop
```

По умолчанию в OpenVZ настроен IPTables который блокирует некоторые соединения.

Очищаем все правила IPTables, оставив в файле строки:

```
# vim /etc/sysconfig/iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
```

```
# service iptables restart
```

¹В `vzctl` версии 4.4 и выше, настройки параметров ядра не обязательны

²SELinux (англ. Security-Enhanced Linux) — реализация системы принудительного контроля доступа, которая может работать параллельно с классической избирательной системой контроля доступа

Для применения новых настроек нужно перезагрузиться:

```
# reboot
```

Во время перезагрузки в меню загрузчика GRUB появится пункт OpenVZ (рис. 7). С этого ядра и нужно загружаться.

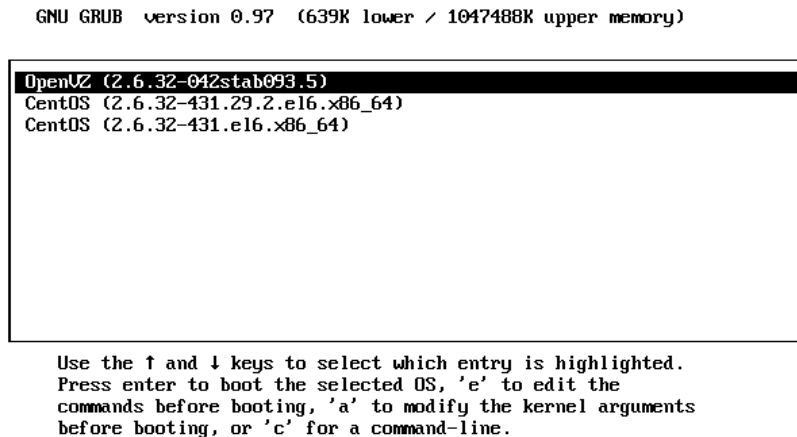


Рис. 7: Меню загрузчика GRUB

2.5 Загрузка шаблонов для гостевых ОС

На основе файлов шаблонов OpenVZ, будут работать гостевые операционные системы.

Полный список доступных для загрузки шаблонов¹ можно посмотреть командой:

```
# vztmpl-dl --list-all
```

Загрузим шаблоны для Debian, Ubuntu, CentOS и openSUSE:

```
# vztmpl-dl debian-7.0-x86_64-minimal \  
> ubuntu-14.04-x86_64-minimal \  
> centos-7-x86_64-minimal \  
> suse-13.1-x86_64-minimal \  
> centos-6-x86_64-minimal \  
>
```

¹Шаблон — согласно определению, это ряд пакетов от некоторого распределения Linux, используемых для заселения VPS. Шаблон ОС состоит из системных программ, библиотек, и скриптов, необходимых, так же как основные приложения и утилиты

3 Создание и настройка нового сервера VPS

3.1 Проверка сети и дискового пространства

После загрузки в `vzkernel`, первым делом необходимо проверить настройки сети:

```
# ifconfig | grep "eth\|lo\|venet" -A 1
eth0      Link encap:Ethernet  HWaddr 08:00:27:14:41:DE
          inet addr:192.168.0.100  Bcast:192.168.0.255
          Mask:255.255.255.0
--
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
--
venet0    Link encap:UNSPEC  HWaddr
          00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet6 addr: fe80::1/128 Scope:Link
```

Должно быть доступно три сетевых интерфейса:

- `eth0` (192.168.0.100) — интерфейс реальной сетевой карты;
- `lo` (127.0.0.1) — виртуальный интерфейс локальная «петля»;
- `venet0` — виртуальный сетевой интерфейс для OpenVZ.

Проверим свободное место на диске:

```
# df -h | grep vz
/dev/sda2      32G  1.1G   30G   4% /vz
```

По этому примеру видно, что всего доступно 32G дискового пространства на разделе `/vz`, из них занято 1.1G, скачанными ранее шаблонами ОС. С учетом того, что на один контейнер будем тратить не более 1G дискового пространства, то 30G должно хватить.

3.2 Идентификаторы контейнеров

При создании, каждый контейнер имеет идентификатор (CTID)¹:

- CTID с номером 0 — это хост-компьютер;
- CTID от 1 до 100 резервируются OpenVZ для внутренних нужд.

¹CTID — ConTainer IDentificator

CTID должен быть уникальным для каждого контейнера. Хорошей практикой является создание контейнера с CTID от 101 до 999.

Существует также схема присвоения идентификаторов по IP адресам. Например, для адреса 10.0.2.1, CTID = 102, для 192.168.123.33, CTID = 33123 и т. д. [5]

3.3 Просмотр списка контейнеров и шаблоны

Посмотрим список созданных контейнеров. Так как, пока VPS не созданы, то можно увидеть такое:

```
# vzlist -a
```

CTID	NPROC	STATUS	IP_ADDR	HOSTNAME
------	-------	--------	---------	----------

Для создания контейнера нужно использовать один из ранее скачанных шаблонов. Проверим, какие шаблоны доступны на хост-компьютере:

```
# vztmpl-dl --list-local
centos-6-x86_64-minimal
centos-7-x86_64-minimal
debian-7.0-x86_64-minimal
suse-13.1-x86_64-minimal
ubuntu-14.04-x86_64-minimal
```

3.4 Конфигурационные файлы

Каждый контейнер имеет свой конфигурационный файл (далее «конфиг»), который хранится в каталоге /etc/sysconfig/vz-scripts/.

Именуются конфиги по CTID контейнера. Например, для контейнера с CTID = 101, конфиг будет называться 101.conf.

При создании контейнера можно использовать типовую конфигурацию для VPS. Типовые файлы конфигураций находятся в том же каталоге /etc/sysconfig/vz-scripts/:

```
# ls -l /etc/sysconfig/vz-scripts/ | grep sample
ve-basic.conf-sample
ve-custom.conf-sample
ve-light.conf-sample
ve-vswap-1024m.conf-sample
ve-vswap-1g.conf-sample
ve-vswap-256m.conf-sample
ve-vswap-2g.conf-sample
ve-vswap-4g.conf-sample
ve-vswap-512m.conf-sample
```

В этих конфигурационных файлах описаны контрольные параметры ресурсов, выделенное дисковое пространство, оперативная память и т. д.

Например, при использовании конфига `ve-vswap-1g`, создается VPS с дисковым пространством 2G, оперативной памятью 1G и swap 2G. Это удобно, так как существует возможность создавать свои конфигурационные файлы для различных вариаций VPS.

Создадим свой конфигурационный файл, на базе уже существующего (`vswap-256m`). Исправим в нем только значения `DISKSPACE`, `PHYSPAGES` и `SWAPPAGES`:

```
# cp /etc/vz/conf/ve-vswap-256m.conf-sample /etc/vz/conf/ve-
  -custom.conf-sample
# vim /etc/vz/conf/ve-custom.conf-sample
DISKSPACE="1G:1.1G"
PHYSPAGES="0:128M"
SWAPPAGES="0:128M"
```

Таким образом, при использовании этого конфигурационного файла, будет создаваться контейнер, которому будет доступен 1G выделенного дискового пространства, 128M оперативной памяти и 128M swap.

В дальнейшем, при создании контейнеров будем использовать конфигурационный файл `custom`.

3.5 Создание и настройка контейнера

Для создания контейнера необходимо ввести команду:

```
# vzctl create 101 --ostemplate debian-7.0-x86_64-minimal
  --config custom
```

где:

- 101 — CTID контейнера;
- `debian-7.0-x86_64-minimal` — шаблон ОС;
- `custom` — желаемый шаблон конфигурационного файла.

После нажатия клавиши **Enter** начинается процесс создания VPS. По времени процедура может занимать несколько десятков секунд.

Проверим правильность создания VPS:

```
# vzlist -a
      CTID      NPROC STATUS      IP_ADDR      HOSTNAME
      101        -  stopped      -            -
```


Можно увидеть, что создан контейнер с CTID = 101, сейчас он не включен.

Если же при создании контейнера не указывать желаемый шаблон и файл конфигурации, то OpenVZ будет использовать шаблон и конфигурационный файл по умолчанию. Конфиг, в котором указаны директивы по умолчанию имеет имя: `/etc/sysconfig/vz`. По умолчанию, используется шаблон `centos-6-x86` и конфигурационный файл `vswap-256m`.

Так как планируется создание небольшого количества VPS, основываясь на одном и том же конфиге, то исправим¹ эти значения на нужные:

```
# vim /etc/sysconfig/vz
#CONFIGFILE="vswap-256m"
CONFIGFILE="custom"
#DEF_OSTEMPLATE="centos-6-x86"
DEF_OSTEMPLATE="debian-7.0-x86_64-minimal"
```

Теперь, при создании VPS достаточно указать только CTID контейнера, например:

```
# vzctl create 101
```

Будет создан контейнер на базе `debian-7.0x86_64-minimal`, значения системных параметров будут взяты с конфига `custom`.

Контейнер создан, его можно запускать. Но перед первым запуском необходимо установить его IP адрес, `hostname`, указать DNS сервер и задать пароль суперпользователя.

Для настройки VPS используется команда `vzctl set`.

Для того, чтобы контейнер запускался при старте хост-компьютера (например после перезагрузки), необходимо использовать команду:

```
# vzctl set 101 --onboot yes --save
CT configuration saved to /etc/vz/conf/101.conf
```

При использовании ключа `--save` в `vzctl set`, сохраняются параметры контейнера в соответствующий конфигурационный файл.

Аналогично можно задать `hostname`:

```
# vzctl set 101 --hostname stud1 --save
CT configuration saved to /etc/vz/conf/101.conf
```

Установка IP адреса:

```
# vzctl set 101 --ipadd 192.168.0.101 --save
CT configuration saved to /etc/vz/conf/101.conf
```

¹В файле `/etc/sysconfig/vz` (и многих других) с символа `#` начинается комментарий

Адрес DNS сервера (в большинстве случаев¹ адрес DNS совпадает с адресом хост-компьютера, поэтому можно вместо адреса указать параметр `inherit`):

```
# vzctl set 101 --nameserver inherit --save
CT configuration saved to /etc/vz/conf/101.conf
```

Установка пароля суперпользователя:

```
# vzctl set 101 --userpasswd root:p@ssw0rd
Starting container...
Opening delta /vz/private/101/root.hdd/root.hdd
Adding delta dev=/dev/ploop37965 img=/vz/private/101/root.
hdd/root.hdd (rw)
Mounting /dev/ploop37965p1 at /vz/root/101 fstype=ext4 data
='balloon_ino=12,'
Container is mounted
Container start in progress...
Killing container ...
Container was stopped
Unmounting file system at /vz/root/101
Unmounting device /dev/ploop37965
Container is unmounted
```

Пароль будет установлен в VPS, в файл `/etc/shadow` и не будет сохранен в конфигурационный файл контейнера. Если же пароль будет утерян или забыт, то можно будет просто задать новый.

3.6 Запуск и вход

После настроек нового контейнера, его можно запустить:

```
# vzctl start 101
Starting container...
Opening delta /vz/private/101/root.hdd/root.hdd
Adding delta dev=/dev/ploop37965 img=/vz/private/101/root.
hdd/root.hdd (rw)
Mounting /dev/ploop37965p1 at /vz/root/101 fstype=ext4 data
='balloon_ino=12,'
Container is mounted
Adding IP address(es): 192.168.0.101
Setting CPU units: 1000
Container start in progress...
```

¹Если же нужно явно указать адрес DNS сервера, то вместо `inherit` можно указать IP адрес, например `192.168.0.1`

Для того, чтобы выполнить команду внутри контейнера существует команда `vzctl exec`. Подробнее об этой команде позже.

Проверяем сетевые интерфейсы внутри гостевой ОС:

```
# vzctl exec 101 ifconfig | grep "lo\|venet" -A 1
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
--
venet0      Link encap:UNSPEC  HWaddr
            00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
            inet addr:127.0.0.2  P-t-P:127.0.0.2  Bcast
            :0.0.0.0  Mask:255.255.255.255
--
venet0:0    Link encap:UNSPEC  HWaddr
            00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
            inet addr:192.168.0.101  P-t-P:192.168.0.101
            Bcast:192.168.0.101  Mask:255.255.255.255
```

Должны присутствовать сетевые интерфейсы:

- `lo` (127.0.0.1);
- `venet0` (127.0.0.2);
- `venet0:0` (192.168.0.101).

Если сеть в порядке, то можно соединиться к контейнеру по SSH с хост-компьютера:

```
# ssh root@192.168.0.101
root@192.168.0.101's password: p@ssw0rd
```

Вход в контейнер напрямую с хост-компьютера осуществляется командой `vzctl enter`:

```
# vzctl enter 101
entered into CT 101
root@stud1:/#
```

Выход из контейнера:

```
root@stud1:/# exit
logout
exited from CT 101
```

3.7 Статус VPS

Для того, чтобы узнать статус контейнера, используется команда:

```
# vzctl status 101
CTID 101 exist mounted running
```

По выводу команды можно видеть, что контейнер с CTID = 101 существует, смонтирован и запущен.

Команда `vzlist -a` выводит список всех существующих в системе контейнеров. Рассмотрим подробно вывод команды `vzlist -a`:

```
# vzlist -a
```

CTID	NPROC	STATUS	IP_ADDR	HOSTNAME
101	15	running	192.168.0.101	stud1
102	-	stopped	192.168.0.102	stud2

где:

- CTID — ID контейнера;
- NPROC — число запущенных процессов в контейнере;
- STATUS — состояние контейнера (запущен/не запущен);
- IP_ADDR — IP адрес;
- HOSTNAME — имя контейнера.

3.8 Остановка, перезапуск и удаление контейнера

Для остановки контейнера используется команда:

```
# vzctl stop 101
```

Для полной остановки контейнера, системе требуется немного времени (примерно 2 минуты).

Иногда нужно выключить VPS как можно быстрее, например, если контейнер был подвержен взлому. Для того чтобы срочно выключить VPS, нужно использовать ключ `--fast`:

```
# vzctl stop 101 --fast
Killing container ...
Container was stopped
Unmounting file system at /vz/root/101
Unmounting device /dev/ploop37965
Container is unmounted
```

Для перезапуска контейнера можно использовать команду:

```
# vzctl restart 101
Restarting container
Stopping container ...
Container was stopped
Unmounting file system at /vz/root/101
Unmounting device /dev/ploop37965
Container is unmounted
Starting container...
Opening delta /vz/private/101/root.hdd/root.hdd
Adding delta dev=/dev/ploop37965 img=/vz/private/101/root.
hdd/root.hdd (rw)
Mounting /dev/ploop37965p1 at /vz/root/101 fstype=ext4 data
='balloon_ino=12,'
Container is mounted
Adding IP address(es): 192.168.0.101
Setting CPU units: 1000
Container start in progress...
```

А для того чтобы удалить контейнер, его нужно сначала остановить:

```
# vzctl stop 101
Stopping container ...
Container was stopped
Unmounting file system at /vz/root/101
Unmounting device /dev/ploop37965
Container is unmounted
```

Для удаления используется команда:

```
# vzctl destroy 101
CTID 101 deleted unmounted down
```

Команда выполняет удаление частной области сервера и переименовывает файл конфигурации, дописывая к нему `.destroyed`¹.

3.9 Запуск команд

Как уже было сказано выше, для запуска команд в контейнере используется команда:

```
# vzctl exec 101 command
```

¹Например, после удаления контейнера с CTID = 101, конфиг стал называться `/etc/vz/conf/101.conf.destroyed`

Например, для того, чтобы соединиться к VPS по SSH¹, нужно сначала включить SSH:

```
# vzctl exec 101 service ssh start
Starting OpenBSD Secure Shell server: sshd.
```

Теперь можно соединиться к контейнеру по SSH:

```
# ssh root@192.168.0.101
root@192.168.0.101's password: p@ssw0rd
root@stud1:/#
```

Иногда бывает нужно выполнить команду на нескольких VPS. Для этого можно использовать команду:

```
# for i in `vzlist -o veid -H`; do \
> echo "VPS $i"; vzctl exec $i command; done
```

Например, можно узнать, сколько времени работают все запущенные контейнеры:

```
# for i in `vzlist -o veid -H`; do \
> echo "VPS $i"; vzctl exec $i uptime; done
VPS 101
 05:45:01 up 2 min, 0 users, load average: 0.01, 0.02, 0.03
VPS 102
 05:46:01 up 1 min, 0 users, load average: 0.04, 0.05, 0.06
```

Или узнать системную информацию о дистрибутивах в контейнерах:

```
# for i in `vzlist -o veid -H`; do \
> echo "VPS $i"; vzctl exec $i uname -rv; done
VPS 101
2.6.32-042stab093.5 #1 SMP Wed Jan 01 12:12:12 MSK 2014
VPS 102
2.6.32-042stab093.5 #1 SMP Wed Jan 01 12:12:12 MSK 2014
```

¹SSH позволяет безопасно передавать в незащищенной среде практически любой сетевой протокол. Таким образом, можно не только удаленно работать на компьютере через командную оболочку, но и передавать по зашифрованному каналу звуковой поток или видео. Также SSH может использовать сжатие передаваемых данных для последующего их шифрования

4 Управление ресурсами

Системный оператор контролирует, доступные частному серверу ресурсы, с помощью набора параметров управления ресурсами. Все эти параметры можно редактировать в файлах шаблонов, в каталоге `/etc/sysconfig/vz-scripts/`. Их можно установить вручную, редактируя соответствующие конфиги или используя утилиты OpenVZ.

Параметры контроля ресурсов условно разделяют на три группы:

- Дисковые (управление квотами диска, фрагментацией);
- Процессорные (распределение процессорного времени);
- Системные (сеть, память).

4.1 Дисковые параметры

Администратор OpenVZ сервера может установить дисковые квоты, в терминах дискового пространства и количества inodes, число которых примерно равно количеству файлов. Это первый уровень дисковой квоты. В дополнение к этому, администратор может использовать обычные утилиты внутри окружения, для настроек стандартных дисковых квот UNIX для пользователей и групп.

Основные параметры:

- `DISKSPACE` — общий размер дискового пространства;
- `DISKINODES` — общее число дисковых inodes¹;
- `QUOTATIME` — время (в секундах) на которое VPS может превысить значение `soft` предела.

Первые два параметра записываются в виде:

```
COMMAND="softlimit:hardlimit"
```

где:

- `COMMAND` — команда (`DISKSPACE` или `DISKINODES`);

¹inode (индексный дескриптор) — структура данных в традиционных для ОС UNIX файловых системах, таких как UFS. В этой структуре хранится метainформация о стандартных файлах, каталогах или других объектах файловой системы, кроме непосредственно данных и имени

- **softlimit** — значение которое превышать нежелательно, после пересечения этого предела наступает grace период, по истечении которого, дисковое пространство или inodes прекратят свое существование;
- **hardlimit** — значение которое превысить нельзя.

Например, запись:

```
DISKSPACE="1G:1.1G"  
DISKINODES="100000:110000"  
QUOTATIME="600"
```

означает, что задается **softlimit** для дискового пространства равным 1G и **hardlimit** равный 1.1G, то же самое с inode 100000 и 110000 соответственно.

Если размер занятого дискового пространства или inodes будет выше **softlimit**, то в течении 600 сек (10 мин), в случае не освобождения дискового пространства или inodes, они прекратят свое существование.

4.2 Параметры процессора

Планировщик процессора в OpenVZ также двухуровневый. На первом уровне планировщик решает, какому контейнеру дать квант процессорного времени, базируясь на значении параметра **CPUUNITS** для VPS. На втором уровне стандартный планировщик GNU/Linux решает, какому процессу в выбранном контейнере дать квант времени, базируясь на стандартных приоритетах процесса.

Основные параметры:

- **VE0CPUUNITS** — целое число, определяющее минимальное количество процессорного времени, доступное для хост-компьютера;
- **CPULIMIT** — верхний лимит процессорного времени в процентах;
- **CPUUNITS** — гарантируемое минимальное количество времени процессора, которое получит соответствующий VPS.

Утилиты контроля ресурсов процессора, гарантируют любому VPS количество времени центрального процессора, которое собственно и получает этот VPS. При этом контейнер может потреблять больше времени, чем определено этой величиной, если нет другого конкурирующего с ним за время CPU сервера.

4.3 Системные параметры

В терминах OpenVZ лимиты и гарантии ресурсов называются User Beancounters (UBC). Всего существует около 20 UBC, контролирующих почти все возможные ресурсы системы. Каждый UBC имеет свою опцию в команде `vzctl`, а также строку в конфигурационном файле `/proc/user_beancounters`, с помощью которого можно узнать о текущем количестве выделенных ресурсов и определить их нехватку.

Файл представляет собой таблицу, каждая строка которой содержит информацию об одном ресурсе, а колонки отражают следующие данные:

- `uid` — идентификатор контейнера;
- `resource` — имя ресурса;
- `held` — текущая утилизация ресурса;
- `maxheld` — максимальный уровень утилизации ресурса за все время работы контейнера;
- `barrier` — максимальный уровень утилизации ресурсов, который может быть временно превышен;
- `limit` — жесткое ограничение утилизации ресурса, которое никогда не может быть превышено;
- `failcnt` — счетчик отказов, который увеличивается каждый раз, когда контейнер делает запрос ресурсов сверх своего лимита.

Не обязательно разбираться во всех тонкостях системы подсчета ресурсов OpenVZ, чтобы эффективно управлять контейнерами. Достаточно время от времени поглядывать на значение колонки `failcnt` и, если оно оказывается больше нуля, начинать предпринимать меры либо по оптимизации исполняемого в рамках контейнера ПО, либо по увеличению количества выделяемых контейнеру ресурсов [6].

Начиная с версий ядра RHEL 6 042stab04x, появилась поддержка vSwap [7]. Теперь не нужно высчитывать UBC лимиты, достаточно при создании гостевой системы указать всего лишь `PHYSPAGES` и `SWAPPAGES`.

Начиная с ядра 042stab068.8 появилась возможность ограничивать использование контейнерами дискового кэша.

Более подробную информацию про User Beancounters можно получить по адресам: <http://kb.sp.parallels.com/ru/112807> и <http://openvz.org/UBC>.

5 VPN в контейнере

Технология VPN¹ позволяет устанавливать безопасное сетевое соединение между компьютерами. Для того чтобы VPN работала в контейнере, необходимо разрешить использование TUN/TAP устройств для контейнера.



Рис. 8: Схема работы VPN

Прежде чем запускать контейнер нужно убедиться, что модуль TUN загружен:

```
# lsmod | grep tun
```

В случае, если он не загружен, загрузить его можно следующей командой:

```
# modprobe tun
```

```
# lsmod | grep tun
```

```
tun      1957      0
```

¹VPN (англ. Virtual Private Network) — обобщенное название технологий, позволяющих обеспечить одно или несколько сетевых соединений поверх другой сети (например, Интернет)

Разрешаем использовать устройство TUN контейнеру:

```
# vzctl set 101 --devnodes net/tun:rw --save
CT configuration saved to /etc/vz/conf/101.conf
```

```
# vzctl set 101 --devices c:10:200:rw --save
CT configuration saved to /etc/vz/conf/101.conf
```

```
# vzctl set 101 --capability net_admin:on --save
CT configuration saved to /etc/vz/conf/101.conf
```

Запускаем контейнер:

```
# vzctl start 101
Starting container...
Opening delta /vz/private/101/root.hdd/root.hdd
Adding delta dev=/dev/ploop17343 img=/vz/private/101/root.
hdd/root.hdd (rw)
Mounting /dev/ploop17343p1 at /vz/root/101 fstype=ext4 data
='balloon_ino=12,'
Container is mounted
Adding IP address(es): 192.168.0.101
Setting CPU units: 1000
Setting devices
Container start in progress...
```

Создаем в контейнере собственное устройство TUN:

```
# 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
```

На этом настройка устройства TUN окончена. Далее необходимо установить ПО для работы с VPN. Например одну из программ:

- tinc (<http://tinc-vpn.org>);
- OpenVPN (<http://openvpn.net>);
- VTun (<http://vtun.sourceforge.net>).

Для установки сервера OpenVPN, можно обратиться к официальной документации, расположенной по адресу: <http://openvpn.net/index.php/open-source/documentation.html>

6 Резервное копирование и восстановление

Утилитами `vzdump` и `vzrestore` можно осуществлять резервное копирование и восстановление контейнеров.

Установка `vzdump` и необходимых зависимостей:

```
# yum install cstream perl-LockFile-Simple
# rpm -ivh "http://ftp.openvz.org/contrib/utils/vzdump/
vzdump-1.2-4.noarch.rpm"
Retrieving http://ftp.openvz.org/contrib/utils/vzdump/
vzdump-1.2-4.noarch.rpm
Preparing... ##### [100%]
1:vzdump ##### [100%]
```

6.1 Резервная копия контейнера

Резервная копия контейнера создается командой:

```
# vzdump 101
```

где 101 — CTID контейнера.

По умолчанию, копия VPS сохраняется в каталоге `/vz/dump/`. Размер копии из примера равен 281M:

```
# ls -lh /vz/dump/vzdump-openvz-*.tar
-rw-r--r-- 1 root root 281M Nov  1 14:02 /vz/dump/vzdump-
openvz-101-2014_01_01-12_12_12.tar
```

У `vzdump` есть много параметров, позволяющих осуществлять гибкое резервное копирование¹.

Создадим резервную копию того же контейнера (CTID = 101), но с использованием дополнительных параметров:

```
# vzdump --suspend --compress --dumpdir /root/ --exclude-
path /tmp/ 101
INFO: starting new backup job: vzdump --suspend --compress
--dumpdir /root/ --exclude-path /tmp/ 101
INFO: Starting Backup of VM 101 (openvz)
INFO: CTID 101 exist mounted running
INFO: status = CTID 101 exist mounted running
...
INFO: archive file size: 85MB
INFO: Finished Backup of VM 101 (00:00:52)
INFO: Backup job finished successfully
```

¹Подробнее о параметрах можно узнать в `man vzdump`

где:

- `--suspend` — приостановить контейнер;
- `--compress` — сжимать архив в формате `.tgz`;
- `--dumpdir` — сохранить архив в указанный следующим параметром каталог;
- `--exclude-path`¹ — исключить, указанный следующим параметром каталог, из резервной копии.

Можно заметить, что в сжатом виде архив весит меньше (86М):

```
# ls -lh /root/vzdump-openvz-101-2014_01_01-12_12_13.tgz
-rw-r--r-- 1 root root 86M Nov  1 14:25 /root/vzdump-openvz-101-2014_01_01-12_12_13.tgz
```

Результат операции резервного копирования можно просмотреть в лог-файле, который находится в том же каталоге, что и архив:

```
# tail -4 /root/vzdump-openvz-101-2014_01_01-12_12_13.log
Nov 01 14:25:23 INFO: tar: Exiting with failure status due
to previous errors
Nov 01 14:25:23 INFO: archive file size: 85MB
Nov 01 14:25:23 INFO: delete old backup '/root/vzdump-
openvz-101-2014_01_01-12_12_13.tgz'
Nov 01 14:25:23 INFO: Finished Backup of VM 101 (00:00:52)
```

Чтобы скопировать полученный архив на удаленный сервер по безопасному SSH-соединению, можно воспользоваться утилитой `scp`:

```
# scp /root/vzdump-openvz-101-2014_01_01-12_12_13.tgz
root@backupserver:/backups
root@backupserver's password: b@ckupp@$$wd
vzdump-openvz-101-2014_01_01-12_12_13.tgz 100% 86MB 17.1
MB/s 00:05
```

На удаленном сервере можно проверить, скопировался ли архив:

```
root@backupserver:/# ls -lh /backups/
total 86M
-rw-r--r-- 1 root root 86M Nov  1 08:00 vzdump-openvz-
-101-2014_01_01-12_12_13.tgz
```

¹В версии `vzdump-1.2-4` частично не работает параметр `--exclude-path`

6.2 Восстановление контейнера из резервной копии

Командой `vzrestore` можно восстановить контейнер с архива. `vzrestore` принимает всего 2 параметра: файл архива и CTID контейнера, в который будет разворачиваться архив.

Создадим контейнер с CTID = 103, содержащий резервную копию контейнера с CTID = 101:

```
# vzrestore /vz/dump/vzdump-openvz-101-2014_01_01-12_12_12.
tar 103
INFO: restore openvz backup '/vz/dump/vzdump-openvz
-101-2014_01_01-12_12_12.tar' using ID 103
INFO: extracting archive 'vzdump-openvz-101-2014_01_01-12
_12_12.tar'
INFO: Total bytes read: 86611520 (86MiB, 26MiB/s)
INFO: extracting configuration to '/etc/vz/conf/103.conf'
INFO: restore openvz backup '/vz/dump/vzdump-openvz
-101-2014_01_01-12_12_12.tar' successful
```

Проверяем новый контейнер:

```
# vzlist 103
```

CTID	NPROC	STATUS	IP_ADDR	HOSTNAME
103	-	stopped	192.168.0.101	stud1

Зададим параметры для нового контейнера:

```
# vzctl set 103 --hostname stud3 --save
CT configuration saved to /etc/vz/conf/103.conf
```

```
# vzctl set 103 --ipdel 192.168.0.101 --save
CT configuration saved to /etc/vz/conf/103.conf
```

```
# vzctl set 103 --ipadd 192.168.0.103 --save
CT configuration saved to /etc/vz/conf/103.conf
```

```
# vzctl start 103
Starting container...
Opening delta /vz/private/103/root.hdd/root.hdd
Adding delta dev=/dev/ploop12539 img=/vz/private/103/root.
hdd/root.hdd (rw)
Mounting /dev/ploop12539p1 at /vz/root/103 fstype=ext4 data
='balloon_ino=12,'
Container is mounted
Adding IP address(es): 192.168.0.103
Setting CPU units: 1000
Setting devices
Container start in progress...
```

7 ploop

Для работы OpenVZ с файлами контейнера, существует два метода:

- `simfs` (каталоги и файлы в файловой системе хост-компьютера);
- `ploop` (отдельный файл для каждого контейнера).

`simfs` уже осталась в прошлом, в последнее время ее вытеснил `ploop`, который готов для использования в production¹.

По умолчанию в OpenVZ используется `ploop`. Настраивается это в конфигурационном файле `/etc/vz/vz.conf`:

```
# cat /etc/vz/vz.conf | grep VE_LAYOUT
VE_LAYOUT=ploop
```

Основные преимущества `ploop` [8]:

- Поддержка корректной и надежной изоляции пользователей друг от друга;
- Простое резервное копирование;
- Журнал файловой системы больше не является узким местом;
- Живая миграция;
- Поддержка различных типов хранения данных.

В `ploop` игнорируются параметры `DISKQUOTA`, `DISKINODES`, `QUOTATIME`. Параметр `DISKSPACE` не игнорируется.

`ploop` может работать только с файловой системой `ext4`.

Сами диски хранятся в каталоге `/vz/private` и имеют имя `root.hdd`:

```
# ls -l /vz/private/*/root.hdd/root.hdd
/vz/private/101/root.hdd/root.hdd
/vz/private/102/root.hdd/root.hdd
/vz/private/103/root.hdd/root.hdd
```

Можно быстро изменять размер `ploop`-диска, не отключая при этом контейнер. Просмотрим размеры `ploop`-дисков (1G):

¹production — производственная «боевая» среда

```
# vzlist -o smart_name,diskspace.h
SMARTNAME    DSPACE.H
    101      1113684
    102      1113684
    103      1113684
    104      1113684
```

Изменим размер диска контейнера 103 на 2G:

```
# vzctl set 103 --diskspace 2G --save
dumpe2fs 1.41.12 (17-May-2010)
Changing balloon size old_size=1182793728 new_size
=217055232
Successfully truncated balloon from 1182793728 to 217055232
bytes
UB limits were set successfully
CT configuration saved to /etc/vz/conf/103.conf
```

```
# vzlist -o smart_name,diskspace.h
SMARTNAME    DSPACE.H
    101      1113684
    102      1113684
    103      2056788
    104      1113684
```

За подробной информацией о ploop можно обратиться по адресу:
https://openvz.org/Ploop/Getting_started

8 Управление VPS через Web-браузер

OpenVZ Web Panel¹ представляет собой инструмент для управления серверами OpenVZ через веб-интерфейс. Основные особенности представлены ниже [9]:

- Интуитивно понятный интерфейс;
- Автоинсталлятор панели;
- Поддержка 10 языков интерфейса (в том числе русский и английский);
- Создание/удаление виртуальных серверов;
- Настройка лимитов виртуальных серверов (размер диска, объем памяти, лимиты на CPU);
- Возможность подключения нескольких физических серверов;
- Бэкап/восстановление виртуальных серверов;
- Клонирование виртуальных серверов;
- Быстрая переустановка виртуального сервера;
- Графики использования диска, памяти и процессора;
- Многопользовательская система с ролями.

Установка:

```
# wget -O - http://ovz-web-panel.googlecode.com/svn/
  installer/ai.sh | sh
```

После установки, можно получить доступ к панели по адресу:
`http://IP_address:3000`.

Логин/пароль по умолчанию: `admin/admin`.

Приложение написано на Ruby, с использованием фреймворка Ruby on Rails. Также в проекте используются Ext JS², и SQLite³.

¹OpenVZ Web Panel (OWP) разрабатывается Алексеем Южаковым и распространяется под лицензией GNU GPL v2.0

²Ext JS — библиотека JavaScript для разработки веб-приложений и пользовательских интерфейсов

³SQLite — компактная встраиваемая реляционная база данных

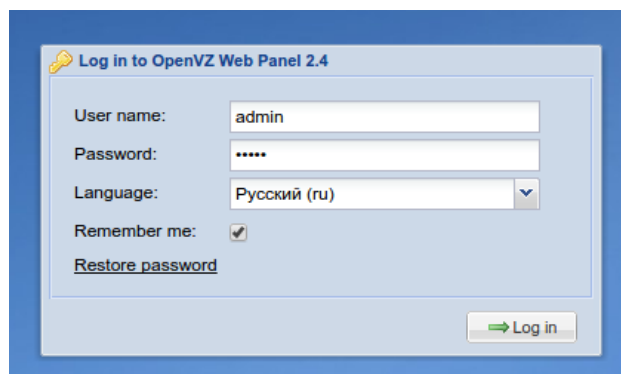
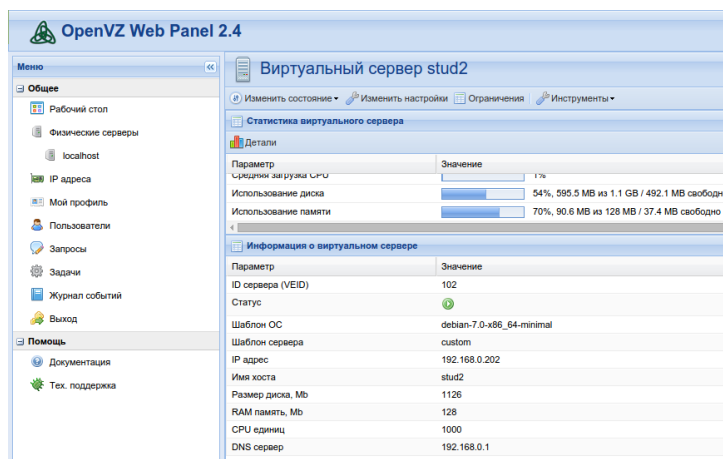


Рис. 9: Вход в панель управления



Параметр	Значение
ID сервера (VEID)	102
Статус	🟢
Шаблон ОС	debian-7.0-x86_64-minimal
Шаблон сервера	custom
IP адрес	192.168.0.202
Имя хоста	stud2
Размер диска, Mb	1126
RAM память, Mb	128
CPU единиц	1000
DNS сервер	192.168.0.1

Рис. 10: Информация о виртуальном сервере в OWP

Всего за небольшой период разработки продукта, OWP обрела обширную аудиторию. В версии 2.0 запланированы Remote API и интеграция с биллингом WHMCS. Исходные файлы проекта в публичном доступе: <https://github.com/sibprogrammer/owp>

9 Рекомендации по работе с системой

- Если работа хост-компьютера замедлилась, можно воспользоваться утилитами `ps`, `dmesg`, `*top`¹;
- По возможности оптимизируйте работу сервера;
- Для обнаружения сетевых проблем можно воспользоваться утилитами `ping`, `traceroute`, `nmap`, `mtr`, `tcpdump`;
- Используйте RAID (Redundant Array of Inexpensive Disks);
- Никогда не перезагружайте компьютер без выяснения обстоятельств неполадок, делайте это только в самых крайних случаях;
- Если хост-компьютер был некорректно отключен, при его следующем запуске все разделы должны быть проверены заново и разделы дисков повторно вычислены для каждого VPS;
- Не запускайте блобы² или скрипты, которые принадлежат VPS, непосредственно с хост-компьютера;
- Вы должны быть способны обнаружить любой руткит³ в контейнере. Для этого рекомендуют использовать пакет `chkrootkit`;
- Следите за нагрузкой сервера, обезопасьтесь от Dos/DDoS;
- Делайте резервные копии (backup) важных данных;
- Следите за свободным местом на жестких дисках, используйте ротацию логов;
- Проверяйте каталог `/var/log/`, который содержит логи системы;
- Используйте `IPTables`, `SSH`, `SSL`, `fail2ban` как на хост-компьютере, так и в контейнерах;
- Подбирайте сложные для перебора пароли, периодически меняйте их, уведомляйте пользователей об этом;

¹`top`, `htop`, `atop`, `iotop`, `iftop`

²Блоб (англ. binary linked object — объект двоичной компоновки) — объектный файл без публично доступных исходных кодов, загружаемый в ядро операционной системы

³Руткит (англ. rootkit) — набор программных средств (например, исполняемых файлов, скриптов, конфигурационных файлов), для обеспечения маскировки объектов, контроля и сбора данных

- Аккуратно работайте на сервере под учетной записью `root`;
- Следите за рассылками новостей по безопасности;
- Обновляйте ПО, систему и ее компоненты;
- Следите за правами пользователей, файлов и каталогов на сервере;
- Используйте системы мониторинга ресурсов (например Cacti, Munin, MRTG, Zabbix, Nagios);
- Ведите внутреннюю документацию по серверам и их настройке;
- В случае обнаружения проблем, можно обратиться к документации проектов OpenVZ и Parallels Cloud Server, а также на задать вопрос на тематических форумах.

Список литературы

- [1] А.Р. Умеров и Е.Н. Машенко. «Анализ технологий контейнерной виртуализации». В: *Мир компьютерных технологий. Материалы внутривузовской студенческой научно-технической конференции*. Севастополь: СевНТУ, 2014, с. 32.
- [2] М. Тим Джонс. «Виртуальный Linux. Обзор методов виртуализации, архитектур и реализаций». В: (2007). URL: <http://www.ibm.com/developerworks/ru/library/l-linuxvirt/>.
- [3] Pradeep Padala и др. «Performance evaluation of virtualization technologies for server consolidation». В: *HP Labs Tec. Report* (2007), с. 4. URL: <http://www.hpl.hp.com/techreports/2007/HPL-2007-59R1.pdf>.
- [4] «OpenVZ Linux Containers Performance. Benchmark results». В: (2011). URL: http://openvz.org/Performance/Response_Time.
- [5] SWSoft Inc. «OpenVZ User's Guide». В: (2005), с. 119. URL: <http://download.openvz.org/doc/OpenVZ-Users-Guide.pdf>.
- [6] Джон Сноу. «Виртуальная реальность по-русски: Осваиваем виртуализацию уровня ОС на примере OpenVZ». В: *Журнал Хакер* (2011). URL: <http://xakep.ru/56244/>.
- [7] Kir Kolyshkin. «On vSwap and 042stab04x kernel improvements». В: (2011). URL: <http://openvz.livejournal.com/39644.html>.
- [8] Kir Kolyshkin. «Introducing container in a file aka ploop». В: (2012). URL: <http://openvz.livejournal.com/40830.html>.
- [9] Алексей Южаков. «OpenVZ Web Panel». В: (2011). URL: <http://habrahabr.ru/post/111907/>.

А Установка CentOS 6

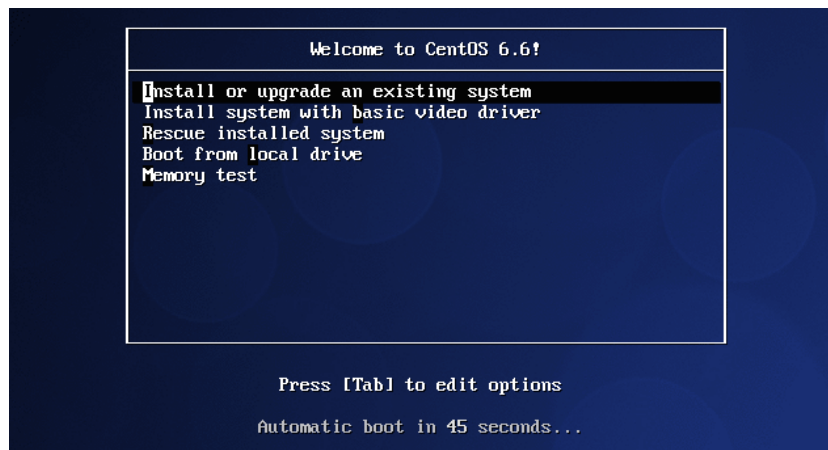


Рис. 11: Меню выбора загрузки с носителя



Рис. 12: Предложение установщика проверить на ошибки носитель

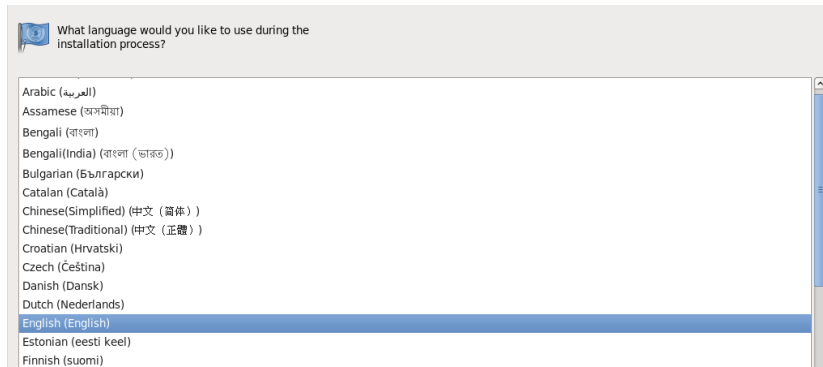


Рис. 13: Выбор языка установки



Рис. 14: Выбор языковой раскладки клавиатуры

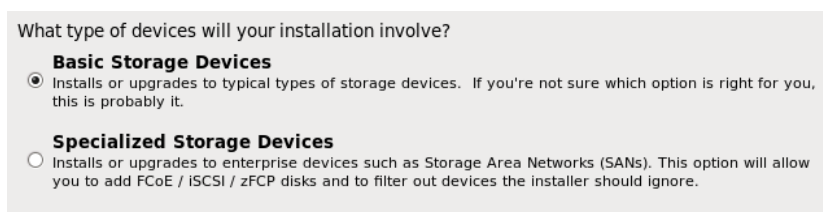


Рис. 15: Проверка наличия специализированных устройств



Рис. 16: Проверка наличия данных на жестком диске

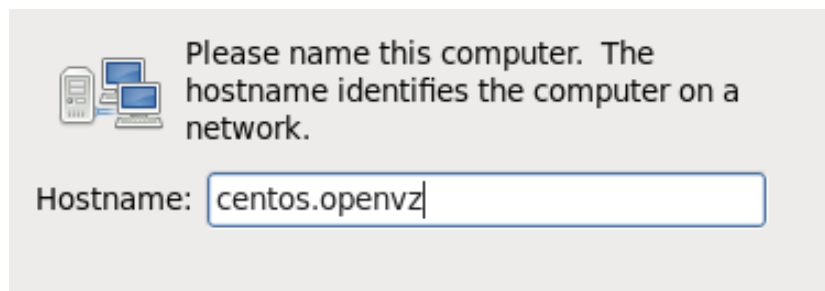


Рис. 17: Задание имени компьютера (hostname)

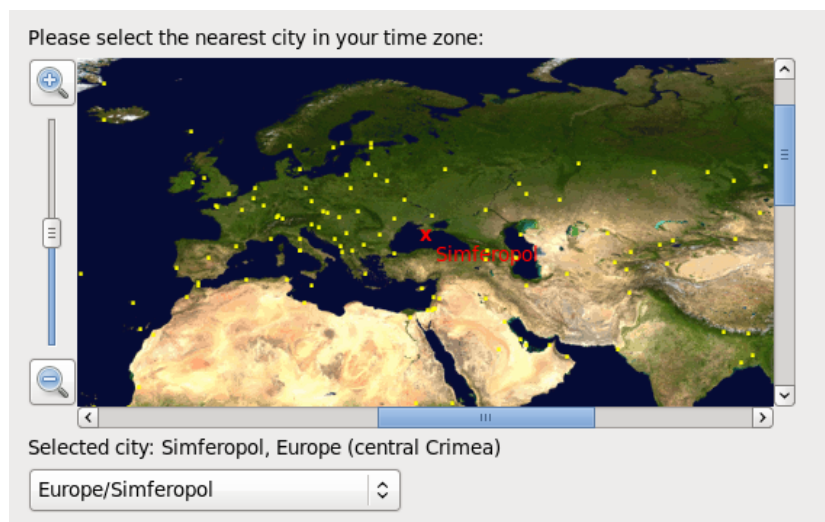



Рис. 18: Выбор часового пояса



The root account is used for administering the system. Enter a password for the root user.

Root Password:

Confirm:

Рис. 19: Задание пароля суперпользователя

Which type of installation would you like?

- ☐ **Use All Space**
Removes all partitions on the selected device(s). This includes partitions created by other operating systems.
Tip: This option will remove data from the selected device(s). Make sure you have backups.
- ☐ **Replace Existing Linux System(s)**
Removes only Linux partitions (created from a previous Linux installation). This does not remove other partitions you may have on your storage device(s) (such as VFAT or FAT32).
Tip: This option will remove data from the selected device(s). Make sure you have backups.
- ☐ **Shrink Current System**
Shrinks existing partitions to create free space for the default layout.
- ☐ **Use Free Space**
Retains your current data and partitions and uses only the unpartitioned space on the selected device(s), assuming you have enough free space available.
- ☒ **Create Custom Layout**
Manually create your own custom layout on the selected device(s) using our partitioning tool.

Рис. 20: Выбор типа разделения жесткого диска

Please Select A Device					
Device	Size (MB)	Mount Point/ RAID/Volume	Type	Format	
▼ Hard Drives					
▼ sda (/dev/sda)					
sda1	6000	/	ext4	✓	
sda2	1024		swap	✓	
sda3	33935	/vz	ext4	✓	

Рис. 21: Разметка жесткого диска

☒ Install boot loader on /dev/sda. [Change device](#)

☐ Use a boot loader password [Change password](#)

Boot loader operating system list

Default	Label	Device
<input checked="" type="radio"/>	CentOS	/dev/sda1

Рис. 22: Установка загрузчика

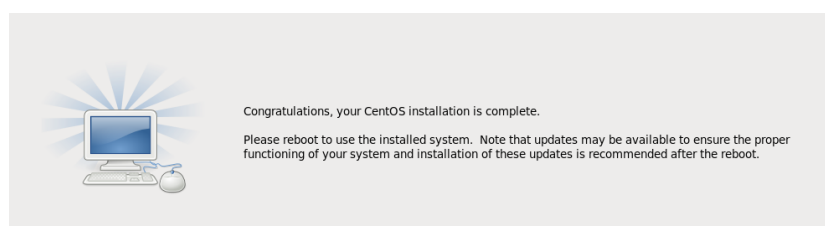


Рис. 23: Установка ОС завершена