

# Содержание

## 1. Предисловие

(данная глава отсутствует :-)

*Информация об этой брошюре - руководстве*

*Кому предназначена эта брошюра*

*Содержание этой брошюры*

*Документация договора*

Типографские условия

Оформительские подсказки-примеры

Основные условия договора

Обратная связь

## 2. Теоретические основы (Философия) Open VZ

*Информация о компьютерной программе OpenVZ*

Что такое OpenVZ?

Приложения OpenVZ

*Отличительные особенности OpenVZ*

Операционная система Virtualization

Сеть Virtualization

Шаблоны

Менеджмент ресурсов

*Главные принципы работы с OpenVZ*

Основы технологии OpenVZ

Про шаблоны

Про лицензии

Конфигурация OpenVZ

*Доступность ядра системы*

## 3. Предварительные операции и операции инсталляции

*Основные требования для инсталляции программы*

Системные требования

Сетевые требования

*Инсталляция и преобразование основной ОС на Hardware Node*

Выбор типа системы

Деление диска

Окончание процесса инсталляции ОС.

*Инсталляция компьютерной программы OpenVZ*

Загрузка и установка ядра Open VZ

Настройка Boot loader ОС

Настройка параметров sysctl

Загрузка и установка архивированных файлов OpenVZ

Установка шаблонов ОС

## 4. Операции с серверами VPS

*Создание и настройка нового виртуального частного сервера*

Прежде, чем начать работу

Выбор идентификационного номера нового VPS

Выбор шаблона ОС

Непосредственное создание VPS

Конфигурирование (преобразование) VPS

*Запуск, остановка, перезагрузка, вопросы относительно состояния VPS*

*Внесение VPS в список VPS*

*Удаление VPS*

*Основные команды VPS*

## 5. Управление шаблонами

**Жизненный цикл шаблонов**  
**Внесение шаблонов в список**  
**Работа с VPS**

**6. Управление ресурсами**

---

**Контрольные параметры ресурсов**

**Управление квотами диска**

Что такое квоты диска?

Параметры квот диска.

Включение и отключение квот Per-VPS диска

Настройка параметров квоты диска Per-VPS

Включение и отключение второстепенных (second-level) квот для Virtual Private Server

Настройка параметров второстепенных квот диска

Проверка статуса квоты

**Управление CPU Share**

**Управление системными параметрами**

Контроль потребления системных ресурсов (System Resources Consumption)

Контроль Memory Consumption (потребления ресурсов памяти)

**Управление конфигурацией ресурсов VPS**

Разделение Hardware Node на равные части

Утверждение конфигурации Virtual Private Server

**7. Задачи первостепенного выполнения**

---

**Определение ID VPS**

**Изменение системного времени VPS**

**Устройства доступа на VPS**

**Перемещение сетевого адаптера на VPS**

**Обеспечение VPN для VPS**

**Загрузка iptables модулей**

Загрузка iptables модулей на hardware node

Загрузка iptables модулей на определенный VPS

**Создание конфигурационных файлов для нового разделения Linux**

**Перезагрузка VPS**

**8. Поиск ядерных неисправностей**

---

**Общие рассмотрения**

**Выявление ядерных неисправностей**

Сохранение ядерных ошибок (OOPS)

Обнаружение ядерной функции, вызвавшей процесс D

**Проблемы с управлением VPS**

Отказ создать VPS

Отказ запускать VPS

Отказ в доступе к VPS

Отказ войти на VPS

**Проблемы с функционированием VPS**

Перерыв при доступе к удаленным hosts

**Проблемы с использованием Linux утилит (?)**

**Получение технической поддержки (?)**

**9. Ссылки**

---

**Конфигурирование Open VZ**

Матрица файлов конфигурации Open VZ

Глобальный файл конфигурации

Файл конфигурации VPS

Управление скриптами Open VZ

**Интерфейс командных строк Open VZ (OpenVZ Command Line Interface)**

Матрица утилит командных строк OpenVZ.

vzctl

vzlist  
vzquota  
Утилиты управления шаблоном  
Дополнительные инструменты

## ***Глоссарий***

---

## ***Глава 2***

### ***Теоретические основы (Философия) OpenVZ***

*В этой главе:*

- 1. Информация о компьютерной программе OpenVZ**
  - 2. Отличительные особенности OpenVZ**
  - 3. Главные принципы работы с OpenVZ**
  - 4. Информация относительно доступности Hardware Node (ядра системы)**
- 

## **1. Информация о компьютерной программе OpenVZ**

### **Что такое OpenVZ?**

OpenVZ – это полностью автоматизированная и виртуализованная система для сервера, разработанная SWsoft. OpenVZ создает многочисленные изолированные Virtual Private Servers (VPSs) на одном физическом сервере, чтобы распределить hardware и усилия по управлению системой с максимальной долей эффективности. Каждый VPS функционирует для его пользователей и приложений как автономный сервер, так как он может быть перезагружен отдельно и, кроме того, имеет отдельный доступ к коренной папке, имеет своих пользователей, IP адреса, память, процессы, файлы, приложения, системные библиотеки, и файлы конфигурации. Легкий и эффективный дизайн OpenVZ делает его правильным выбором виртуализованных production серверов с live applications и real-life data.

Основные возможности OpenVZ VPS:

- Динамичное деление в режиме реального времени: деление физического сервера на десятки VPSs, каждый с полной функционализацией сервера.
- Управление ресурсами: назначает и управляет параметрами ресурса VPS и перераспределяет ресурсы в режиме реального времени.
- Общее управление: управляет множеством физических серверов и VPS комплексно.

### **Приложения OpenVZ**

OpenVZ является уникальным решением для Hosting Service Провайдеров, позволяя им:

- иметь сотни клиентов с их индивидуальными VPS серверами и, тем не менее, разделяющими единый физический сервер;

- обеспечить каждого пользователя гарантированным качеством обслуживания;
- виртуозно, прозрачно перемещать пользователей и их окружение между серверами без ручной реконфигурации.

Если вы управляете множеством выделенных серверов Linux в пределах одного предприятия, каждый из которых выполняет отдельные функции по обслуживанию, вы можете использовать OpenVZ, чтобы объединить все эти серверы и перенести на один компьютер, не теряя ценной информации и не ставя под угрозу работу всей системы.

VPSs действуют как изолированный автономный сервер:

- каждый VPS имеет его собственные процессы, пользователей, файлы и обеспечивает полный доступ к корневым файлам;
- каждый VPS имеет его собственные IP адреса, номера порта, проникающие и направляющие правила;
- каждый VPS может иметь его собственную конфигурацию для системы и прикладного программного обеспечения, также как и его собственные версии системных библиотек.

Также, возможно устанавливать или настраивать пакеты программного обеспечения в VPS независимо от других VPSs или хост системы (Host system). Одна и та же Linux box может управлять многочисленными распределениями пакета.

Фактически, сотни серверов могут группироваться таким образом. Помимо очевидных преимуществ такой консолидации (увеличенная легкость управления и т.п.), есть некоторые, о которых Вы возможно, даже и не думали, скажем, значительное уменьшение платы за электроэнергию!

OpenVZ оказывается неоценимым для IT образовательных учреждений, которые могут теперь обеспечить каждого студента личным сервером Linux, который, в свою очередь, может проверяться и управляться удаленно. Компании, развивающие программное обеспечение, могут использовать virtual environments с целью тестирования и т.п.

Таким образом, OpenVZ может быть эффективно применен в широком диапазоне областей: web хостинг, объединение серверных предприятий, разработка программного обеспечения и его испытание, пользовательское обучение, и так далее.

## ***2. Отличительные Особенности OpenVZ***

Концепция OpenVZ Virtual Private Servers отличается от основной идеи традиционных «virtualization» машин тем, что VPSs серверы всегда управляют тем же самым ОС ядром (OS kernel) как и хост система (the host system) – то есть - (Linux on Linux, Windows on Windows, etc.). Такая едиоядерная технология работы позволяет управлять серверами VPSs с фактически нулевыми расходами. Таким образом, OpenVZ VPSs серверы предлагают на порядок более

высокую эффективность и управляемость в сравнении с традиционными «virtualization» технологиями.

## **ОС Virtualization**

С точки зрения приложений и пользователей VPS, каждый VPS – это независимая система. Эта независимость достигается за счёт специального слоя «virtualization» в ядре хост ОС (the host system). Необходимо заметить, что лишь мельчайшая часть CPU ресурса затрачена на виртуализацию (1-2 %). Главными особенностями слоя «virtualization», заложенного в OpenVZ, являются:

- VPS выглядит как обыкновенная система Linux. VPS имеет стандартные скрипты запуска, программное обеспечение от продавцов может работать внутри VPS без специализированных модификаций OpenVZ или особенного регулирования;
- Пользователь может изменить любой файл конфигурации и установить дополнительное программное обеспечение;
- VPSs полностью изолированы друг от друга (система файла, процессы, IPS, sysctl переменные);
- Процессы, принадлежащие VPS намечаются для выполнения на всех доступных CPU.

Следовательно, VPSs не привязаны только к одному CPU и могут использовать всю доступную мощность CPU.

## **Сеть Virtualization**

Слой виртуализации сети OpenVZ спроектирован таким образом, чтобы изолировать серверы VPS друг от друга и от физической сети:

- Каждый VPS имеет его собственный IP адрес; разрешается/возможно также иметь несколько IP адресов на одном сервере VPS;

Трафик сети VPS изолирован от других серверов VPSs. Другими словами, серверы VPS защищены друг от друга в пути, так что, шпионаж по трафику исключен;

- Внутри сервера VPS может использоваться система Firewalling (т.е. пользователь может создать такие условия - правила, ограничивающие доступ к некоторым сервисам, используя канонический iptables инструмент внутри VPS). То есть, возможно создать эти способы защиты внутри VPS;
- Список направляющих действий позволяют извлечь выгоду из современных-передовых качеств направления (routing). Например, устанавливая различные MTUs для всяческих

целей, определяя различные адреса источника обращается для различных целей, и так далее.

## **Шаблоны (Templates)**

Образец ОС в OpenVZ является набором пакетов от распространения Linux, используемого для наполнения одного или более серверов VPS. С системой OpenVZ, различные распределения могут сосуществовать на одном и том же модуле обеспечения (hardware box), таким образом, многочисленные шаблоны ОС являются доступными. Образец ОС состоит из системы программ, библиотек, и скриптов необходимых для загрузки и управления системой (VPS) и нескольких приложений и утилит. Приложения, такие как компилятор и сервер SQL, обычно не включаются в образец ОС.

За более подробной информацией относительно шаблонов OpenVZ обращайтесь в раздел «Про шаблоны» (Understanding Templates).

## **Менеджмент ресурсов**

Менеджмент ресурсов OpenVZ контролирует количество ресурсов, доступных для серверов VPS. Управляемые ресурсы включают такие параметры как мощность CPU, место на диске, ряд параметров, связанных памятью. Менеджмент ресурса позволяет OpenVZ:

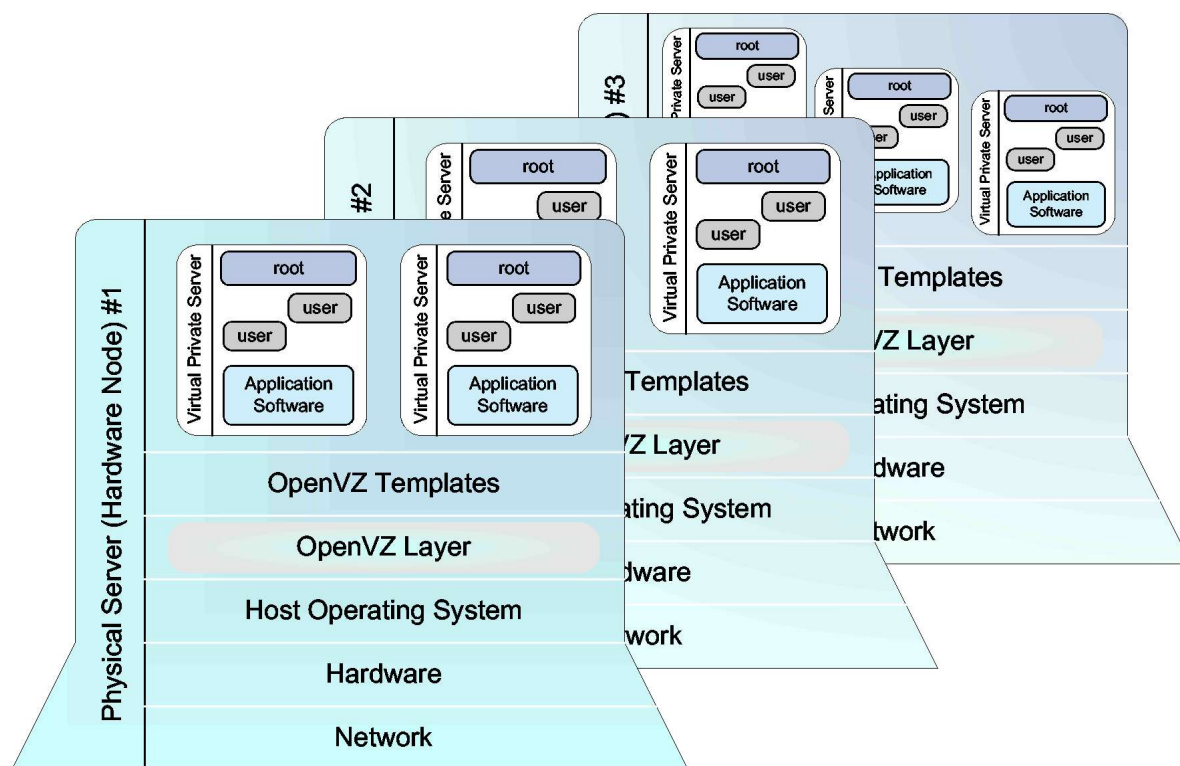
- эффективно разделять доступные ресурсы Hardware Node между серверами VPS ;
- гарантировать качество сервиса (QoS) в соответствии с уровнем сервисного соглашения (SLA);
- обеспечить работу и изоляцию ресурса и защитить от внезапных отказов в обслуживании;
- одновременно назначать и управлять ресурсами множества серверов VPS, и т.д.

Менеджмент ресурсов куда более важнее для Open VZ чем для автономного компьютера, так как утилизация компьютерного ресурса в системе, базирующейся на OpenVZ значительно выше чем в обычной системе.

## **3. Главные принципы работы с OpenVZ**

### **Основы технологии OpenVZ**

В этом разделе мы постараемся сформировать у Вас более или менее ясное представление о том как система OpenVZ будет работать на Вашем компьютере. Пожалуйста, обратите внимание на следующий рисунок:



**Рисунок 1 Технология Open VZ**

Исходя из этого рисунка, предполагается, что у Вас имеется некоторое количество физических серверов, объединенных в сеть. Фактически, для эффективного использования OpenVZ для потребностей Вашей сети, Вам может быть достаточно лишь одного выделенного сервера. Если у Вас более одно физического сервера, базирующегося на системе OpenVZ, то каждый из серверов будет иметь подобную структуру. В соответствии с терминологией OpenVZ, такие серверы именуются Hardware Nodes (HN или Nodes), так как они представляют единицы аппаратных (hardware units) средств в пределах сети.

Система OpenVZ установлена на Fedora Core 3 или Red Hat Enterprise Linux 4, преобразованных определенным образом. Например, такая настроенная конфигурация должна включать создание /vz partition (разделение), которое является основным разделением для хостинг серверов VPS и которое должно быть больше, чем корневое деление. Эти и подобные проблемы конфигурации могут быть легко решены в процессе установки Linux на Hardware Node. Более детальные инструкции по установке Linux (названной *Host Operating System* или *Root Operating System* на *рисунке 1*) на Hardware Node будут подробно рассмотрены в следующей главе.

OpenVZ устанавливается таким образом, что Вы будете в состоянии загрузить ваш компьютер



либо с помощью OpenVZ, либо без него. В первом случае поддержка OpenVZ будет представлена как “OpenVZ” на Вашем загрузчике (boot loader) и показана как Слой OpenVZ (*OpenVZ Layer*) на рисунке 1.

Однако, на этом этапе Вы еще не способны создавать какие-либо серверы VPS. Сервер VPS по своим функциям идентичен изолированному автономному серверу, имея свои собственные IP адреса, процессы, файлы, пользователей, собственные файлы конфигурации, приложения, системные библиотеки и так далее. Серверы VPS делят один и тот же *Hardware Node* и ядро ОС. Однако, они изолированы друг от друга. VPS - своего рода ‘песочница’ для процессов и пользователей.

Различные серверы VPS могут управлять различными версиями Linux (например, SuSE 9.2 или Fedora Core 4 и многие другие). Каждый VPS может управлять своей собственной версией Linux. В этом случае мы говорим, что VPS основан на определенном шаблоне ОС. Шаблоны ОС – это пакеты, доставленные OpenVZ. Прежде, чем создать VPS, Вы должны установить соответствующий шаблон ОС в OpenVZ. Это показано как Шаблоны OpenVZ (*OpenVZ Templates*) на рисунке 1.

После того, как Вы установили, по крайней мере, один шаблон ОС, Вы можете создать любое количество серверов VPS с помощью стандартных утилит OpenVZ, сформировать их сеть, отрегулировать другие параметры настройки, и начать работать с этими серверами VPS как с полнофункциональными серверами Linux.

## **Про шаблоны**

Шаблон - это блок построения VPS. Шаблон ОС - это ряд пакетов необходимых для управления VPS. Обычно шаблоны создаются непосредственно прямо на вашем Hardware Node. Все, что Вам необходимо – это инструменты шаблона (vzpkg) и блок информации о шаблонах (template metadata)

### ***Блок информации о шаблонах (template metadata)***

Template metadata - информация о конкретном шаблоне ОС. Эта метадата содержит:

- список пакетов, включенных в этот шаблон (в форме названий пакета);
- местоположение (сети) репозиторий пакетов;
- дистрибутивно-специфические скрипты необходимые для их же осуществления на различных стадиях инсталляции шаблона;
- общий ключ (и) GPG необходимый для проверки подписей пакетов;

Вся эта информация содержится в нескольких файлах, установленных в директории /vz/template/osname/osrelease/config/. Например, метадата для шаблона Fedora Core 4 установлена в директории /vz/template/fedoracore/4/config.

Наряду с template metadata, предусмотрено несколько специфических OpenVZ пакетов; они установлены в директории /vz/template/osname/osversion/vz-addons/.

### ***Шаблон кэш (Template cache)***

Шаблон metadata обеспечивает достаточно информации, чтобы создать шаблон ОС. Во время создания шаблона ОС, необходимые файлы пакета загружаются из репозитория сети на Hardware Node и устанавливаются на временном сервере VPS, который затем упаковывается в gzipped tarball, именуемый КЭШем (тайником) шаблона - template cache.

Template cache используется для быстрого обеспечения VPS – так как в основном, это сервер VPS на стадии создания. Таким образом, чтобы создать сервер VPS, необходимо распечатать этот файл (template cache). Файлы template cache сохранены в директории /vz/template/cache/.

Любой template cache устаревает со временем, поскольку постоянно выпускаются обновления. Естественно, существует путь быстрого апдейта кэш шаблона, так же как обновляются устаревающие серверы VPS. есть путь быстро обновить тайник шаблона так же как весь предварительно созданный VPSs с самыми новыми обновлениями.

В то время как Вы в состоянии выполнить все виды задач в пределах сервера VPS, включая создание пакетов *rpm* и их установку, OpenVZ обеспечивает более легкий и эффективный путь установки приложений необходимых на серверах VPS. Точно также как Вы устанавливаете шаблон ОС в системе OpenVZ для создания необходимого количества серверов VPS соответственно, Вы можете установить приложения в OpenVZ с целью разделения пакета файлов среди любого количества серверов VPS. Затем Вы можете тогда добавить эти приложения к любому количеству серверов VPS.

Само собой разумеется, что в случае, если Вы хотите установить приложение только на одном сервере VPS, нет никакой необходимости работать с шаблонами: Вы вполне можете работать внутри соответствующих серверов VPS.

### **Про лицензии**

Программное обеспечение OpenVZ состоит из ядра OpenVZ и инструментов пользовательского уровня, которые лицензируются посредством двух различных открытых лицензий.

- Ядро OpenVZ основано на ядре Linux, распространяемое на условиях GPL и лицензируемое GNU GPL версия 2. Текст лицензии находится по адресу <http://openvz.org/documentation/licenses/gnu-gpl>.
- Инструменты пользовательского уровня (vzctl, vzquota, и vzpkg) лицензируются в соответствии с условиями лицензии QPL. Текст лицензии Вы сможете найти по адресу <http://openvz.org/documentation/licenses/qpl>.

### ***Конфигурация OpenVZ***

OpenVZ позволяет Вам легко формировать различные параметры настройки для системы OpenVZ в целом, также как и для каждого VPS в отдельности. Среди этих параметров настройки - дисковая и пользовательская доли, параметры сети, местоположения файла по умолчанию, файлы образца конфигурации, и другие.

OpenVZ хранит информацию конфигурации в двух типах файлов: глобальный файл конфигурации */etc/sysconfig/vz* и файлы конфигурации VPS */etc/sysconfig/vzscripts/vpsid.conf*. Глобальный файл конфигурации определяет общие параметры и параметры по умолчанию для операции VPS, например, регистрируя параметры настройки, задействуя и отключая дисковую квоту для серверов VPS, файл конфигурации по умолчанию и шаблон ОС, на основе которых создается новый сервер VPS, и так далее. С другой стороны, файл конфигурации сервера VPS определяет параметры для конкретного сервера VPS, типа дисковой квоты и пределов ресурсов, IP адресов и наименования хост компьютера (HOST NAME). В случае если параметр формируется и в глобальном файле конфигурации OpenVZ, и в файле конфигурации сервера VPS, файл конфигурации сервера VPS получает приоритетную позицию. Для того, чтобы ознакомиться со списком параметров, составляющих глобальный файл конфигурации и файлы конфигурации сервера VPS, обратитесь к главе Ссылки (*Reference*).

Файлы конфигурации могут быть изучены, лишь, когда работает система OpenVZ и/или серверы VPS. Однако, стандартные утилиты системы OpenVZ, например, *vzctl*, позволяют Вам тут же изменить параметры настроек конфигурации OpenVZ, либо, не изменяя соответствующие файлы конфигурации, либо модифицируя их (если Вы хотите, чтобы изменения произошли в следующий раз, как только Вы начнете работу с системой OpenVZ и/или серверами VPS).

#### ***4. Информация относительно доступа (доступности/пригодности Hardware Node) к Hardware Node***

Необходимо заметить, что доступ к ядру системы (т.е. Hardware Node) является более опасным чем доступ к ядру сервера стандартного ПК. Так как Hardware Node управляет многочисленными серверами VPS, выполняя множество важных функций, приостановка работы ядра (т.е. главного компьютера, где установлена система Open VZ) может серьезно, негативно сказаться на работе всей системы, сети. То есть выход из строя ядра сравним по последствиям с одновременной приостановкой работы нескольких серверов, выполняющих важные функции в системе.

Чтобы увеличить степень пригодности Hardware Node, предлагается сделать следующее:

- Используйте хранилище RAID для важных частных областей сервера VPS. Рекомендуется отдать предпочтение hardware RAID, в самом крайнем случае software mirroring RAID.
- Не используйте программным обеспечением непосредственно на Hardware Node. Создайте специальные серверы VPS, где вы можете разместить необходимые серверы, типа BIND,FTPD,HTTPD и другие. На самом Hardware Node Вам потребуется лишь SSH daemon. Желательно, чтобы он соединялся лишь с predetermined количеством IP адресов.
- Не создавайте пользователей на самом Hardware Node. Для этого используйте серверы VPS, где Вы можете создать столько пользователей сколько душе угодно. Внимание: ставя под угрозу Hardware Node, Вы делаете тоже самое с серверами VPS.

## ***Глава 3***

### ***Предварительные операции и операции инсталляции***

*В этой главе:*

- 5. Основные требования для инсталляции программы***
- 6. Инсталляция и преобразование главной (хост) Операционной Системы на Hardware Node***
- 7. Инсталляция компьютерной программы OpenVZ***

---

#### ***1. Основные требования для инсталляции программы***

После того как Вы решили, какую структуру выбрать для Вашей системы OpenVZ, Вам необходимо убедиться, что все *Hardware Nodes*, на которых Вы собираетесь запустить систему OpenVZ для Linux, соответствуют следующим общесистемным (hardware и software), а также системным требованиям.

### **Системные требования**

Здесь мы делаем акцент первую очередь на общесистемных требованиях для программного обеспечения системы OpenVZ для Linux.

### **Совместимость оборудования (hardware)**

Основными Hardware Node требованиями для стандартного 32-битного издания OpenVZ являются:

- ПК совместимый с IBM;
- Intel Celeron, Pentium II, Pentium III, Pentium 4, Xeon, или AMD Athlon CPU;
- По крайней мере, 128 MB аудиометафайла (RAM);
- По меньшей мере, 4 ГБ свободного места на жестком диске (дисках);
- Карта сети (рекомендуемые варианты - Intel EtherExpress100 (i82557-, i82558- or i82559-based) или 3Com (3c905 или 3c905B или 3c595), или RTL8139- based).

Компьютер должен соответствовать требованиям Red Hat Enterprise Linux или Fedora Core hardware (списки совместимости оборудования (hardware) ищите по адресу [www.redhat.com](http://www.redhat.com)).

Точная компьютерная конфигурация зависит, от количества серверов VPS, которые Вы собираетесь установить, и от предполагаемой нагрузки, которую будут нести Ваши серверы VPS. Чтобы выбрать правильную конфигурацию следуйте рекомендациям ниже:

- Центральные процессоры (CPU). Чем больше серверов VPS Вы планируете использовать одновременно, тем больше CPU Вам потребуется.
- Память. Чем больше у Вас памяти, тем больше серверов VPS Вы можете иметь. Точная цифра зависит от количества и качества приложений Вы планируете иметь на серверах VPS. В среднем, на каждые 20-30 серверов VPS требуется 1 ГБ RAM.
- Место на диске. Каждый сервер VPS занимает в среднем около 400-600 МБ жесткого диска), о включает системные файлы), плюс, данные пользователя (user data) внутри сервера VPS (такие как web site content). Все это Вы должны учесть, планируя деление диска и количество серверов VPS.

Типовой 2-way Dell PowerEdge 1650 lu-mountable с 1 ГБ RAM и 36 ГБ жестких дисков подходит для обеспечения 30 серверов VPS.

### **Совместимость программного обеспечения**

Hardware Node должен использовать либо Red Hat Enterprise Linux 3 или 4, либо Fedora Core 3 или 4, либо же CentOS 3.4 или 4. Для более качественной работы OpenVZ смотрите подробные инструкции по установке этих операционных систем в следующих разделах.

Следует заметить, что это требование не ограничивает способность OpenVZ предоставить другие версии Linux в качестве операционной системы для сервера VPS. Распределение Linux, установленное на сервере VPS может отличаться от дистрибуции в хост ОС (host OS).

### **Требования Сети**

Предпосылки сети, описанные здесь помогут Вам избежать неполадок с запуском и управлением системой OpenVZ для Linux. Прежде всего, Вы должны позаботиться о следующем:

- Локальная сеть (LAN) для Hardware Node;
- Интернет-связь для Hardware Node;
- Валидный \ действующий IP адрес и другие параметры (default gateway, network mask, DNS конфигурация (configuration) для Hardware Node;
- По крайней мере, один действующий IP адрес для каждого сервера VPS. Общее количество адресов должны быть не меньше чем запланированное число серверов VPS. Адреса могут быть

размещены в различных IP сетях;

- Если используется брандмауэр, проверьте, чтобы IP адреса серверов VPS были доступны снаружи.

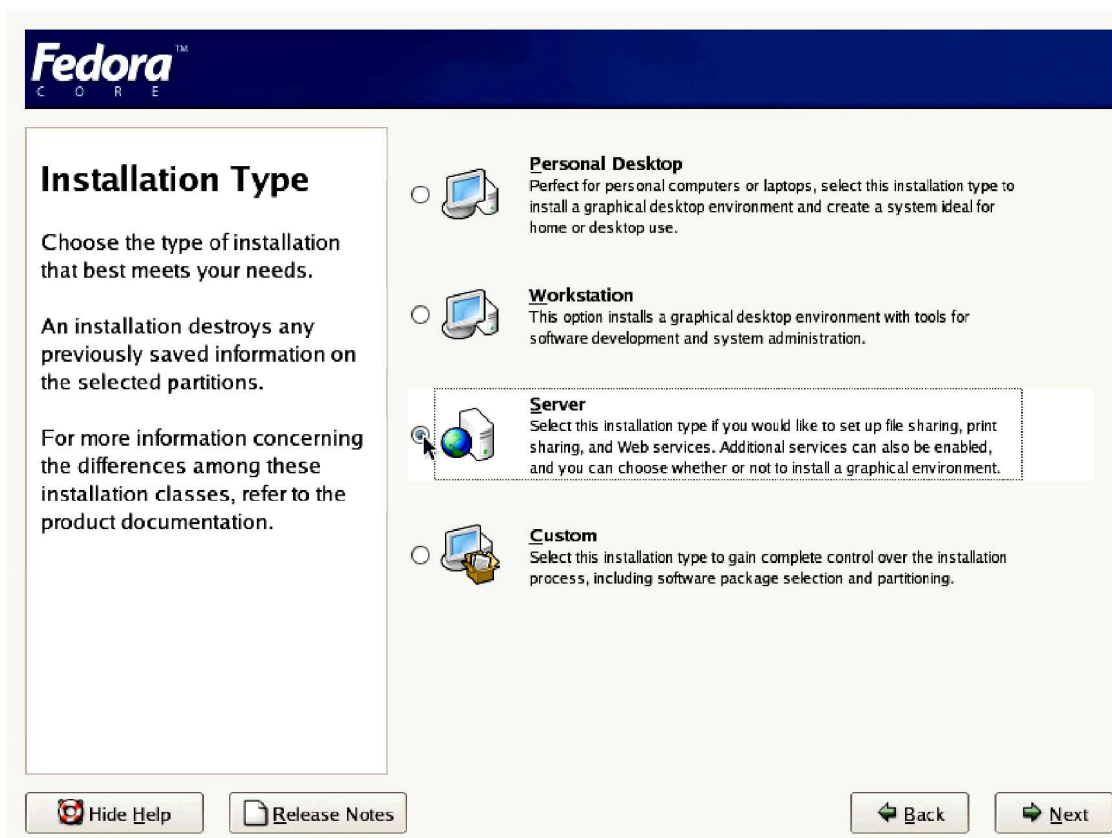
## ***2. Инсталляция и преобразование основной (Хост) Операционной Системы на Hardware Node***

В этом разделе Вы узнаете, как установить Fedora Core 4 на Hardware Node и как изменить ее применительно к OpenVZ. Если Вы используете другое распределение, пожалуйста, обратитесь по вопросу специфики установки к соответствующим руководствам по установке.

### **Выбираем тип системы**

Пожалуйста, устанавливая ОС на Hardware Node, следуйте инструкциям вашего руководства по установке. После нескольких первых картинок, на экране появиться картинка с выбором типа

инсталляции. Для OpenVZ требуется установка Системы сервера (Server System) , поэтому выберите "Сервер" как это показано на рисунке ниже.



*Рисунок 2 Установка Fedora Core - выбираем тип системы*

На рисунке: **Installation type** - тип инсталляции. Выберите необходимый тип инсталляции, который наиболее верно отвечает Вашим требованиям. Инсталляция разрушает любую ранее сохраненную информацию на выбранных частях (partitions). За более подробной информацией о разнице между типами инсталляции обращайтесь к документации данного продукта.

**Personal desktop** – индивидуальный рабочий стол - идеально подходит для ПК и лэптопов. Выберите этот тип инсталляции для того, чтобы установить графические параметры рабочего стола и создать систему идеально подходящую для домашнего использования.

**Workstation** – рабочая станция. Выбрав эту опцию, Вы загрузите графические параметры рабочего стола вместе с инструментами для развития программного обеспечения и системно администрирования.

**Server** – сервер – если хотите установить совместное пользование файлов, совместную печать, совместные web сервисы. Могут быть также установлены дополнительные сервисы. Кроме того, можно выбрать устанавливать или нет графические параметры.

**Custom** –обычный – чтобы получить полный контроль над процессом инсталляции, включая выбор пакет программного обеспечения или деления диска.

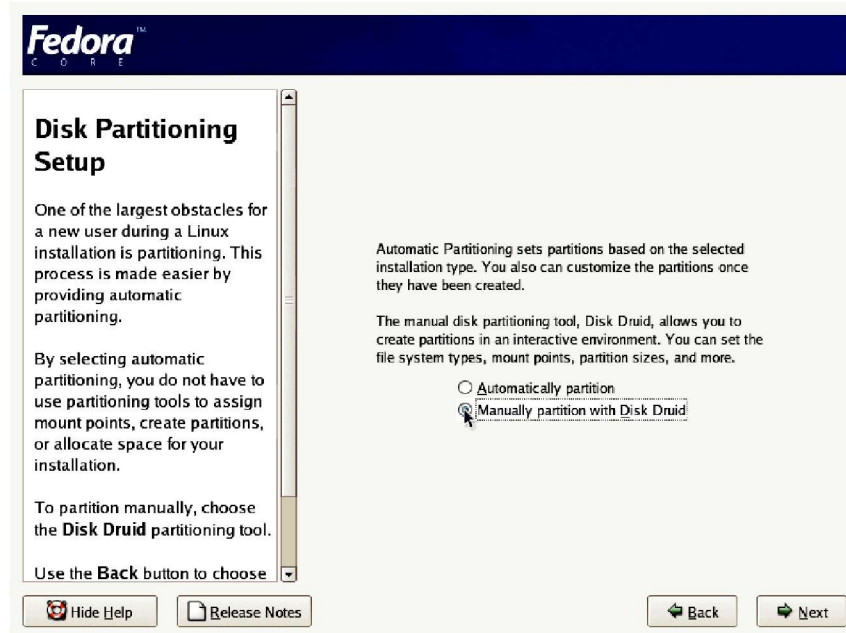
Не рекомендуют устанавливать дополнительные пакеты непосредственно на Hardware Node из-за особенной важности Hardware Node (см. **Hardware Node Availability Considerations** в этой главе).

Вы будете в состоянии управлять любыми необходимыми сервисами, внутри отдельных серверов

VPS.

### Деление диска

Выберите **Manual partition with Disk Druid** на экране с окошком **Disk Partitioning Setup**. Не выбирайте автоматическое деление, так как этот тип деления создаст раскладку диска (disk layout), предназначенных для систем управляющих многочисленными сервисами. В случае с системой Open VZ, все сервисы будут действовать внутри сервера VPS.



*Рисунок 3 Fedora Core Installation - выбор деления вручную*

Создайте следующее разделение на Hardware Node:

Create the following partitions on the Hardware Node:

Partition	Description	Typical size
/	Root partition containing all Hardware Node operating system files	2-4 Gb
swap	Paging partition for the Linux operating system	2 times RAM
/vz	Partition to host OpenVZ templates and Virtual Private Servers	all the remaining space on the hard disk

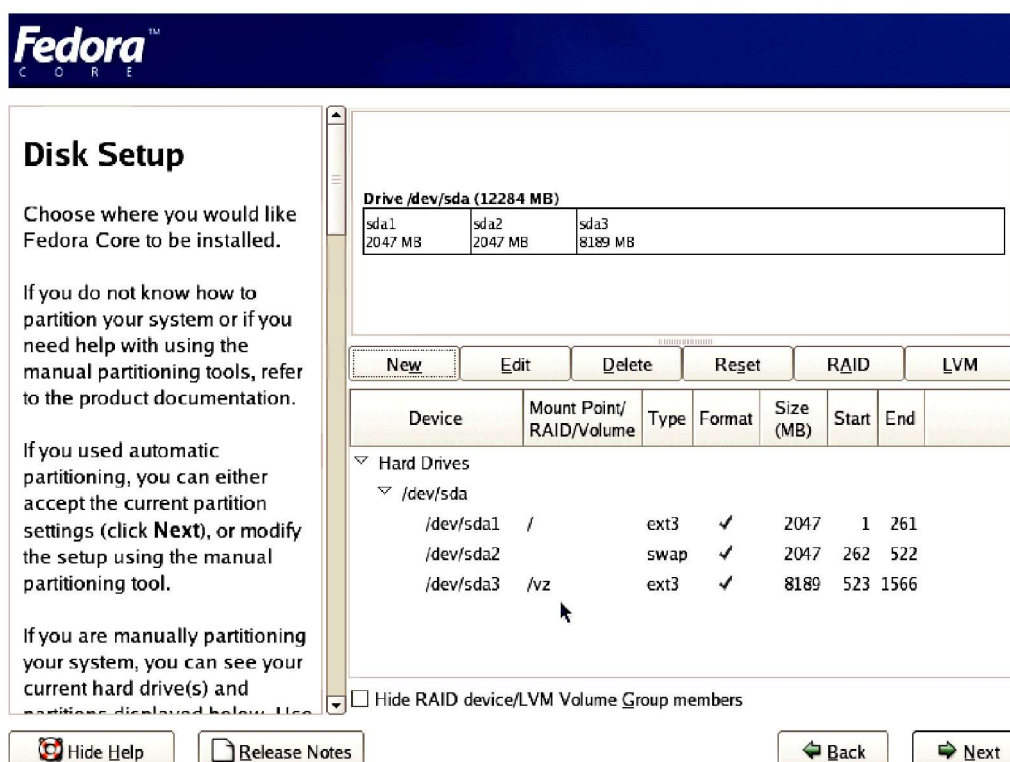
<i>Деление</i>	<i>Описание</i>	<i>Обычный размер</i>
/	корневое деление, содержащие все файлы операционной системы Hardware Node	2-4 ГБ
swap	страничное деление для операционной системы Linux	2 RAM
/vz	деления для хост шаблонов Open VZ и серверов VPS	все оставшееся место на жестком диске



Предлагается использовать систему файлов *ext3* для деления */vz*. Это разделение используется для содержания всех данных серверов VPS, существующих на Hardware Node. Выделите как можно больше места на диске для этого разделения. Не рекомендуют использовать систему файлов *reiserfs* поскольку доказано, что она менее устойчива чем *ext3*, а стабильность имеет первостепенную важность для компьютеров, базирующихся на системе OpenVZ.

Корневое деление будет оперировать (to host?) файлами операционной системы. Набор сервера Fedora Core 4 занимает приблизительно 1 ГБ дискового пространства, таким образом, 1 ГБ - минимальный размер корневого деления. Размер обменного деления должен равняться двум размерам физического RAM, установленного на Hardware Node.

Рисунок ниже представляет систему с 12 ГБ SCSI жесткого диска.



*Рисунок 4 Инсталляция Fedora Core – друид диска*

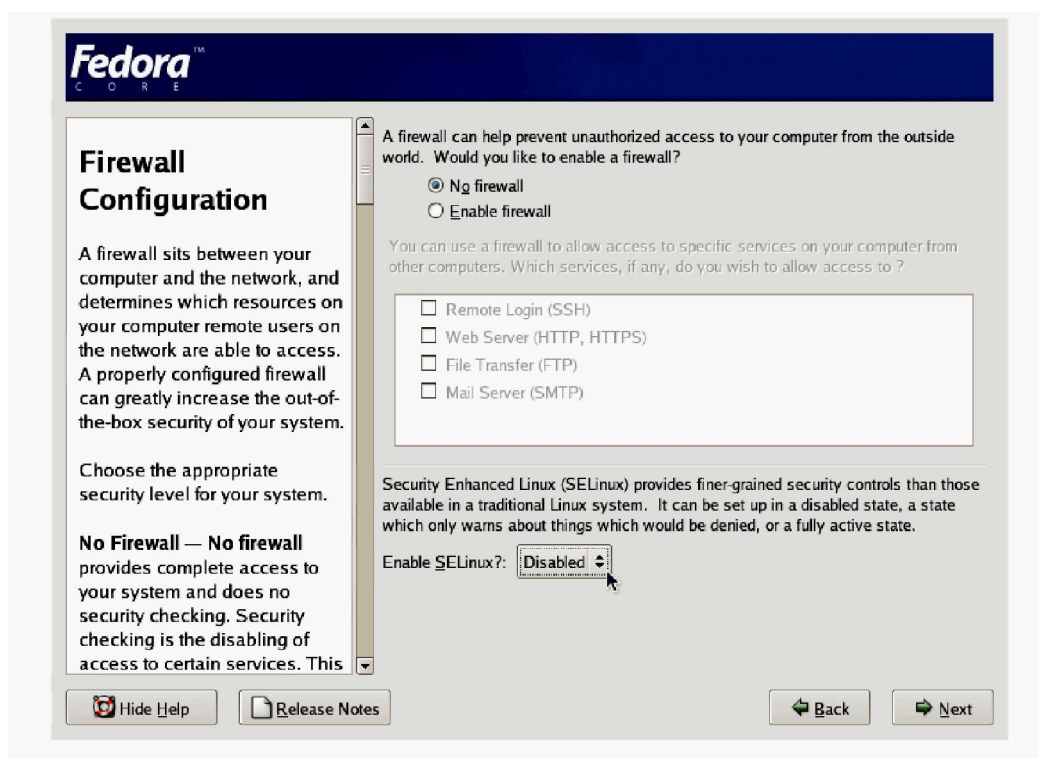
Имейте в виду, что частные сферы серверов VPS, содержащие всю базу данных этих серверов должны находиться на одном единственном */vz* дисковом делении вместе со всеми установленными шаблонами.

### Окончание Установки ОС

После надлежащего деления вашего жесткого диска (дисков), следуйте далее в соответствии с инструкциями Вашей инструкции по инсталляции (Installation Guide).

В то время как на экране появится Network Configuration (конфигурация сети), Вы должны убедиться в правильности IP адреса Hardware Node, хост имени, DNS, и gateway информации по умолчанию. Если Вы используете DHCP, удостоверьтесь, что он работает должным образом. В случае необходимости, проконсультируйтесь с администратором вашей сети.

На картинке Firewall Configuration, выберите No Firewall (никакого брандмауэра). Опция **Enable SELinux** должна быть выбрана недействующей (disabled).



*Рисунок 5 Инсталляция Fedora Core- выключение Firewall и SELinux*

После окончания процесса установки и перезагрузки вашего компьютера, Вы наконец – то готовы установить систему Open VZ на Ваш компьютер.

### **3. Инсталляция компьютерной программы OpenVZ**

Прежде всего, Вы должны загрузить двоичное ядро RPM (kernel binary RPM) по следующей ссылке <http://openvz.org/download/kernel/>. Вам необходимо только одно ядро RPM, поэтому выберите подходящее бинарное ядро в соответствии с Вашим оборудованием (hardware)

- Если на Вашем Hardware Node существует больше одного CPU (или CPU с гиперпронизывание (hyperthreading), выберите *vzkernel - smp* RPM.
- Если доступно более 4 Гб (Gb) RAM, выберите *vzkernel - version* RPM.

- В любом другом случае рекомендуется выбирать uniprocessor kernel RPM (*vzkernel1 – version*)

Затем, Вы должны установить на Hardware Node ядро RPM по Вашему выбору, выполнив следующую команду:

```
# rpm -ihv vzkernel-name*.rpm
```

Замечание: Не используйте команду *rpm -U* (U в этом случае означает upgrade) иначе все недавно установленные на Node ядра будут перемещены.

## Настройка Boot Loader

В том случае если Вы пользуетесь загрузчиком GRUB, Boot loader будет настроен и сформирован автоматически. Только убедитесь, чтобы строчки ниже присутствовали в файле */boot/grub/grub.conf* на Node:

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

Тем не менее, мы рекомендуем сформировать, настроить этот файл следующим образом:

- Замените *Fedora Core* на *Open VZ* (просто для ясности, чтобы ядра Open VZ не перемещались с другими ядрами)
- Удалите все дополнительные аргументы (действительные параметры) со строки ядра, оставив только *root=...* параметр.

В конечном итоге, измененный файл *grub.conf* должен выглядеть следующим образом:

```
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 параметров sysctl

Существует ряд ограничений ядра, которые должны быть правильно настроены, чтобы система Open VZ работала качественно. Open VZ доставляется с настроенным файлом */etc/sysctl.conf*. Ниже приводится список составных частей этого файла:

```
# 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 Explicit 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
```

Пожалуйста, редактируйте файл как описано. Чтобы добавить изменения примените следующую команду:

```
# sysctl -p
```

Соответственно, изменения будут применены после следующей перезагрузки.

Необходимо также упомянуть, что Вы должны включить forwarding (*net.ipv4.ip\_forward*), так как Hardware Node направляет пакеты, предназначенные для или идущие от серверов VPS.

После этого, Вы должны перезагрузить ваш компьютер и выбрать "OpenVZ" на загрузочном меню.

## Загрузка и установка пакетов Open VZ.

После того, как Вы успешно установили и загрузили ядро Open VZ, Вы можете продолжить установку инструментов пользовательского уровня для Open VZ.

Вы должны установить следующие пакеты Open VZ:

- *vzctl*: этот пакет используется, чтобы выполнять различные функции на серверах VPS Open VZ (создает, разрушает, запускает, останавливает, устанавливает параметры и т.д.).
- *vzquota*: этот пакет используется, чтобы управлять долями VPS.
- *vzpkg*: этот пакет используется для работы с шаблонами Open VZ.

Вы можете загрузить соответствующие бинарные RPM на <http://openvz.org/download/utls/>.

Дальше Вы должны установить эти утилиты, выполнив следующую команду:

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

**Замечание:** В течение установки пакетов, Вы можете получить сообщение о том, что *rpm* нашел нерешенные, сомнительные зависимости. В этом случае Вы должны сначала решить эти зависимости, а затем повторить процесс установки.

---

Теперь Вы можете запустить Open VZ. С этой целью, выполните следующую команду:

```
# /etc/init.d/vz start
```

Эта команда запускает все необходимые модули ядра Open VZ. В течение следующей перезагрузки, это будет происходить уже автоматически.

## Установка шаблонов ОС

Шаблон (или набор пакетов) – это ряд файлов пакета, которые будут установлены на сервере VPS. Шаблоны операционной системы используются, чтобы создать новые серверы VPS с предварительно установленной операционной системой. Поэтому, Вы обязаны загрузить, по крайней мере, один шаблон ОС на <http://openvz.org/download/template/> и установить его.

Шаблон metadata ОС содержит информацию, необходимую для создания тайника шаблона (template cache). Вы должны определить, уточнить шаблон ОС для создания сервера VPS на создании VPS, таким образом, Вы должны установить metadata, по крайней мере, для одного шаблона ОС и подготовить тайник шаблона (template cache).

Вот так, например, будет выглядеть подготовка шаблона для Fedora Core 3:

```
# rpm -ihv vztmpl-fedora-core-3-1.0-2.noarch.rpm
Preparing...      ##### [100%]
 1: vztmpl-fedora-##### [100%]
# vzpkgcache
Creating cache for fedora-core-3 OS template
Setting up install process
<...some output skipped for clarity...>
Packing cache file fedora-core-3.tar.gz ...
Cache file fedora-core-3.tar.gz [130M] created.
```

Первая команда устанавливает шаблон metadata, в то время как вторая создает тайник шаблона. Эта операция может занять десятки минут.

Для создания VPS Вы также можете использовать один из уже заранее сохраненных (pre-cached) шаблонов ОС, доступных на <http://openvz.org/download/template/cache/>. С этой целью, Вы должны загрузить соответствующий шаблон ОС и поместить его в директорию `/vz/template/cache` на Node.

## Глава 4

### Операции с серверами VPS

Здесь описывается, как совершать ежедневные операции с серверами VPS в целом.

---

**Замечание:** Итак, Вы успешно установили, сформировали и отрегулировали систему OpenVZ.

Если еще нет, обратитесь к 3 главе, в которой есть полная информация по установке.

---

В этой главе:

1. *Создание и настройка нового сервера VPS*
  2. *Запуск, остановка, перезагрузка, статус запроса (Querying Status) сервера VPS*
  3. *Перечень серверов VPS*
  4. *Удаление серверов VPS*
  5. *Основные команды на серверах VPS*
- 

#### 1. *Создание и настройка нового сервера VPS*

Этот раздел будет направлять Вас в процессе создания сервера VPS. Естественно Вы уже установили систему Open VZ и хотя бы один шаблон ОС. Если у Вас нет ни одного шаблона ОС, установленного на Hardware Node, обратитесь к разделу *Managing Templates* в 1 главе.

#### **Перед началом**

Перед созданием сервера VPS Вы должны:

- проверить имеется / видим ли Hardware Node в Вашей сети. Вы должны иметь возможность соединиться от Hardware Node с другими хост-машинами и наоборот. Иначе, Ваши серверы VPS будут не доступны с других компьютеров.

- проверить есть ли у Вас хотя бы по одному IP адресу на каждый сервер VPS, и есть ли адреса, принадлежащие той же сети, что и Hardware Node, или установлено ли направление к серверам VPS через Hardware Node.

Чтобы создать новый сервер VPS, Вы должны:

- выбрать новый идентификатор сервера VPS (ID VPS);
- выбрать шаблон ОС, чтобы использовать сервер VPS;
- создать непосредственно новый сервер VPS.

## Выбор ID нового сервера VPS

Каждый сервер имеет свой номерной идентификатор, известный также как VPS ID, который представляет этот сервер. ID – это 32-битовый целый номер, начинающийся с 0 и уникальный для всего Hardware Node. Выбирая ID для Вашего, учитывайте, пожалуйста, следующие указания:

- цифра 0 означает Hardware Node. Не стоит стараться создавать сервер с ID 0.
- система Open VZ резервирует ID от 0 до 100. Хотя Open VZ использует только номер 0, различные версии могут использовать дополнительные ID для внутренних нужд. Чтобы установить апгрейд, не создавайте серверы VPS с ID меньше 101.

Единственным строгим требованием для ID конкретного VPS это то, чтобы этот ID был уникальным для данного Hardware Node. Однако если Вы собираетесь иметь несколько компьютеров, работающих с системой Open VZ, мы рекомендуем выбрать различные уровни для номеров ID VPS. Например, на Hardware Node 1 создавайте серверы с ID уровня от 101 до 1000; Hardware Node 2 – 1001 – 2000 и так далее. Так проще запомнить на каком Hardware Node был создан сервер VPS и уменьшит возможность конфликта между серверами, когда сервер мигрирует со одного Node на другой.

Другой подход по присвоения ID адресов серверам состоит в том, чтобы последовать схеме IP адресов. То есть если у Вас есть подсеть с уровнем адресов 10.0.x.x. можно серверу с IP адресом 10.0.15.17 назначить ID 17015, серверу с IP 10.0.39.108 – ID 39108 и так далее. С этой системой легче управлять утилитами системы Open VZ, уменьшается необходимость проверки IP адресов разных серверов по их ID и др. Можете сами придумать систему присвоения адресов в зависимости от конфигурации Вашей сети и специфических потребностей.

Прежде, чем Вы выберете новый ID для VPS, нужно удостовериться, что на Hardware Node нет ни одного VPS с таким ID. Самый легкий способ сделать это выполнить следующую команду:

```
# vzlist -a 101  
VPS not found
```

Вывод: VPS с таким номером не найден, иначе его номер появился бы в командной строке на экране.

## Выбор шаблона ОС

Теперь, Вы должны выбрать, на каком шаблоне ОС будет базироваться Ваш новый VPS. На Hardware Node может быть несколько установленных шаблонов; используйте команду *vzpkgls*, чтобы узнать какие шаблоны установлены на вашей системе:

```
# vzpkgls
fedora-core-3
fedora-core-4
centos-4
```

## Создание VPS

После того, как Вы выбрали ID Вашего VPS и установленный шаблон ОС, можете создать частную область VPS с помощью команды *vzctl*. Частная область - это директория, содержащая частные файлы данного VPS. Частная область установлена в директории */vz/root/vpsid/* на Hardware Node и обеспечивает пользователей VPS полным деревом системных файлов Linux.

Для команды *vzctl* требуются в качестве аргументов только ID VPS и названия шаблона ОС; однако, чтобы избежать установки всех контрольных параметров ресурса VPS после создания частной области, Вы можете определить типовую конфигурацию, которая будет использоваться для вашего нового VPS. Типовые файлы конфигурации находятся в директории */etc/sysconfig/vzscripts* и носят названия под маской: *ve-config\_name.conf-sample*. Наиболее часто используемый шаблон - это файл *ve-vps . basic. conf -sample*; этот файл содержит контрольные параметры ресурса, подходящие для большинства website VPS.

Таким образом, Вы можете создать новый VPS, напечатав следующую строку:

```
# vzctl create 101 --ostemplate fedora-core-4 --config vps.basic
Creating VPS private area
VPS private area was created
```

В этом случае, система Open VZ создаст VPS с ID 101. Частная область в этом случае основана на шаблоне ОС *fedora-core-4*, а параметры конфигурации взяты из файла образца конфигурации *ve-vps . basic. conf-sample*.

Если Вы сами не определите ни шаблон ОС, ни типовую конфигурацию, команда *vzctl* возьмет соответствующие данные (values) с глобального файла конфигурации системы Open VZ */etc/sysconfig/vz*. Так что, Вы можете установить данные умолчанию (default values) в этом файле, используя, ваш любимый текстовый редактор:

```
DEF_OSTEMPLATE="fedora-core-4"
CONFIGFILE="vps.basic"
```



тем самым Вы уходите от необходимости определения этих параметров, каждый раз создавая новый VPS.

Наконец Вы можете создать VPS с ID 101, выполнив следующую команду:

```
# vzctl create 101
Creating VPS private area: /vz/private/101
VPS is mounted
Postcreate action done
VPS is unmounted
VPS private area was created
```

В принципе, на этой стадии Вы уже можете запустить Ваш только что созданный VPS. Однако, перед первым запуском необходимо установить его IP адрес в сети, host name, адрес сервера DNS и корневой пароль. Чтобы выполнить все эти операции обратитесь к следующей секции.

## Настройка VPS

Она состоит из нескольких этапов:

- регулировка параметров запуска VPS;
- регулировка параметров VPS сети;
- регулировка пользовательских паролей;
- настройка параметров качества сервиса - Quality of Service (Service Level).

Для того чтобы выполнить все эти операции используется команда `vzctl set`. О работе этой команды касательно первых 3 пунктов мы расскажем далее. О проблемах управления Service Level будет рассказано в главе *Управление ресурсами (Менеджмент ресурсов)*.

## Регулировка параметров запуска VPS

Следующие варианты команды `vzctl set` определяют параметры запуска VPS: `onboot` и `capability` (возможности). Чтобы сервер VPS 101 запускался автоматически вместе с Hardware Node, выполните следующую команду:

```
# vzctl set 101 --onboot yes --save
Saved parameters for VPS 101
```

## Регулировка параметров VPS сети

Для того, чтобы сервер был доступен из сети, у него должен быть правильный IP адрес и host name; адреса сервера DNS должны быть также сформированы. На рисунке ниже показано как настраиваются параметры сети:

```
# vzctl set 101 --hostname test101.my.org --save
Hostname for VPS set: test101.my.org
Saved parameters for VPS 101
# vzctl set 101 --ipadd 10.0.186.1 --save
Adding IP address(es): 10.0.186.1
Saved parameters for VPS 101
# vzctl set 101 --nameserver 192.168.1.165 --save
File resolv.conf was modified
Saved parameters for VPS 101
```

С помощью этой команды Вы назначите VPS 101 IP адрес *10.0.186.1*, host имя *test101.my.org*, и установите адрес сервера DNS - *192.168.1.165*. Строка *--save* означает, что все параметры в файле конфигурации VPS.

Вы можете выполнить вышеупомянутые команды, когда VPS уже работает. В этом случае, если Вы не хотите, чтобы данные (applied values сохранялись, достаточно просто пропустить строку *--save* и они будут действительны до прекращения работы VPS.

Чтобы проверить, работает ли SSH на VPS, используйте команду *vzctl exec*, которая позволяет выполнить любые команды в сфере VPS:

```
# vzctl start 101
[This command starts VPS 101, if it is not started yet]
# vzctl exec 101 service sshd status
sshd is stopped
# vzctl exec 101 service sshd start
Starting sshd: [ OK ]
# vzctl exec 101 service sshd status
sshd (pid 16036) is running...
```

Предполагается, что VPS 101 в данном случае создан на основе шаблона Fedora Core. При использовании других шаблонов ОС обратитесь к соответствующей документации.

Для получения дополнительной информации о выполнении команд на VPS с компьютера Hardware Node обратитесь к соответствующей секции в разделе Выполнение команд на VPS (Running Commands in Virtual Private Server).

### Установка корневого пароля для VPS

По умолчанию, аккаунт корня находится в только что созданном VPS, и Вы не можете туда попасть. Чтобы это сделать, необходимо создать аккаунт пользователя внутри VPS и установить для него пароль или же открыть корневой аккаунт. Легче всего это сделать с помощью команды:

```
# vzctl start 101
[This command starts VPS 101, if it is not started yet]
# vzctl set 101 --userpasswd root:test
```

В этом примере, мы тестируем пароль корня для VPS 101. Вы можете загрузиться к VPS через

SSH в качестве корня и управлять им таким же образом, как и отдельно взятым компьютером Linux. То есть Вы можете устанавливать дополнительное программное обеспечение, добавлять пользователей, настраивать сервисы, и так далее.

Пароль будет установлен в VPS в файле `/etc/shadow` в зашифрованной форме и не будет сохранен в файле конфигурации VPS. Поэтому, если Вы забудете пароль, Вы должны перезагрузить его.

Имеете в виду, что - - `userpasswd` - единственная разновидность команды `vzctl set`, которая не требует использовать - - `save`, пароль так или иначе постоянно устанавливается для каждого конкретного VPS.

В то время как Вы можете создать пользователей и устанавливать пароли, используя при этом команды `vzctl exec` или `vzctl set`, предполагается, что Вы делегируете пользовательское управление администратору VPS, сообщая ей\ему о пароле корневого аккаунта VPS.

## 2. *Запуск, остановка, перезагрузка, статус запроса (Querying Status) сервера VPS*

Когда VPS уже создан, его может запускать и выключать, как это Вы делаете с обычным компьютером. Чтобы VPS 101, используйте следующую команду:

```
# vzctl start 101
Starting VPS ...
VPS is mounted
Adding IP address(es): 10.0.186.101
Hostname for VPS 101 set: test.my.org
VPS start in progress...
```

Чтобы проверить статус VPS, используйте команду `vzctl status vpsid`:

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

С помощью этой команды Вы узнаете:

- Существует ли частная область VPS;
- Установлена ли эта частная область;
- Работает ли VPS.

В нашем случае, команда `vzctl` сообщает, что VPS 101 существует, его частная область установлена, и VPS работает. Кроме того, Вы команду `vzlist`:

```
# vzlist 101
VPSID      NPROC STATUS  IP_ADDR      HOSTNAME
101        20  running 10.0.186.101 test.my.org
```

Еще один способ проверить статус VPS - проверка файла `/proc/vz/veinfo`. В этом файле отображаются списки всех VPS, работающих в настоящее время Hardware Node. Каждая строка соответствует серверу в формате `<VPS_ID> <reserved> <number of processes> <IP address>`:

```
# cat /proc/vz/veinfo
      101      0      20      10.0.186.1
      0        0      48
```

Этот рисунок показывает, что VPS 101 работает, идет 20 процессов на VPS, и показывает его IP адрес - 192.168.1.1. Замечание: второе поле зарезервировано; оно не имеет никакого особенного значения, но должно всегда равняться нулю.

Последняя строка соответствует VPS с ID 0 и является Hardware Node.

Следующая наиболее распространенная команда – остановка VPS:

```
# vzctl stop 101
Stopping VPS ...
VPS was stopped
VPS is unmounted
# vzctl status 101
VPS 101 exist unmounted down
```

Чтобы закрыть все скрипты VPS, команде `vzctl` требуется 2 минуты. Если VPS не закрывается и через две минуты, система в любом случае закроет VPS, даже если он серьезно поврежден. Если Вы знаете, что кто-то проник в Ваш сервер и его нужно срочно закрыть, не дожидаясь 2 минут, используйте команду быстрого выключения (`--fast switch`):

```
# vzctl stop 101 --fast
Stopping VPS ...
VPS was stopped
VPS is unmounted
```

Убедитесь, что Вы не используете эту команду для выключения серверов, с которыми все в порядке. Знайте, что намеренное быстрое отключение опасно для работоспособности сервера. Команды `vzctl start` и `vzctl stop` наоборот являются командами нормального запуска и выключения Linux OS внутри VPS. В случае дистрибуции как и Red Hat, скрипты инициализации Системы V будут выполнены как и на обычном компьютере. Вы можете настроить скрипты запуска внутри VPS как Вам необходимо.

Чтобы перезапустить VPS, Вы можете также использовать команду `vzctl restart`:

```
# vzctl restart 101
Restarting VPS
Stopping VPS ...
VPS was stopped
VPS is unmounted
Starting VPS ...
VPS is mounted
Adding IP address(es): 10.0.186.101
VPS start in progress...
```

### 3. Перечень серверов VPS

Иногда Вам необходимо получить краткий обзор серверов VPS, существующих на данном Hardware Node, получить о них дополнительную информацию - их IP адреса, hostnames, текущее потребление ресурса, и т.д. В общем, Вы можете получить список всех VPS, используя следующую команду:

```
# vzlist -a
  VPSID      NPROC STATUS  IP_ADDR      HOSTNAME
    101         8 running 10.101.66.1  vps101.my.org
    102         7 running 10.101.66.159 vps102.my.org
    103         - stopped 10.101.66.103 vps103.my.org
```

Переключатель *-a* заставляет утилиту *vzlist* управлять и останавливать серверы VPS. По умолчанию, показаны только работающие серверы VPS. Колонки по умолчанию показывают ID серверов VPS, число процессов внутри серверов VPS, их статус, IP адреса, и hostnames. Этот вывод может быть настроен как желательный, если Вы будете использовать построчную команду *vzlist*. Например:

```
# vzlist -o veid,diskinodes.s -s diskinodes.s
  VPSID DQINODES.S
    1      400000
   101      200000
   102      200000
```

Здесь показаны только работающие серверы VPS + информация об их ID и программных ограничениях лимитах на диске inodes (см. главу Управление ресурсами \ Менеджмент) со списком, отсортированным по этому ограничению. Полный список построчных команды включения\выключения (command line switches), а также список вариантов сортировки Вы можете найти в подсекции *vzlist* главы Ссылки (Reference).

### 4. Удаление серверов VPS

Вы можете удалить VPS, если он больше не нужен, с помощью команды *vzctl destroy VPS\_ID*. Эта команда полностью удаляет частную область сервера VPS, а также переименовывает файл конфигурации VPS и скрипты действия, прибавляя к ним *.destroyed*.

Работающий сервер не может быть уничтожен с помощью команды *vzctl destroy*. На примере мер ниже показано уничтожение сервера VPS 101:

```
# vzctl destroy 101
VPS is currently mounted (umount first)
# vzctl stop 101
Stopping VPS ...
VPS was stopped
VPS is unmounted
# vzctl destroy 101
Destroying VPS private area: /vz/private/101
VPS private area was destroyed
# ls /etc/sysconfig/vz-scripts/101.*
/etc/sysconfig/vz-scripts/101.conf.destroyed
/etc/sysconfig/vz-scripts/101.mount.destroyed
/etc/sysconfig/vz-scripts/101.umount.destroyed
# vzctl status 101
VPS 101 deleted unmounted down
```

Если Вам не нужна резервная копия файлов конфигурации VPS (с суффиксом *.destroyed*), Вы можете удалить их вручную.

## 5. Основные команды на серверах VPS

Обычно, администратор сервера VPS загружается туда через сеть и выполняет на сервере различные также как и на любом Linux box. Однако, иногда необходимо выполнять команды внутри сервера VPS, избегая обычного порядка регистрации. Это может случиться если:

- у Вас нет информации о логине VPS, но должны выполнить некоторые диагностические команды, чтобы проверить эксплуатацию VPS.
- отсутствует доступ сети для VPS. Например, если администратор VPS случайно применил неправильные правила firewalling или остановил SSH daemon.

В этом случае система Open VZ позволяет выполнить следующие команды на сервере. Чтобы управлять процессом внутри VPS с данным ID используйте команду `vzctl exec VPS_ID`. На картинке ниже показана ситуация, когда SSH daemon не запущен.

```
# vzctl exec 101 /etc/init.d/sshd status
sshd is stopped
# vzctl exec 101 /etc/init.d/sshd start
Starting sshd:[ OK ]
# vzctl exec 101 /etc/init.d/sshd status
sshd (pid 26187) is running...
```

Теперь пользователи VPS могут загрузиться к VPS через SSH.

Выполняя команды на VPS от shell scripts, используйте команду `vzctl exec2`. У нее тот же синтаксис, что и у `vzctl exec`, но когда ее используют, она возвращает коды выхода команды вместо кодов `vzctl`. Вы можете проверить код выхода для того, чтобы узнать, завершилась ли команда или

нет.

Если Вы хотите выполнить команду на всех работающих VPS, используйте следующий скрипт:

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

В этом случае *<command>* - это команда, которая должна быть выполнена на всех работающих VPS.

Например:

```
# for i in `vzlist -o veid -H`; do\  
echo "VPS $i"; vzctl exec $i uptime; done  
VPS 101  
  2:26pm up 6 days,  1:28,  0 users,  load average: 0.00, 0.00, 0.00  
VPS 102  
  2:26pm up 6 days,  1:39,  0 users,  load average: 0.00, 0.00, 0.00  
[The rest of the output is skipped...]
```

## Глава 5

### Управление Шаблонами

Шаблон – 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, для загрузки и функционирования системы (VPS). Такие приложения как компилятор и сервер SQL обычно не включаются в шаблон ОС.

*В этой главе:*

- 1. Жизненный цикл шаблонов (Lifecycle)**
- 2. Внесение шаблонов в список**
- 3. Работа с VPS**

---

### **1. Жизненный цикл шаблонов (Lifecycle)**

Template cache - шаблон ОС, установленный в VPS и затем упакованный в gzipped tar архив. Это позволяет ускорить создание нового Сервера. Вместо установки всех пакетов, включающих распределение Linux, vzctl распаковывает только архив.

Template metadata - ряд файлов, содержащих информацию для восстановления template cache. Он содержит следующую информацию:

- Список пакетов, которые включает этот шаблон
- Местоположения пакетов в сети
- Скрипты, которые должны быть извлечены на разных стадиях установки шаблона
- Общественный ключ (и) GPG, необходимые для проверки подписей пакетов
- Дополнительные пакеты, специфичные для OpenVZ

Чтобы работать с шаблоном, Вы должны сначала создать его metadata (доступную на <http://openvz.org/download/template/metadata/>). Чтобы создать template cache нужно запустить vzpkgcache утилиту. Она загружает из сети все пакеты для данного распределения, которые включает этот шаблон и устанавливает эти пакеты во временный VPS, который затем архивируется для дальнейшего использования при создании нового VPS.

Так как этот процесс предполагает загрузку большого количества файлов из интернета (для типичного распределения приблизительно 400 файлов объемом до 200 МБ), он (процесс) мог бы быть ускорен, используя снимки уже принесенных для данного распределения сетевых репозитов. Такие снимки доступны на <http://openvz.org/download/template/reposcache/>. Они должны быть



распакованы в `/vz/template` директорию. Пожалуйста имейте в виду, что этот шаг является дополнительным.

В случае, если `template cache` (то есть `tar` архив) уже существует, `vzpkgcach` попытается включить его в данные, применяя последние обновления, доступные для распределения. Так как в настоящее время распределения Linux обновляются регулярно, разумно запускать эту утилиту хотя бы время от времени, или по крайней мере перед созданием массовых VPS.

Если нет никакой потребности обрабатывать все шаблоны, их названия могут быть определены после `vzpkgcache` в командной строке. Например, следующая команда создает или обновляет тайник только для шаблона `Fedora Core 4`:

```
# vzpkgcache fedora-core-4
```

Когда `template cache` готов, он может быть установлен на VPS или, другими словами, VPS может быть создан на основе шаблона. Это выполняется с помощью команды `vzctl create--ostemplate name`, где `name` - название шаблона (например `fedora-core-4`).

## 2. *Внесение шаблонов в список*

`vzpkgls` утилита позволяет Вам вносить в списки шаблоны, установленные на `hardware node`. Они могут быть уже использованы или не использованы определенными VPS:

```
# vzpkgls
fedora-core-4
centos-4
```

Как Вы видите, `fedora-core-4` и `centos 4` шаблоны доступны на `hardware node`. Имейте в виду, что некоторые из них могли и не быть сохранены, спрятаны (`cached`) заранее. Чтобы увидеть только те шаблоны, которые были спрятаны (и таким образом готовы для создания VPS), используйте `-cache` флажок с `vzpkgls`:

```
# vzpkgls --cached
fedora-core-4
```

Данная схема означает, что centos-4 шаблон только что установлен, но еще не скрыт.

При определении номера VPS в качестве параметра, следующая команда печатает шаблон, используемый определенным VPS:

```
# vzpkgls 101
```

```
fedora-core-4
```

### 3. Работа с VPS

Если Вам нужно обновить уже существующий VPS новыми пакетами, доступными на репозиториях или установить те пакеты, которые не являются частью шаблона, используйте команду `vzyum`, которая является простой `yum` оберткой. Например, чтобы обновить VPS с идентификационным номером 123 запустите:

```
# vzyum 123 update
```

Она найдет, загрузит, и установит все доступные обновления.

Как вы могли заметить, большинство приложений не установлено в шаблонах операционной системы. Они должны быть установлены отдельно, используя `vzyum` утилиту. Например, если вам нужен сервер MySQL внутри VPS 123, используйте следующую команду:

```
# vzyum 123 install mysql-server
```

`vzyum` вызовет менеджера `yum` пакета и предоставит ему все пути к репозиториям, подходящим для распределения, установленного на данном VPS. `Yum` вычислит зависимости, представит вам список пакетов для установки/обновления/удаления, основанные на ваших запросах и, в случае подтверждения, осуществит перенос, чтобы фактически выполнить все необходимые операции. Для получения подробной информации смотрите нужную страницу `yum` данной брошюры (`man 8 yum`).

Если пакет, который вы хотите установить, уже доступен на `hardware node`, можно использовать `Vzrpm` утилиту, чтобы установить его (её?) на VPS, например:

```
# vzrpm 123-ihv mypackage-1.0-2.i386.rpm
```

Она установит туpackage RPM на VPS 123.

Конечно, вы можете выполнять все обычные операции прямо внутри любого VPS: строить, устанавливать, модернизировать, и удалять программное обеспечение.

## ***Глава 6***

### ***Управление Ресурсами***

Главная цель контроля ресурсов в OpenVZ состоит в том, чтобы обеспечить уровень управления обслуживания или качество обслуживания (QoS) для серверов. Правильные настройки контроля ресурсов предотвращают серьезные последствия, появляющиеся вследствие чрезмерного использования (случайного или умышленного) ресурсов любым частным сервером на других серверах. Использование параметров управления контролем ресурсов для качественного сервисного обслуживания также позволяет добиться правильного использования ресурсов среди частных серверов и лучшего качества обслуживания для конкретных VPS.

*В этой главе:*

1. *Каковы параметры контроля ресурсов*
  2. *Управление дисковыми квотами \ долями*
  3. *Управление CPU Share*
  4. *Управление системными параметрами*
  5. *Управление конфигурацией ресурсов VPS*
- 

### 1. *Каковы параметры контроля ресурсов?*

Системный оператор контролирует доступные частному серверу ресурсы с помощью набора параметров управления ресурсами. Все эти параметры определены либо в глобальном файле конфигурации (/etc/sysconfig/vz) OpenVZ , либо в соответствующих файлах конфигурации VPS (/etc/sysconfig/vz-scripts/VPSID.conf), либо в обоих. Вы можете установить их вручную, редактируя соответствующие файлы конфигурации или используя утилиты командных линий OpenVZ. Эти параметры могут разделяться на дисковые, сетевые, CPU и системные категории.

Эти группы суммируются в таблице:

<b>Группа</b>	<b>Описание</b>	<b>Названия параметра</b>	<b>Объяснено в</b>
Диск	Эта группа параметров определяет фрагментацию диска в OpenVZ. Фрагментация диска в OpenVZ реализуется на двух уровнях: на VPS уровне и на уровне пользователя/группы. Вы можете вкл\откл дисковую фрагментацию/квоту на любом уровне и установить ее настройки.	DISK_QUOTA, DISKSPACE, DISKINODES, QUOTATIME, QUOTAUGIDLIMIT	Управлении фрагментацией диска

CPU	Эта группа параметров определяет время CPU, которое гарантированно получают различные VPSs	VE0CPUUNITS, CPUUNITS	Управлении разделением CPU
Система	Эта группа параметров определяет аспекты использования различными VPS системной памяти, гнезд TCP, IP пакетов и других, подобных параметров.	avnumproc, numproc, numtcpsock, numothersock, vmguarpages, kmemsize, tcpsndbuf, tcprcvbuf, othersockbuf, dgramrcvbuf, oomguarpages, lockedpages, shmpages, privvmpages, physpages, numfile, numflock, numpty, numsiginfo, dcachesize, numiptent	Управлении системными параметрами

## 2. Управление дисковыми разделами / долями - квотами

Эта секция объясняет, что такое дисковые доли, определяет параметры дисковых долей и описывает, как выполнять всевозможные операции, связанные с дисковыми долями:

- Включение и отключение дисковых долей (первого уровня) на VPS;
- Настройка дисковых параметров дисковой доли первого уровня на VPS;
- Включение и отключение дисковых долей (второго уровня) пользователя/группы на VPS;
- Настройка долей второго уровня для пользователя и/или для группы;
- Проверка статистики дисковых долей;
- Проверка состояния VPS в определенных случаях.

### Что такое дисковая доля, квота?

Дисковые доли позволяют системным администраторам контролировать размер системных файлов Linux, ограничивая количество дискового пространства и числа inodes, которое может

использовать сервер. Эти доли известны в OpenVZ как per – VPS (на каждый сервер) доли или доли первого уровня. Кроме того, OpenVZ позволяет администратору сервера ограничить количество дискового пространства и числа inodes, которые могут использовать индивидуальные пользователи и группы VPS. Эти доли называются долями пользователя или группы или долями второго уровня.

По умолчанию, в OpenVZ установлены доли первого уровня (который определен в основном в файле конфигурации OpenVZ), тогда как доли второго уровня должны быть включены для каждого частного сервера отдельно (в соответствующих файлах конфигурации VPS). Невозможно включить дисковые доли второго уровня для частного сервера, если для него выключены дисковые доли первого уровня.

Размер блока дисковой доли в OpenVZ всегда 1024 байта. Он может отличаться от размера блока выделенного системного файла.

OpenVZ содержит статистику использования доли, которая находится в специальном файле доли /var/vzquota/quota.vpsid. Файл доли имеет специальный флажок «dirty» - загрязненный файл. Файл становится «грязным», когда его содержание несовместимо с реальным использованием VPS. Это означает что, когда дисковое пространство или использование inodes меняется в процессе использования VPS, эти статистические данные автоматически не синхронизируются с файлом доли, и файл маркируется как "грязный". Они синхронизируются только, когда VPS остановлен или когда HN закрыт. После синхронизации, флажок удаляется. Если Hardware Node был неправильно закрыт (например, скачок мощности), файл остается "грязным", и доля переинициализируется при последующем запуске VPS. Эта операция может заметно увеличить время запуска Node. Таким образом, особенно рекомендуются закрывать Hardware Node надлежащим образом.

### Параметры дискового раздела.

Таблица ниже суммирует параметры дисковых долей, которыми вы можете управлять. Колонка Файла указывает, определен ли параметр в (G) - основном файле конфигурации OpenVZ, или в (V) - файлах конфигурации VPS, или же он определен в глобальном файле конфигурации, но не поддерживается в конкретном файле конфигурации VPS (VG).

Параметр	Описание	Файл
disk_quota	Указывает, включены или отключены ли доли первого уровня для всех VPS или для какого-то отдельно-взятого VPS.	VG
diskspace	Полный размер дискового пространства, которое может потреблять VPS, в блоках на 1 КБ.	V
diskinodes	Общее количество дисковых inodes (файлы, справочники, и символичные связи), которые может ассигновать VPS.	V

quotatime	Период grace для сверхиспользования доли диска, измеряемого в секундах. VPS разрешается на некоторое время превышать его пределы долю software, но лишь на период не дольше ограниченного в QUOTATIME.	V
quotauidlimit	Число ID пользователя/группы, разрешенных для внутреннего использования дисковых долей на VPS. Если эта величина равна 0, доля UID/GID не будет запускаться.	V

### **Включение и отключение на VPS дисковые доли первого уровня.**

Параметр, который определяет, использовать ли дисковые доли первого уровня, - `DISK_QUOTA` находится в основном файле конфигурации OpenVZ (`/etc/sysconfig/vz`). Установив его в позиции «нет», вы сделаете доли OpenVZ не действительными.

Этот параметр может быть также определен в файле конфигурации VPS - (`/etc/sysconfig/vz-scripts/vpsid.conf`). В этом случае его величина будет восходить к той, что была определена в основном файле конфигурации. Если Вы намереваетесь поддерживать работу нескольких VPS с включенными или выключенными долями, рекомендуется установить значение `DISK_QUOTA` в позицию "да" в основном файле конфигурации и в позицию "Нет" в файле конфигурации того VPS, который не нуждается в данной доле.

Схема иллюстрирует сценарий, когда доли первого уровня включены по умолчанию и отключены на конкретном VPS 101:

```
[checking that quota is on]
# grep DISK_QUOTA /etc/sysconfig/vz
DISK_QUOTA=yes
[checking available space on /vz partition]
# df /vz
Filesystem 1k-blocks Used Available Use% Mounted on
/dev/sda2 8957295 1421982 7023242 17% /vz
[editing VPS configuration file to add DISK_QUOTA=no]
# vi /etc/sysconfig/vz-scripts/101.conf
[checking that quota is off for VPS 101]
# grep DISK_QUOTA /etc/sysconfig/vz-scripts/101.conf
DISK_QUOTA=no
# vzctl start 101
Starting VPS ...
VPS is mounted
Adding IP address(es): 192.168.1.101
Hostname for VPS set: vps101.my.org
VPS start in progress...
# vzctl exec 101 df
Filesystem 1k-blocks Used Available Use% Mounted on
simfs 8282373 747060 7023242 10% /
```

Как показывает пример выше, единственное ограничение дискового пространства, которое имеет VPS с отключенной функцией дисковой доли, - это доступное место и inodes на том разделении VPS, где локализуется его частная область.

**Замечание:** Вы должны менять параметр DISK\_QUOTA в основном файле конфигурации OpenVZ только, когда все VPS остановлены, и в файле конфигурации VPS – только тогда, когда соответствующий VPS остановлен. Иначе, конфигурация может конфликтовать с величиной реального использования доли, что в свою очередь, будет мешать нормальному функционированию hardware node.

### **Настройка параметров дисковой доли VPS первого уровня.**

Существует три параметра, которые определяют, сколько дискового пространства и inodes может использовать VPS.

Эти параметры определены в файле конфигурации VPS:



## **DISKSPACE**

Общий размер дискового пространства, которое может потребляться VPS, измеряемый в 1- КБ блоках. Когда место, используемое VPS превышает предел, установленный на software, VPS может ассигновать дополнительное дисковое пространство до предела, установленного на hardware в течение всего периода grace, определенного параметром QUOTATIME.

## **DISKINODES**

Общее количество дисковых inodes (файлы, справочники, и символичные связи), которые может ассигновать VPS . Когда число inodes, используемых VPS превышает предел, установленный на software, VPS может создать дополнительные записи файла до предела, установленного на hardware в течение всего периода grace, определенного параметром QUOTATIME .

## **QUOTATIME**

Период grace дисковой доли, измеряемый в секундах. VPS может на какое-то время превысить величину предела и дисковых inodes квот, установленных на software, но не дольше, чем тот период, который определен этим же параметром.

Первые два параметра имеют как hard так и soft пределы (или, просто, пределы и барьеры /limits and barriers ).

Hard предел - предел, который не может быть превышен ни при каких обстоятельствах. Soft предел может быть превышен до hard предела, но по истечению периода grace, дополнительное дисковое пространство или распределения inodes прекратят свое существование. Барьеры и пределы отделены двоеточиями («:») в файлах конфигурации сервера и в командных строках.

Следующая схема определяет дисковое пространство, доступное для VPS 101, равное приблизительно 1Gb, и позволяет VPS ассигновать до 90 000 inodes. Период grace для доли равен десяти минутам:

```
# vzctl set 101 --diskspace 1000000:1100000 --save
Saved parameters for VPS 101
# vzctl set 101 --diskinodes 90000:91000 --save
Saved parameters for VPS 101
# vzctl set 101 --quotatime 600 --save
VPS 101
Saved parameters for # vzctl exec 101 df
Filesystem 1k-blocks Used Available Use% Mounted on
000 747066 252934 75% /
simfs 1000# vzctl exec 101 stat -f /
File: "/"
ID: 0 Namelen: 255 Type: ext2/ext3
Blocks: Total: 1000000 Free: 252934 Available: 252934 Size: 1024
Inodes: Total: 90000 Free: 9594
```

Для работающего VPS возможно изменять дисковые параметры долей первого уровня. Изменения вступят в силу немедленно. Если Вы не хотите, чтобы ваши изменения сохранились при следующем запуске VPS – не используйте переключатель - switch

### **Включение и отключение долей VPS второго уровня.**

Параметр, который управляет дисковыми долями второго уровня, - QUOTAUGIDLIMIT, находящийся в файле конфигурации VPS. По умолчанию, величина этого параметра - ноль, что соответствует отключенной (disabled) per-user/group квоте.

Если Вы присписываете этому параметру QUOTAUGIDLIMIT величину, отличную от нулевой, это может привести к двум следующим результатам:

1. Дисковые доли второго уровня (per-user/group ) валидны для данного сервера
2. Величина, которую вы присписываете этому параметру, будет ограничивать число групп и владельцев этого файла на данном VPS, включая пользователей системы Linux. Имейте в виду, что теоретически Вы будете способны создавать дополнительных пользователей этого VPS, но если число владельцев файла в VPS уже достигло своего предела, эти пользователи просто будут не в состоянии использовать данные файлы.

Предоставление возможности существования per-user/group квот на VPS требует перезагрузки VPS. Для этого должна быть тщательно выбрана величина; чем большую величину Вы устанавливаете, тем большую ядерная память создает этот VPS. Эта ценность должна быть больше

или равна числу записей в VPS файлах /etc/passwd и/etc/group. Принимая во внимание, что недавно созданный на основе Linux VPS Red Hat имеет приблизительно всего 80 записей , типичная величина должна равняться приблизительно 100. Тем не менее, для серверов с большим количеством пользователей эта величина должна быть значительно увеличена.

Сессия ниже позволяет осуществить включение долей второго уровня для VPS 101:

```
# vzctl set 101 --quotauidlimit 100 --save
Unable to apply new quota values: ugid quota not initialized
Saved parameters for VPS 101
# vzctl stop 101; vzctl start 101
Stopping VPS ...
VPS was stopped
VPS is unmounted
Starting VPS ...
VPS is mounted
Adding IP address(es): 192.168.1.101
Hostname for VPS set: vps101.my.org
VPS start in progress...
```

## Настройка параметров для дискового раздела второго уровня

Для того чтобы работать с дисковыми долями на самом VPS, Вы должны установить стандартный набор инструментов

```
# vzctl exec 101 rpm -q quota
quota-3.12-5
```

Эта команда показывает, что пакет quota установлен на вашем VPS. Используйте утилиты из этого пакета (как указано в вашем руководстве Linux), для того, чтобы установить доли второго уровня для данного VPS OpenVZ. Например:

```

# ssh ve101
root@ve101's password:
Last login: Sat Jul 5 00:37:07 2003 from 10.100.40.18
[root@ve101 root]# edquota root
Disk quotas for user root (uid 0):
Filesystem blocks soft hard inodes soft hard
/dev/simfs 38216 50000 60000 45454 70000 70000
[root@ve101 root]# repquota -a
*** Report for user quotas on device /dev/simfs
Block grace time: 00:00; Inode grace time: 00:00
Block limits File limits
User used soft hard grace used soft hard grace
-----
root -- 38218 50000 60000 45453 70000 70000
[the rest of repquota output is skipped]
[root@ve101 root]# dd if=/dev/zero of=test
dd: writing to `test': Disk quota exceeded
23473+0 records in
23472+0 records out [root@ve101 root]# repquota -a
*** Report for user quotas on device /dev/simfs
Block grace time: 00:00; Inode grace time: 00:00
Block limits File limits
User used soft hard grace used soft hard grace
-----
root +- 50001 50000 60000 none 45454 70000 70000
[the rest of repquota output is skipped]

```

Этот пример отображает сессию команд, в случае, когда root пользователь устанавливает долю дискового пространства на hard предел 60 000 1 КБ блоков и на soft предел 50 000 1 КБт блоков; Необходимо отметить, что как hard так и soft ограничивают число установленных inodes до 70 000.

Также возможно установить grace период отдельно для блоковых ограничений и ограничений inodes с помощью команды /usr/sbin/setquota. Для получения дополнительной информации об использовании утилит пакета quota, обращайтесь к гиду управления системой прилагаемой к распределению Linux или страницам ручной настройки, включенным в пакет.

### Проверка статуса дисковой доли

Как и системный администратор `hardware node`, Вы можете проверить статус доли для любого VPS при помощи команд `vzquota stat` and `vzquota show`. Первая команда информирует вас о статусе ядра и используется для работающего VPS; вторая команда сообщает о статусе, используя при этом файл доли (расположенный в `/var/vzquota/quota.vpsid`) и будет использоваться для остановленных VPS. Обе команды имеют один и тот же формат.

Сессия ниже показывает типичный вид статистических данных для доли VPS 101

```
# vzquota stat 101 -t
resource usage softlimit hardlimit grace
1k-blocks 38281 1000000 1100000
inodes 45703 90000 91000
User/group quota: on,active
Ugids: loaded 34, total 34, limit 100
Ugid limit was exceeded: no
User/group grace times and quotafile flags:
type block_exp_time inode_exp_time dqf_flags
user 0h
group 0h
User/group objects:
ID type resource usage softlimit hardlimit grace status
0 user 1k-blocks 38220 50000 60000 loaded
0 user inodes 45453 70000 70000 loaded
[the rest is skipped]
```

Первые три линии этой сессии команд показывают статус дисковых долей первого уровня для VPS. Остальная часть отображает статистику для долей пользователя/группы и имеет отдельные строки для каждого пользователя и группы ID, существующих в системе.

Если Вы не нуждаетесь в статистике доли второго уровня, Вы можете убрать выключатель `-t` в командной строке `vzquota`.

### 3. Управление разделом (долей, квотой) CPU

В этой секции дается информация относительно параметров ресурсов центрального процессора (CPU) (раздел CPU), для того, чтобы Вы могли осуществлять формирование и контроль вашего сервера.

Таблица ниже снабдит Вас названиями и описаниями параметров центрального процессора. Колонка FILE указывает, значится ли параметр в (G) - основном файле конфигурации OpenVZ

или в (V) - файлах конфигурации VPS.

Parameter -параметр	Description - описание	File-файл
ve0cpuunits	Это - положительное целое число, определяющее гарантируемое минимальное количество времени, которое получит центральный процессор VPS 0 (непосредственно hardware node). Рекомендуют установить величину этого параметра не более 5-10 % от мощности hardware node.	G
cpuunits	Это - положительное целое число, определяющее гарантируемое минимальное количество времени CPU, которое получит соответствующий VPS	V
cpulimit	Это - положительное число, в процентах указывающее время центрального процессора, которое не позволено превышать VPS	V

Утилиты контроля ресурсов центрального процессора CPU OpenVZ гарантируют любому VPS количество времени центрального процессора, которое собственно и получает этот VPS. При этом VPS может потреблять больше времени, чем определено этой величиной, если нет другого конкурирующего с ним за время CPU сервера или, в случае, когда не задан cpulimit параметр.

Чтобы составить представление об оптимальной величине доли, которая должна быть назначена тому или иному VPS, проверьте текущее использование CPU на hardware node:

```
# vzcpucheck
```

```
Current CPU utilization: 5166
```

```
Power of the node: 73072.5
```

Выполнение этой команды показывает общее количество так называемых единиц CPU, потребляемых всеми работающими в данный момент серверами, а также процессами, запущенными на hardware node. Это число вычисляется в OpenVZ с помощью специального алгоритма. Вышеупомянутый пример иллюстрирует ситуацию, в которой Hardware Node недогружен. Другими словами, все запущенные серверы получают больше времени центрального процессора, чем гарантировалось им ранее.

В следующем примере, VPS 102, по заданной схеме, получит приблизительно 2 % от времени CPU, даже если hardware node полностью используется, или другими словами, если использование центрального процессора в текущий момент равняется мощности hardware node. Кроме того, даже если центральный процессор полностью не загружен, VPS 102 не будет получать больше чем 4 %

от времени CPU.

```
# vzctl set 102 --cpuunits 1500 --cpulimit 4 --save
```

```
Saved parameters for VPS 102
```

```
# vzctl start 102
```

```
Starting VPS ...
```

```
VPS is mounted
```

```
Adding IP address(es): 192.168.1.102
```

```
progress...
```

```
VPS start in # vzcpucheck
```

```
Current CPU utilization: 6667
```

```
Power of the node: 73072.5
```

VPS 102 получит от 2 до 4 % времени CPU, если hardware node не перегружен. (Например, если работающим серверам обещалось больше единиц CPU, чем может позволить мощность самого hardware node.) В этом случае VPS может получить меньше чем 2 процента гарантируемого времени.

#### ***4. Управление системными параметрами***

Ресурсы, которые VPS может выделить, определены в параметрах управления системными ресурсами. Эти параметры можно разделить на следующие категории: первичные, вторичные, и вспомогательные параметры.

*Первичные параметры* – отправной пункт для того, чтобы создать конфигурацию VPS на пустом месте.

*Вторичные параметры* зависят от первичных и вычисляются на их основе с рядом ограничений.

*Вспомогательные параметры* помогают улучшить изоляцию ошибок среди приложений на одном и том же VPS, а также усовершенствовать способ, которым приложения сервера обращаются с ошибками и потребляют ресурсы сервера. Они также помогают администратору реализовать управление сервером таким образом, чтобы ограничить ресурсы, запрашиваемые приложениями и не дать приложению запуститься на VPS

Все параметры управления системными ресурсами перечислены ниже. Параметры, начинающиеся с "num" измеряются в целых числах. Параметры, заканчивающиеся на "buf" или "size" измеряются в байтах.

Параметры, содержащие в своих названиях "page", измеряются в 4096-байтовых страницах (IA32 архитектура).

Колонка «Файл» указывает, что все параметры системы определены в (V) - соответствующих файлах конфигурации VPS.

## Первичные параметры

<i>Параметр</i>	<i>Описание</i>	<i>Файл</i>
avnumproc	Среднее число процессов и нитей.	V
numproc	Максимальное число процессов и нитей, которые VPS может создать.	V
numtcpsock	Число гнезд TCP (PF_INET семьи, SOCK_STREAM типы). Этот параметр ограничивает число связей TCP и, таким образом, число клиентов, с которым данное приложение может обращаться в одно и то же время.	V
numothersock	Число гнезд, отличных TCP. Местные гнезда (UNIX domain) используются для коммуникаций в системе. Гнезда UDP используются, например, для вопросов Обслуживания Названия Области (DNS). UDP и другие гнезда могут также использоваться в некоторых очень специализированных приложениях (SNMP agents и другие).	V
vmguarpages	Гарантия распределения памяти, в страницах (одна страница - 4 КБ). VPS приложения гарантировано в состоянии ассигновать дополнительную память столь долго, сколько потребуется, чтобы количество памяти, подсчитанной в privvmpages (см.вспомогательные (auxiliary) параметры), не превышало формируемый барьер параметров vmguarpages. Если этот барьер превзойден, распределение дополнительной памяти не гарантируется и может привести к неудаче в случае полной нехватки памяти.	V

## Вторичные параметры

<b>Параметр</b>	<b>Описание</b>	<b>Файл</b>
kmemsize	Размер unswappable ядерной памяти, ассигнованной для внутренних ядерных структур для процессов на отдельно взятом VPS.	V



tcpwndbuf	Полный размер send буферов для гнезд TCP, то есть количество ядерной памяти, ассигнованной для данных, посланных приложениями на гнезда TCP, но все еще не признанных удаленной стороной.	V
tcprcvbuf	Полный размер receive буферов для гнезд TCP, то есть количество ядерной памяти, ассигнованной для данных, полученных удаленной стороной, но не прочитанных местными приложениями.	V
othersockbuf	Полный размер буферов гнезд UNIX domain, UDP, и других datagram протоколов send буфера .	V
dgramrcvbuf	Полный размер receive буферов UDP и других datagram протоколов	V
oomguarpages	Гарантия, не включенная в память, в страницах (одна страница - 4 КБ). Любой VPS процесс не будет остановлен даже в случае сильной нехватки памяти, даже если текущее потребление памяти (включая физическую память и swap), не достигает барьера oomguarpages.	V

### Вспомогательные параметры

Параметр	Описание	Файл
lockedpages	Память, которую нельзя поменять (заперта в mlock () запросах системы), в страницах.	V
shmpages	Полный размер совместно используемой памяти (включая IPC, общие анонимные картографии и объекты tmpfs) ассигнованной процессами конкретного VPS, в страницах.	V
privvmpages	Размер частной (или потенциально частной) памяти, ассигнованной приложениями. Память, которая обычно разделяется среди различных приложений, в этот параметр ресурса не включается.	V
numfile	Число файлов, открываемых всеми процессами VPS.	V
numflock	Число файловых замков, созданных всеми процессами VPS.	V
numpty	Число псевдотерминалов, типа ssh сессий, screen или xterm приложений и т.д.	V

numsiginfo	Число siginfo структур (по существу, ограничение размера параметров по очереди поставки сигнала).	V
dcachesize	Полный размер структур dentry и inode, которые ограничены памятью.	V
physpages	Полный размер RAM, используемой процессами VPS.В настоящее время. Этот параметр используется только для подсчета учета. Он показывает, как VPS использует RAM. Для страниц памяти, используемых несколькими VPS (например, картографии общих библиотек) изменяется только соответствующая фракция страницы каждого VPS. Сумма использования physpages для всех VPS соответствует общему количеству страниц, используемых в системе всеми подсчитанными пользователями.	V
numiptent	Число пакетов IP, фильтрующих записи.	V

Вы можете редактировать любой из этих параметров в файле */etc/sysconfig/vz-scripts/vpsid.conf* соответствующего VPS с помощью вашего любимого редактора текста (например, vi или emacs), или управляя командой набора vzctl. Например:

```
# vzctl set 101 --kmemsize 2211840:2359296 --save
Saved parameters for VPS 101
```

## Контроль потребления системных ресурсов

Внутри VPS можно проверять статистику параметров контроля системных ресурсов. Основное использование этих статистических данных служит для того, чтобы понять то, какой именно из этих ресурсов имеет ограничения, препятствующие запуску приложений сервера. Кроме того, эти статистические данные сообщают о текущем и максимальном потреблении ресурсов для функционирующего VPS. Эта информация может быть получена из файла */proc/user\_beancounters*.

Ниже мы можем видеть типичный вид такой сессии:

```
# vzctl exec 101 cat /proc/user_beancounters
Version: 2.5
uid resource held maxheld barrier limit failcnt
101: kmemsize 803866 1246758 2457600 2621440 0
lockedpages 0 0 32 32 0
privvmpages 5611 7709 22528 24576 0
shmpages 39 695 8192 8192 0
dummy 0 0 0 0 0
numproc 16 27 65 65 0
physpages 1011 3113 0 2147483647 0
vmguarpages 0 0 6144 2147483647 0
oomguarpages 2025 3113 6144 2147483647 0
numtcpsock 3 4 80 80 0
numflock 2 4 100 110 0
numpty 0 1 16 16 0
numsiginfo 0 2 256 256 0
tcpsndbuf 0 6684 319488 524288 0
tcpvbuf 0 4456 319488 524288 0
othersockbuf 2228 9688 132096 336896 0
dgramrcvbuf 0 4276 132096 132096 0
numothersock 4 17 80 80 0
dcachesize 78952 108488 524288 548864 0
numfile 194 306 1280 1280 0
dummy 0 0 0 0 0
dummy 0 0 0 0 0
dummy 0 0 0 0 0
numiptent 0 0 128 128 0
```

`failcnt` колонка указывает число неудачных попыток ассигновать какой-то ресурс. Если эти числовые данные увеличиваются после запуска приложений, тогда соответствующий предел ресурса имеет меньший эффект, чем тот, который необходим приложению.

`held` колонка показывает текущее использование ресурсов, а `maxheld` колонка – максимальную величину потребления ресурса для последнего периода, в который проводился подсчет. Значение колонок `barrier` и `limits` зависят от параметров и объяснено в гиде [OpenVZ Management of System Resources](#).

Внутри VPS файл `/proc/user_beancounters` отображает информацию относительно только данного

VPS, тогда как на hardware node этот файл показывает информацию относительно всех VPS.

Чтобы проверить использование UBC для HN (резюме для всех работающих VPS), Вы можете использовать следующий скрипт:

(для любого ресурса, подсчитанного на страницах)

```
# for res in lockedpages totvmpages ipcshmpages anonshpages rsspages;\
do echo;echo "$res usage for all VEs, in MB:";cat \
/proc/user_beancounters |grep $res|awk 'BEGIN{ cur=max=lim=0; } \
{ cur+=$2; max+=$3;lim+=$5 } END {print "held:",cur*4/1024, "max:", \
max*4/1024, "limit:", lim*4/1024}'; done
```

(поскольку kmemsize и другие ресурсы считали в байтах),

```
# for res in tcpseendbuf tcprecvbuf unixsockbuf sockrcvbuf kmemsize; \
do echo;echo "$res usage for all VEs, in MB:";cat \
/proc/user_beancounters |grep $res|sed "s/[[:digit:]]\+//g" \
|awk 'BEGIN{ cur=max=lim=0; } { cur+=$2; max+=$3;lim+=$5 } \
END {print "held:",cur/1024/1024, "max:", max/1024/1024, \
"limit:", lim/1024/1024}'; done
```

## Контроль потребления ресурсов памяти

Вы можете контролировать множество параметров памяти как для целого hardware node так и для отдельно взятого VPS с помощью утилиты `vzmemcheck` . Например:

```
# vzmemcheck -v
Output values in %
veid LowMem LowMem RAM MemSwap MemSwap Alloc Alloc Alloc
util commit util util commit util commit limit
101 0.19 1.93 1.23 0.34 1.38 0.42 1.38 4.94
1 0.27 8.69 1.94 0.49 7.19 1.59 2.05 56.54
-----
Summary: 0.46 10.62 3.17 0.83 8.57 2.02 3.43 61.48
```

-v опция используется, чтобы показать информацию о состоянии памяти для каждого VPS, но не для hardware node в целом. Также при использовании переключателя -A возможно отобразить абсолютные величины в мегабайтах. Проверяемые параметры - (слева направо в вышеуказанном примере) низкое использование памяти, низкое обязательство памяти, использование RAM, использование memory+swap, memory+swap обязательства, использование ассигнованной памяти, ассигнованных обязательств памяти, ассигнованный предел памяти.

Чтобы разобраться с этими параметрами, позвольте нам сначала привести различие между уровнями использования и обязательств.

*Уровень использования (utilization)* - количество ресурсов, потребляемых VPSs в данное время. В целом, низкие показатели этой величины указывает на то, что систему используют недостаточно. Часто, это означает то, что система способна поддерживать большее количество серверов при таком же использовании ресурсов памяти уже существующими. Большие величины использования (в общих показателях, превышающие 1, или 100 %), означают, что система перегружена и уровень обслуживания VPS ухудшен.

*Уровень обязательств (commitment)* показывает, сколько ресурсов "обещано" существующему VPS. Низкий уровень обязательств означает, что система способна поддерживать большее количество серверов. Уровень обязательства больший 1 подразумевает, что

серверам обещано больше ресурсов, чем имеется в системе, и система, как говорят, переполнена. Если система управляет большим количеством VPS, некоторая перегрузка допустима, потому что маловероятно, что все VPS будут запрашивать ресурсы одновременно. Однако, завышенный уровень обязательств приведет к фактической неудаче ассигновать и использовать ресурсы, обещанные им. Такая ситуация может повредить стабильность системы.

Далее следует краткий обзор, всех ресурсов, проверяемых утилитой `vzmemcheck`. Их полное описание дается в гиде `OpenVZ Management of System Resources` Управление системными ресурсами OpenVZ. Низкая/малая память - самая важная область RAM, представляющая часть памяти, находящейся в низко располагающихся адресах и доступная непосредственно от самого ядром. В OpenVZ размер области "низкой" памяти ограничен 832 MB в (uniprocessor) и версией ядра SMP, и 3.6 ГИГАБИТами в версии ядра Enterprise. Если полный размер компьютерной RAM - меньше этого предела (832 MB или 3.6 ГИГАБИТА, соответственно), то фактическая величина области "низкой" памяти равна полному размеру памяти.

*Объединение RAM и swar пространства* - главный компьютерный ресурс, определяющий количество память, доступной приложениям. Если полный размер памяти, используемой приложениями превышает размер RAM, ядро Linux перемещает некоторые данные на swar, и загружает их заново в том случае, если они необходимы приложениям. Те данные, которые используются чаще, имеют тенденцию оставаться в RAM, менее часто используемые данные требуют больше времени для замены. Действия по включению и выключению функции swar в некоторой степени уменьшает производительность системы. Однако, если эта деятельность не чрезмерна, уменьшение работы не очень примечательно. С другой стороны, преимущества использования пространства swar также велики, поскольку позволяют увеличивать число VPS до двух раз. Пространство swar играет существенную роль при работе с системой загрузки. Система с достаточным местом обмена только замедляется при высоких показателях загрузки (load bursts), тогда как система без пространства swar реагирует на высокие показатели загрузки отказом распределения памяти (приложения отказываются узнавать клиентов или заканчивать свою работу) или просто разрушением каких-то приложений.

К тому же, присутствие пространства `swap` помогает системе лучше балансировать память и осуществлять перемещение данных между низкой областью памяти и остальной частью RAM.

*Allocated* (выделенная, ассигнованная) память - "более виртуальный" ресурс системы чем RAM или RAM и `swap` пространство вместе взятые.

Приложения могут ассигновать память, но начинают пользоваться этим лишь по истечению некоторого времени, и только в этом случае количество фактически свободной памяти действительно уменьшится. Сумма размеров памяти, ассигнованной всем серверам – всего лишь показатель того, сколько физической памяти потребуется, если все приложения ее потребуют.

Память, доступная для распределения может быть не только использована (**Alloc util** колонка) или обещана (the **Alloc commit** колонка), но также и ограничена

(приложения не в состоянии ассигновать больше ресурсов чем, обозначено в колонке **Alloc limit**).

## 5. Управление конфигурацией ресурсов VPS

Любой VPS конфигурируется посредством собственного файла конфигурации. Вы можете осуществлять управление конфигурацией ресурсов VPS разными способами.

1. используя образцы файлов конфигурации, прилагаемые к OpenVZ. Эти файлы используются при создании нового VPS (за деталями см.раздел СОЗДАНИЕ И КОНФИГУРАЦИЯ НОВЫХ VPS) они хранятся в той же директории, что и файлы конфигурации VPS (`/etc/sysconfig/vz-scripts/`) и имеют имя `ve-name.conf-sample`. Обычно, прилагаемые файлы образцы могут быть двух видов

- `light` используемые для создания “light” VPS с ограничениями верхнего предела качества сервисных параметров
- `vps.basic` используемые на обычных VPS

**Замечание:** что файлы образцы не могут содержать пробелов в своих названиях.

Любой образец файлов конфигурации, может прилагаться к VPS после его создания. Вы можете воспользоваться этим в том случае например, если вы хотите увеличить или снизить величину ресурсов используемых конкретным сервером.

```
# vzctl set 101 --applyconfig light --save
```

Эта команда переносит все параметры из файла `ve-light.conf-sample` на VPS. Исключение составляют `OSTEMPLATE`, `VE_ROOT`, and `VE_PRIVATE` параметры, они должны уже существовать в файле-образце

- используя специальные утилиты OpenVZ для подготовки файлов конфигурации к выполнению. Задания, которые выполняют эти утилиты, описаны в следующих разделах этой главы.
- прямо создавая и редактируя соответствующий файл конфигурации (`/etc/sysconfig/vz-scripts/VPS_ID.conf`). это можно выполнить с помощью любого текстового редактора. Соответствующие инструкции даются в четырех предыдущих разделах. В этом случае, вам придется редактировать все параметры конфигурации отдельно, один за другим.

### 1Разделение hardware node на равные части

Можно создать конфигурацию сервера, в общих чертах представляющую фракцию hardware node. Если вы хотите создать такой файл конфигурации, который позволял бы одновременную работу до 20 серверов от одного узла, Вы можете сделать это при помощи следующей командной сессии.

```
# cd /etc/sysconfig/vz-scripts/  
# vzsplrit -n 20 -f vps.mytest  
Config /etc/sysconfig/vz-scripts/ve-vps.mytest.conf-sample was created  
# vzcfgvalidate ve-vps.mytest.conf-sample  
Recommendation: kmemsize.lim-kmemsize.bar should be > 253952 \  
(currently, 126391)  
2Recommendation: dgramrcvbuf.bar should be > 132096 (currently, 93622)
```

Обратите внимание, что производимая конфигурация зависит от ресурсов hardware node. Однако, важно утвердить полученный файл конфигурации до его запуска. Это можно сделать, используя утилиту *vzcfgvalidate*.

Число одновременно функционирующих на hardware node серверов значительно выше того, что указано в командной строке, потому что в нормальном состоянии серверы не потребляют того количества ресурсов, которые им приписываются. Для того чтобы проиллюстрировать данную идею давайте поподробнее рассмотрим следующую секцию:

```
# vzctl create 101 --ostemplate fedora-core-4 --config vps.mytest
```

```
Creating VPS private area: /vz/private/101
```

```
VPS private area was created
```

```
# vzctl set 101 --ipadd 192.168.1.101 --save
```

```
or VPS 101
```

```
Saved parameters f# vzctl start 101
```

```
Starting VPS ...
```

```
VPS is mounted
```

```
Adding IP address(es): 192.168.1.101
```

```
VPS start in progress...
```

```
# vzcalc 101
```

```
Resource Current(%) Promised(%) Max(%)
```

```
Memory 0.53 1.90 6.44
```

Как следует из этих таблиц, если VPS использует все гарантируемые ему ресурсы, тогда одновременно могут работать около 20 серверов. Однако, принимая во внимание колонку Promised, достаточно безопасна и одновременная работа 40-50 серверов.

### Утверждение конфигурации VPS

Параметры контроля системных ресурсов имеют сложные взаимозависимости. Нарушение этих взаимозависимостей может быть катастрофическим для VPS. Чтобы гарантировать, что конкретный VPS не нарушает их, важно утвердить файл конфигурации VPS еще до создания VPS на основе этого файла.

Типичный сценарий утверждения показан ниже:

```
# vzcfgvalidate /etc/sysconfig/vz-scripts/101.conf
```

```
Error: kmemsize.bar should be > 1835008 (currently, 25000)
```

```
Recommendation: dgramrcvbuf.bar should be > 132096 (currently, 65536)
```

```
Recommendation: othersockbuf.bar should be > 132096 (currently,
```

```
122880) # vzctl set 101 --kmemsize 2211840:2359296 --save
```

```
Saved parameters for VPS 101 # vzcfgvalidate /etc/sysconfig/vz-scripts/101.conf
```

```
Recommendation: kmemsize.lim-kmemsize.bar should be > 163840 (currently, 147456)
```

```
Recommendation: dgramrcvbuf.bar should be > 132096 (currently, 65536)
```

```
Recommendation: othersockbuf.bar should ba > 132096 (currently, 122880)
```

```
Validation completed: success
```

Utility проверяет ограничения на параметры управления ресурсами и показывает весь список



найденных нарушений при заданных ограничениях. Могут быть три уровня степени серьезности нарушения:

***Recommendation (рекомендация)***

Это - предложение, которое не является критическим для VPS или Hardware Node. В целом конфигурация действительна; однако, если у системы достаточно памяти, лучше согласно рекомендациям увеличить параметры настройки.

***Warning (предупреждение)***

Ограничение не удовлетворено, и конфигурация недействительна. Приложения VPS с такой недействительной конфигурацией могут плохо выполнять свою работу или где-то ошибиться.

***Error (ошибка)***

Важное ограничение не удовлетворено, и конфигурация недействительна. Приложения VPS с такой недействительной конфигурацией имеют все шансы неожиданно потерпеть неудачу в работе, закрыться или зависнуть.

В сценарии, данном выше, первый запуск утилиты `vzcfgvalidate` обнаружил критическую ошибку для величины параметра `kmemsize`. После установки разумных величин для `kmemsize`, окончательная конфигурация вывела только уровень рекомендации, и VPS мог быть благополучно запущен с этой конфигурацией.

## Задачи первостепенного выполнения

В этой главе:

1. **Определение ID VPS**
  2. **Изменение системного времени VPS**
  3. **Устройства доступа на VPS**
  4. **Перемещение сетевого адаптера на VPS**
  5. **Обеспечение VPN для VPS**
  6. **Загрузка iptables модулей**
  7. **Перезагрузка VPS**
- 

### 1. **Определение ID VPS**

#### **Определение ID (идентификационного номера) VPS**

Каждый процесс имеет собственный PID (идентификатор процесса), который обеспечивает вход в ядерный каталог процессов.

Например, запуск Apache приписан соответствующим процессом ID.

PID используется затем, чтобы осуществлять контроль и управление этой программой.

PID - всегда положительное целое число.

В OpenVZ вы можете использовать `vzpid` (процесс восстановления ID), утилиту, которая печатает ID на сервере, которому принадлежит процесс с данным ID. Многократный процесс идентификации определяется как аргумент. В этом случае утилита напечатает номер/ число сервера /ов/ для каждого из процессов.

Утилита `vzpid` может иметь следующий вид:

```
[root@ts23 root]# vzpid
```

```
Pid VPS Name
```

```
12 4 init
```

В нашем примере процесс с идентификатором 12 назван 'init' и протекает на сервере с номером ID = 4.

### 2. **Изменение системного времени на VPS**

Обычно Изменение системного времени на VPS невозможно. Иначе, различные серверы могли бы вмешиваться в работу друг друга и разрушать те приложения, которые зависят от точности времени системы.

При нормальном состоянии вещей только администратор hardware node может изменить системное время. Однако,

если Вы хотите синхронизировать время посредством сетевого протокола времени (NTP), вы должны запустить программное обеспечение NTP, которое соединится с внешними серверами NTP и обновит время системы. Не желательно запускать прикладное программное обеспечение непосредственно на hardware node, так как недостатки в программном обеспечении могут привести к сбою всех частных серверов, находящихся на этом узле. Таким образом, если вы планируете использовать NTP, лучше создать отдельный VPS для того, чтобы иметь возможность работать с системным временем на sys\_time. Пример ниже иллюстрирует конфигурацию одного из таких серверов:

```
# vzctl set 101 --capability sys_time:on --save
Unable to set capability on running VPS
Saved parameters for VPS 101
```

Выполнение указанной команды предупреждает Вас, что vzctl не может применить изменения для работающего в данный момент сервера. VPS должен быть перезапущен для того, чтобы приписанные изменения вступили в силу

```
# vzctl restart 101
Restarting VPS
Stopping VPS ...
VPS was stopped
VPS is unmounted
Starting VPS ...
VPS is mounted
Adding IP address(es): 192.168.1.101
Hostname for VPS set: vps101.my.org
VPS start in progress...
# ssh root@vps101
root@vps101's password:
Last login: Mon Oct 28 23:25:58 2002 from 10.100.40.18
[root@vps101 root]# date
Mon Oct 28 23:31:57 EST 2002 [root@vps101 root]# date 10291300
Tue Oct 29 13:00:00 EST 2002
[root@vps101 root]# date
Tue Oct 29 13:00:02 EST 2002
[root@vps101 root]# logout
ion to ve101 closed.
Connect# date
Tue Oct 29 13:01:31 EST 2002
```

Эти команды показывают способ поменять системное время на VPS 101. Изменения затронут все VPS и hardware node. Не рекомендуется иметь более одного сервера со способностью регулировать системное время `sys_time`.

NTP описаны в интернет-стандарте RFC 1305; более подробная информация, включая программное обеспечение клиента может быть получена на сетевом сервере NTP (<http://www.ntp.org/>).

### ***3. Устройства доступа на VPS***

Вы можете предоставить серверу доступ к чтению, записи, или к чтению/записи блокирующих устройств. Возможно, что эти операции понадобятся, например, для программного обеспечения oracle database, если Вы хотите использовать его способность работать с сырым дисковым разделением.

В большинстве случаев доступ к иерархии файловых систем на сервере осуществляется при помощи связующих крепления. Однако, они не позволяют вам создавать новые разделения, форматировать их с файловыми системами, и устанавливать их на вашем сервере. Если Вы

Все же намерены уполномочить администратора сервера управлением диска, Вы можете использовать опции `- devices` или `- devnodes` для команды `vzctl`.

Пример ниже иллюстрирует следующую ситуацию: Вы хотите позволить корневому пользователю VPS 101 взять на себя ответственность за управление `/dev/sdb`, `/dev/sdb1` и `/dev/sdb2` устройствами. Иными словами, Вы позволяете системному администратору VPS 101 переразделять `/dev/sdb` устройство и создавать системы файлов на первых двух

разделениях (или использовать их с любым программным обеспечением, способным к работе с сырыми блокирующими устройствами, типа Oracle database software)

Сначала, мы собираемся «разрешить» серверу работу с необходимыми блокирующими устройствами:

```
# vzctl set 101 --devices b:8:16:rw --devices b:8:17:rw --devices
b:8:18:rw --save
Setting devperms
Saved parameters for VPS 101
```

Эта команда устанавливает разрешения для чтения/записи для блокирующих устройств с основным номером 8 и дополнительными номерами 16, 17 и 18 (соответствующими `/dev/sdb`, `/dev/sdb1`, и `/dev/sdb2`). Если Вы не уверены, какие главные и дополнительные числа соответствуют необходимым устройствам блочной конструкции, вы можете запустить эту команду:

```
# ls -l /dev/sdb{,1,2}
brw-rw---- 1 root disk 8, 16 Jan 30 13:24 /dev/sdb
brw-rw---- 1 root disk 8, 17 Jan 30 13:24 /dev/sdb1
brw-rw---- 1 root disk 8, 18 Jan 30 13:24 /dev/sdb2
```

Теперь давайте создадим разделение Linux на 100 МБ в дополнение к уже существующему разделению на 2 Гб на `/dev/sdb1` на VPS 101.

```
[root@vps101 root]# fdisk /dev/sdb
Command (m for help): p
Disk /dev/sdb: 255 heads, 63 sectors, 2231 cylinders
Units = cylinders of 16065 * 512 bytes
Device Boot Start End Blocks Id System
/dev/sdb1 * 1 255 2048256 83 Linux
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (256-2231, default 256):
Using default value 256
Last cylinder or +size or +sizeM or +sizeK \
(256-2231, default 2231): +100M
Command (m for help): p
Disk /dev/sdb: 255 heads, 63 sectors, 2231 cylinders
Units = cylinders of 16065 * 512 bytes
Device Boot Start End Blocks Id System
/dev/sdb1 * 1 255 2048256 83 Linux
/dev/sdb2 256 268 104422+ 83 Linux
Command (m for help): w
```

После того как новая разбивочная таблица написана, она может быть отформатирована и размещена на сервере.

```
[root@vps101 root]# mke2fs /dev/sdb2
[Output of mke2fs is skipped...]
[root@vps101 root]# mount /dev/sdb2 /mnt
[root@vps101 root]# df
Filesystem 1k-blocks Used Available Use% Mounted on
simfs 1048576 149916 898660 15% /
ext2 101107 13 95873 1% /mnt
```

Помните, что необходимо определить все технические детали и характеристики для устройств, которые вы хотели бы наделить наибольшими полномочиями; осуществляя доступ /dev/sdb, вы тем самым разрешаете создавать, изменять и удалять разделы

Но разрешение для разделений, с которыми вы разрешаете работать вашему серверу, должно быть выражено эксплицитно.

#### 4. Перемещение сетевого адаптера на VPS

По умолчанию, все VPS на hardware node связаны между собой и самим узлом с помощью действительного адаптера сети, который называется **venet0**. Однако, для VPS существует возможность непосредственного доступа к физическому адаптеру сети (например, **eth1**). В этом случае адаптер становится недоступен для самого hardware node .

Вы можете проделать это с помощью команды `vzctl`:

```
# vzctl set 101 --netdev_add eth1 --save
Add network device: eth1
Saved parameters for VPS 101
```

Пожалуйста примите во внимание, что сетевое устройство, добавленное на VPS таким способом будет иметь следующие ограничения:

Это устройство будет доступно только для VPS в независимости от его перемещений, но недоступно для hardware node и других VPS на этом узле.

Если такое устройство удалено с VPS ( с помощью команды `vzctl set --netdev_del` ) и добавлено на другой VPS , все сетевые настройки параметров этого устройства будут сброшены. Правильная работа в данном случае организуется следующим образом: необходимо сохранить все параметры настройки устройства в файле `ifcfg-dev` и иметь доступ к этому файлу в директории `/etc/sysconfig/network-scripts` у всех VPSs, которые могут использовать данное устройство (включая VPS0). После того, как устройство было добавлено на VPS, достаточно запустить команду `ifup dev` , чтобы считать параметры настройки упомянутого выше файла. Отметьте, что хотя, таким образом параметры настройки конфигурации сети и не будут восстановлены, таковы правила фильтрации пакетов.

Физическое устройство на VPS не имеет никаких ограничений безопасности, свойственных устройству `venet`. Внутри VPS возможно приписать любой IP адрес этого устройство и затем использовать его в самых разнообразных целях.

#### 5. Обеспечение VPN для VPS

Виртуальная Частная Сеть (VPN) - технология, которая позволяет Вам устанавливать безопасное сетевое соединение даже через небезопасную общественную сеть. Настройка VPN для тдельного VPS возможна посредством устройства TUN/TAP. Однако, требуется выполнить следующие шаги, чтобы разрешить конкретному VPS использовать это устройство.

Прежде чем запускать OpenVZ убедитесь, что модуль TUN загружен:

```
# lsmod | grep tun
```

В случае, если он не загружен, загрузите его следующей командой:

```
# modprobe tun
```

Чтобы позволить VPS использовать устройство TUN/TAP :

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

Создать соответствующее устройство в VPS для установки надлежащего разрешения:

```
# 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
```

Конфигурация надлежащей VPN выполняется как общая задача управления Linux, которую мы не рассматриваем подробно в нашем руководстве по управлению OpenVZ. Вы можете использовать любое из доступных популярных изданий по программному обеспечению Linux и настройке соответствующих устройств, а также драйверов к ним, например на (<http://vtun.sourceforge.net/>) или на (<http://openvpn.sourceforge.net/>).

## 6. Загрузка iptables модулей

Ядро OpenVZ обеспечивает поддержку дополнительным модулям iptables, которые не были загружены автоматически. Если Вы хотите, чтобы какие-то из этих модулей были загружен непосредственно на hardware node или дополнительно, на какой-то отдельный VPS , Вы должны



выполнить ряд операций

### **Загрузка iptables модулей на hardware node**

Для того, чтобы определенные модули iptables загружались при запуске hardware node , Вы должны включить их названия в состав параметра IPTABLES\_MODULES в файле /etc/sysconfig/iptables-config. По умолчанию этот параметр выглядит следующим образом:

```
IPTABLES_MODULES="ip_tables ipt_REJECT ipt_tos ipt_limit ipt_multiport  
iptables_filter iptable_mangle ipt_TCPMSS ipt_tcpmss  
ipt_ttl ipt_length"
```

Вы можете изменить его составляющие, включив туда любой из следующих модулей:

```
ip_conntrack  
ip_conntrack_ftp  
ip_conntrack_irc  
ipt_LOG  
ipt_conntrack  
ipt_helper  
ipt_state  
iptable_nat  
ip_nat_ftp  
ip_nat_irc  
ipt_TOS
```

Все модули, обозначенные как составляющие этого параметра будут загружены на запуске hardware node после того, как вы его перезагрузите.

Однако, если Вы хотите, чтобы этот набор модулей был загружен по умолчанию на VPS этого hardware node, или желаете ограничить загрузку любого из этих модулей на какой-то определенный VPS , Вы должны выполнить дополнительные операции.

### **Загрузка модулей iptables на определенный VPS**

То, какие из модулей iptables загружаются на VPS в данном hardware node по умолчанию, определяется составом параметра IPTABLES в файле /etc/sysconfig/vz.

Естественно, те модули, которые являются основными для данного параметра, будут загружены на VPSs только в том случае, если они также загружены непосредственно на hardware node (см. страницу 70).

Состав этого параметра может быть пересмотрен как в типовых файлах конфигурации VPS (/etc/sysconfig/vz-scripts/ve-sample\_name.conf-sample) так и в файлах конфигурации отдельных VPS (/etc/sysconfig/vz-scripts/vps\_id.conf).

Чтобы загружать дополнительные модули iptables или не загружать определенные модули, установленные по умолчанию на этом VPS, вам необходимо четко указать, какие **именно** модули вам нужно загрузить на VPS. Вы можете достичь этого 2 способами : изменить параметр IPTABLES в соответствующих файлах конфигурации VPS или использовать команду vzctl. Например:

```
# vzctl set 101 --iptables iptable_filter --iptables ipt_length --iptables ipt_limit --iptables  
iptables_mangle --iptables ipt_REJECT --save
```

Эта команда приказывает OpenVZ загружать только следующие модули на VPS 101: iptable\_filter, ipt\_length, ipt\_limit, iptable\_mangle, ipt\_REJECT.

Эта информация будет также сохранена в файле конфигурации VPS благодаря --save функции.

Загрузка нового набора модулей iptables не происходит мгновенно. Вы должны перезагрузить VPS , чтобы обозначенные изменения вступили в силу.

## 7. *Перезагрузка VPS*

Когда вы запускаете команду reboot на панели управления Linux, команда перезагружает систему, используя команду restart, которая передается к компьютерному BIOS.

Затем ядро Linux перезагружает компьютер. По очевидным причинам этот запрос системы блокируется внутри VPS ни один сервер не может получить непосредственный доступ к BIOS; в противном случае, перезагрузка внутри VPS привела бы к перезагрузке всего hardware node. Именно поэтому команда reboot внутри VPS фактически работает обратным способом. При выполнении команды reboot внутри VPS – работа VPS останавливается, а затем начата специальным скриптом (/etc/sysconfig/vz-scripts/vpsreboot), который выполняется посредством cron даемон периодически (каждую минуту по умолчанию). Конфигурация Cron для управления этим скриптом находится в файле /etc/cron.d/vpsreboot.

Если Вы хотите, чтобы VPS был неспособен самостоятельно вызвать перезагрузку, добавьте строку ALLOWREBOOT = " no" в файл конфигурации вашего сервера

(/etc/sysconfig/vz-scripts/vps\_id.conf).

Если Вы хотите, чтобы отказ от перезагрузки VPS был установлен по умолчанию, или хотите точно определить какие из VPS могут перезагружаться - добавьте строку ALLOWREBOOT = "no " в основной файл конфигурации OpenVZ (/etc/sysconfig/vz) и точно введите ALLOWREBOOT = "yes" в соответствующих файлах конфигурации сервера.

## ***Поиск ядерных неисправностей***

В этой главе вы найдете информацию о тех проблемах, которые могут произойти в течение вашей работы с OpenVZ и предлагаемые способы их решения.

*В этой главе:*

- 1. Общие рассмотрения***
  - 2. Выявление ядерных неисправностей***
  - 3. Проблемы с управлением VPS***
  - 4. Проблемы с функционированием VPS***
  - 5. Проблемы с использованием Linux утилит (?)***
  - 6. Получение технической поддержки (?)***
- 

### ***1. Общие рассмотрения***

Основные проблемы, которые необходимо принять во внимание, исследуя Вашу систему OpenVZ на предмет поиска неполадок, перечислены ниже:

Вы должны всегда помнить, в каком терминале сейчас находитесь. Периодически проверяйте это, используя команды `pwd`, `hostname`, `ifconfig`, `cat/proc/vz/veinfo`. Одна и та же команда, выполненная в VPS и в HN может привести к абсолютно разным результатам! Вы можете также настроить переменную среду `PS1`, чтобы показать полную директорию к `bash` prompt. Чтобы сделать это, добавьте эти линии к `/root/.bash_profile`:

```
PS1="[u@h \w]$ "  
export PS1
```

- Если работа `hardware node` по каким-то причинам замедлила свое действие, воспользуйтесь `vmstat`, `ps` (`ps axfw`), `dmesg`, `top`, чтобы узнать, что случилось. Никогда не перезагружайте машину без выяснения обстоятельств неполадок. Если исследование не помогает, восстановите нормальное функционирование, используя последовательность клавиш `Alt+SysRq`, чтобы загрузить память (`showMem`) и процессы (`showPc`). За дополнительной информацией обращайтесь к секции `Использование Последовательности Клавиш Alt+SysRq` (`Using Alt+SysRq keyboard section`).

- Если `hardware node` был некорректно закрыт, при его следующем запуске все его разделы

должны быть проверены заново и разделы дисков повторно вычислены для каждого VPS. Все это, несомненно, значительно увеличит время запуска.

- Не запускайте бинары или скрипты, которые принадлежат VPS, непосредственно с hardware node.

Например, никогда не делайте этого:

```
cd /vz/root/99/etc/init.d  
./httpd status
```

Любой скрипт в VPS, может быть изменен в зависимости от того, что выбирает владелец VPS: он мог быть поражен трояном, заменен на что-то типа gm-gf, и т.д. поэтому, для того, чтобы извлечь программы на VPS Вы можете использовать только `vzctl exec` или `vzctl enter`.

- Не используйте `init` скрипты на hardware node. `init` скрипт может использовать `killall`, чтобы остановить обслуживание. Фактически, это означает, что процессы будут остановлены на всем VPS! Вы можете проверить `/var/run/service.pid`, чтобы не допустить этот процесс.

- Вы должны быть способны обнаружить любой rootkit в VPS. Для этого рекомендуют использовать пакет `chkrootkit` (Вы можете загрузить последнюю версию на [www.chkrootkit.org](http://www.chkrootkit.org)), или, по крайней мере, запустить

```
rpm -Va|grep "S.5"
```

Для осуществления проверки суммы MD5 или какого-нибудь файла типа `rpm` Вы можете также запустить `ntar`, например:

```
# nmap -p 1-65535 192.168.0.1
Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.0.1):
(The 65531 ports scanned but not shown below are in
state: closed)
Port State Service
21/tcp open ftp
22/tcp open ssh
80/tcp open http
111/tcp open sunrpc
Nmap run completed -- 1 IP address (1 host up) scanned
in 169 seconds
```

чтобы проверить, не открыты ли какие-нибудь порты, которые обычно должны быть закрыты.

Вы можете также столкнуться с проблемой, как удалить rootkit из VPS и удостовериться, что он удален на - 100 % . Если Вы не уверены, создаете новый VPS для покупателя и переместите его данные туда.

- Проверьте директорию /var/log/ на hardware node, чтобы узнать, что происходит в системе. Есть множество файлов, которые поддерживаются системой и OpenVZ (the boot.log, messages, vzctl.log log files и т.д.), Однако, в зависимости от вашего распределения файлов на Linux там могут быть размещены и другие программы, приложения и службы, которыми вы пользуетесь, например, связанные с управлением почтовым сервером (maillog файл), автоматические задачи (cron файл), и другие.

Первое место, которое необходимо изучить во время проверки на наличие неполадок, - файл /var/log/messages log .

Именно он содержит сообщения о загрузке и функционировании системы. Ошибки, связанные с входом/выходом, организацией сети, а также другие ошибки, которые можно назвать основными для системы, транспортируются в этот файл. Поэтому, мы рекомендуем, чтобы вы вначале обратили внимание этот, а затем уже все остальные файлы из директории /var/log/

- Подпишитесь, на получение списков файлов, по крайней мере, для обнаружения «Red hat», а также других файлов, опасных для сервера и системы.

- Пользуясь iptables, соблюдайте простое правило для последовательной защиты как Hardware Node так и VPSs:

используйте INPUT, OUTPUT, , чтобы фильтровать пакеты, которые прибывают в HN;

используйте FORWARD, чтобы фильтровать пакеты, которые предназначены для VPSs.

## 2. Выявление ядерных неисправностей

### Использование последовательности клавиш ALT+SYSRQ (Keyboard Sequences )

Нажмите ALT+SYSRQ+H (3 клавиши одновременно) и проверьте, что при этом появляется на панели HN, например:

```
SysRq : HELP : loglevel0-8 reBoot tErm Full kIll saK showMem Nice powerOff showPc  
unRaw Sync showTasks Unmount
```

Эта команда показывает, что ALT+SYSRQ вы можете использовать для того, чтобы выполнить любую команду. Заглавные буквы в названии команды служат для идентификации очередности набора клавиш. Таким образом, если возникали какие-то неполадки с машиной, и вам нужно ее перезагрузить, то сначала выполните следующие действия -

пожалуйста, нажмите в следующей последовательности перед нажимом кнопки Power:

ALT+SYSRQ+M, чтобы загрузить информацию памяти;

ALT+SYSRQ+P, чтобы загрузить управление процессами;

ALT+SYSRQ+S, чтобы синхронизировать диски;

ALT+SYSRQ+U, чтобы перераспределить всех установленных файловых системы;

ALT+SYSRQ+E, чтобы завершить процессы;

ALT+SYSRQ+I, чтобы остановить все процессы

ALT+SYSRQ+U, чтобы попробовать перераспределить их еще раз;

ALT+SYSRQ+B для перезагрузки.

Если после этого компьютер не перезагружается, вы можете нажать кнопку Power.

## Сохранение ядерных ошибок (OOPS)

Вы можете использовать следующую команду для того, чтобы узнать, есть ли ядерные сообщения, которые нужно расшифровать и сообщить разработчикам OpenVZ:

```
grep -E "Call Trace|Code" /var/log/messages*
```

Теперь нам необходимо найти эти же строки в соответствующем log файле, чтобы вычислить, какое ядро было загружено, когда произошло сохранение OOPS. Ищите записи, подобные этой :

```
May 23 16:55:00 ts13 Linux version 2.6.8-022stab026.1 (root@kern26x.build.sw.ru) (gcc version 3.3.3 20040412 (Red Hat Linux 3.3.3-7)) #1 Fri Jul 8 17:31:10 MSD 2005
```

OOPS обычно начинается с описания того, что случилось, и заканчивается строкой кодов, подобно этой:



```
May 24 15:12:07 ts13 Unable to handle kernel paging request at virtual address d0d48b08
May 24 15:12:07 ts13 printing eip:
May 24 15:12:07 ts13 c01b4049
May 24 15:12:07 ts13 *pde = 00044063
May 24 15:12:07 ts13 *pte = 10d48000
May 24 15:12:07 ts13 Oops: 0000 [#1]
May 24 15:12:07 ts13 SMP DEBUG_PAGEALLOC
May 24 15:12:07 ts13 Modules linked in: e100 mii af_packet ip_nat_ftp ip_nat_irc
ipt_helper ip_conntrack_irc ip_conntrack_ftp ipt_TOS ipt_LOG ipt_conntrack ipt_state
iptable_nat ip_conntrack ipt_length ipt_ttl ipt_tcpmss ipt_TCPMSS iptable_mangle iptable_filter
ipt_multiport ipt_limit ipt_tos ipt_REJECT ip_tables
May 24 15:12:07 ts13 CPU: 0, VCPU: 2147483647:0
May 24 15:12:07 ts13 EIP: 0060:[<c01b4049>] Not tainted
May 24 15:12:07 ts13 EFLAGS: 00010206
May 24 15:12:07 ts13 EIP is at proc_pid_stat+0x289/0x5b0
May 24 15:12:07 ts13 eax: d0d48a70 ebx: 00000000 ecx: 00000000 edx: c0128962
May 24 15:12:07 ts13 esi: 00000000 edi: c599fa70 ebp: d93f2f34 esp: d93f2e04
May 24 15:12:07 ts13 ds: 007b es: 007b ss: 0068
May 24 15:12:07 ts13 Process top (pid: 19753, threadinfo=d93f2000 task=d93f1a70)
May 24 15:12:07 ts13 Stack: c599fa70 00000000 c041b980 d93f2e50 c018e7bd cc752f58
c041b980 d8b4df58
May 24 15:12:07 ts13 d93f2e4c c01b0cea c599fa70 c627a00c 00000004 df1c3f58
00000000 d8b4df58
May 24 15:12:07 ts13 cc752f58 00000000 c0d97e94 d93f2e6c c018eb9e 04837000
00000000 d93f2e84
May 24 15:12:07 ts13 Call Trace:
May 24 15:12:07 ts13 [<c010650f>] show_stack+0x7f/0xa0
May 24 15:12:07 ts13 [<c01066df>] show_registers+0x17f/0x220
May 24 15:12:07 ts13 [<c01068c7>] die+0xa7/0x170
May 24 15:12:07 ts13 [<c01188ea>] do_page_fault+0x2fa/0x59e
May 24 15:12:07 ts13 [<c034576f>] error_code+0x2f/0x38
May 24 15:12:07 ts13 [<c01afe71>] proc_info_read+0x51/0x160
May 24 15:12:07 ts13 [<c0171dea>] vfs_read+0xaa/0x130
May 24 15:12:07 ts13 [<c017208b>] sys_read+0x4b/0x80
May 24 15:12:07 ts13 [<c0344cda>] sysenter_past_esp+0x43/0x61
May 24 15:12:07 ts13 Code: 8b 80 98 00 00 00 89 45 b0 f0 ff 05 c0 b7 41 c0 8b 87 88 01
```

Все, что вам нужно – это переместить OOPS в соответствующий файл.

### **Обнаружение ядерной функции, вызвавшей процесс D**

Опишем следующую ситуацию - слишком много процессов находятся в состоянии D, Вы не можете разобраться, что именно происходит. В данных обстоятельствах выполните следующую команду:

```
# objdump -Dr /boot/vmlinux-`uname -r` >/tmp/kernel.dump
```

После которой вы сможете просмотреть список процессов:

```
# ps axfwln
F UID PID PPID PRI NI VSZ RSS WCHAN STAT TTY TIME COMMAND
100 0 20418 20417 17 0 2588 684 - R ? 0:00 ps axfwln
100 0 1 0 8 0 1388 524 145186 S ? 0:00 init
040 0 8670 1 9 0 1448 960 145186 S ? 0:00 syslogd -m 0
040 0 8713 1 10 0 1616 1140 11ea02 S ? 0:00 crond
```

Для искомого процесса - ищите число в колонке WCHAN. Затем необходимо открыть в редакторе /tmp/kernel.dump, найти число в первой колонке и затем вернуться к первичному названию функции, которое может выглядеть примерно так:

```
"c011e910 <sys_nanosleep>:"
```

Теперь вы можете определить, функционирует ли процесс или заблокирован в найденную функцию.

## **3. Проблемы с управлением VPS**

Эта секция включает рекомендации о том, как уладить некоторые проблемы с вашим VPS.

### **Отказ создать VPS**

*Описание ситуации:*

Попытка создавать новый сервер не удалась. На системной панели появилось сообщение  
Cached os template /vz/template/cache/XXX.tar.gz not found.

*Решение :*

На hardware node может отсутствовать необходимый шаблон. Скопируйте шаблон на hardware node, установите его, скройте, и попробуйте создать VPS еще раз.

### **Отказ запускать VPS**

*Описание ситуации:*

Попытка запустить сервер не удалась.

#### *Решение 1*

Если на системной панели появилось сообщение parameters missing, что означает, что некоторые из системных параметров отсутствуют, и далее следует список этих параметров - установите их, используя vzctl set --save команду (см. для инструкций секцию Конфигурация VPS), а затем попробуйте запустить VPS еще раз.

#### *Решение 2*

Если на системной панели появилось сообщение: Address already in use, , выполните следующую команду /proc/vz/veinfo. В результате - вы должны получить данные о числовом идентификаторе VPS, классе VPS, числе процессов VPS и IP адресах VPS для каждого из работающих VPS. Уже само наличие информации означает, что каждый из ваших VPS в норме, т. е. может работать без приписываемых IP адресов. Для настройки IP адреса VPS используйте команду:

```
vzctl set vps_id --ipadd addr --save
```

где vps\_id представляет собой числовой идентификатор VPS, и addr является его фактическим IP адресом.

#### *Решение 3*

Плохо настроенные параметры UBC также могут препятствовать запуску VPS. Попробуйте утвердить конфигурацию VPS (см. раздел Утверждение Конфигурации VPS ). Посмотрите, какие из параметров могли вызвать ошибку, и измените их содержание и величины, используя команду vzctl set --save. .

#### *Решение 4*

VPS, возможно, использовал все разделы диска ( место на диске/ диск inodes). Проверьте соответствующий дисковый раздел на VPS (за подробной информации обращайтесь к главе 6 Управление Дисковыми Долями) и увеличьте, если необходимо, параметры этого раздела (за подробной информации обращайтесь к главе 6 Настройка параметров Разделов Дисков).

### **Отказ в доступе к VPS**

#### *Решение 1*

IP адрес, приписываемый VPS может быть уже использован в вашей сети. Удостоверьтесь, что это не так. Необходимо осуществлять систематические проверки, не возникает ли проблем с адресами VPS, запустив следующую команду

```
# grep IP_ADDRESS /etc/sysconfig/vz-scripts/VPS_ID.conf  
IP_ADDRESS="10.0.186.101"
```

IP адреса других работающих VPSs, могут быть проверены

```
cat /proc/vz/veinfo
```

#### *Решение 2*

Удостоверьтесь, что маршрут к VPS указан правильно. Сервер может использовать для вашей сети маршрут по умолчанию. Если вас это не устраивает, вы можете сами настроить соответствующий маршрут к VPS на hardware node.

### **Отказ войти на VPS**

#### *Описание ситуации:*

VPS загружается, но вы не можете в него войти.

#### *Решение 1*

Вы соединяетесь через SSH, но доступ отрицается. Вероятно, Вы не установили пароль Root-user или просто такого пользователя еще не существует.. В данных обстоятельствах ва лучше всего воспользоваться командой `vzctl set --save --userpasswd`.

. Например, для VPS 101 Вы могли бы воспользоваться командой такого типа:

```
# vzctl set 101 --save --userpasswd root:secret
```

#### *Решение 2*

Проверьте существующие настройки при помощи команды:

```
# cat /proc/sys/ipv4/conf/venet0/forwarding
```

Если в командной строке вы обнаружите 0- измените его на 1 при помощи команды:

```
# echo 1 > /proc/sys/ipv4/conf/venet0/forwarding
```

## **4. Проблемы с функционированием VPS**

### **Перерыв при доступе к удаленным hosts**

#### *Описание ситуации:*

Вход на host не доступен как с hardware node системы Open VZ так и с его VPS, хотя возможно осуществить вход на него с других компьютеров.

#### *Решение*

Часто эти перерывы вызваны тем, что механизм ECN ( Явное Уведомление Скопления ) протокола TCP/IP включен по умолчанию в системе Open VZ и отключен в других системах. Такая ситуация может привести к несовместимости нескольких систем, и естественно, вызвать неполадки в их совместной работе. ECN используется, чтобы избежать ненужных снижений пакетов, для того, чтобы ускорить и улучшить их продукцию. Если система Open VZ не может установить соединение с host, отключите ECN механизм при помощи команды:

```
# sysctl -w net.ipv4.tcp_ecn=0  
net.ipv4.tcp_ecn = 0
```

## Глава 9

### Ссылки

Для того чтобы система Open VZ эффективно функционировала, необходимо научиться ею правильно управлять. В этой главе описываются параметры Open VZ и их влияние на деятельность системы.

В этой главе:

1. **Конфигурирование Open VZ**
  2. **Интерфейс командных строк Open VZ (OpenVZ Command Line Interface)**
- 

### 1. Конфигурирование Open VZ

Для того чтобы система Open VZ правильно функционировала, необходимо научиться её правильно настраивать. Из этой главы Вы узнаете о параметрах системы и их влиянии на работу Open VZ.

#### Матрица файлов конфигурации Open VZ

Существует множество файлов, отвечающих за системную конфигурацию Open VZ. Эти файлы расположены в директории */etc* на Hardware Node; перечень этих файлов прилагается:

<code>/etc/sysconfig/vz</code>	Глобальный файл конфигурации. Хранит параметры настройки всей системы, влияя на местоположение шаблонов VPS и Open VZ, глобальные параметры настройки сети и т.д.
<code>/etc/sysconfig/vz-scripts/vpsid.conf</code>	Частный файл конфигурации, принадлежащий численному файлу VPS <i>vpsid</i> . В файле содержатся определенные параметры настройки VPS - параметры управления ресурса, местоположение частной области, IP адрес и так далее.

/etc/sysconfig/vz-scripts/ve-name.conf-sample	Типовые файлы, содержат множество конфигурации VPS по умолчанию, которые могут использоваться в качестве ссылок при создании VPS. Следующие образцы доставляются вместе с Open VZ: <i>light</i> , <i>vps.basic</i> .. Кроме того, Вы можете создать Ваши собственные образцы, настроенные для Ваших потребностей.
/etc/sysctl.conf	Параметры ядра. Вы должны отрегулировать некоторые параметры ядра <i>sysctl</i> , хранящиеся в файле <i>/etc/sysctl.conf</i> .
/etc/cron.d/vpsreboot	Файл конфигурации для <i>cron</i> даемон. Используя этот файл, Open VZ подражает команде перезагрузка при работе в VPS.
/etc/sysconfig/vz-scripts/dists/dist_name.conf	Обычно файлы конфигурации определяют, какие скрипты должны выполнять функции в сфере VPS (добавление нового IP адреса VPS и т.д.). Эти же скрипты отличаются от скриптов деятельности Open VZ и зависят от версии Linux, которую использует VPS.

### Глобальный файл конфигурации

Система Open VZ хранит общие параметры конфигурации системы в файле конфигурации */etc/sysconfig/vz*. Этот файл находится в формате shell. Имейте в виду, что скрипты Open VZ используют этот файл - таким образом, что shell команды в этом файле заставляют систему выполнять задания под корневым аккаунтом. В этом файле параметры представлены в форме *PARAMETER="value"*. Все параметры относятся к следующим группам: глобальные параметры, logging параметры, параметры дисковой доли, шаблона, серверов VPS, и дополнительные параметры. Ниже приводится описание всех параметров, определенных в этой версии Open VZ.

<i>Параметр</i>	<i>Описание</i>	<i>Ценность по умолчанию (Default value)</i>
-----------------	-----------------	--

<i>VIRTUOZZO</i>	Может быть или «да» или «нет». Загрузочный скрипт системы Open VZ V проверяет этот параметр. Если стоит «нет», тогда модули Open VZ не загружаются. Вы можете настроить «нет», если хотите выполнить обслуживание системы, и не хотите держать все серверы VPS на Hardware Node.	Да
<i>LOCKDIR</i>	Все действия на сервере VPS последовательны, так как две одновременные операции на одном и том же сервере могут разрушить его устойчивость. Open VZ содержит в этой директории также файлы открытия/закрытия, чтобы сделать доступ к VPS более последовательным.	<i>/vz/lock</i>
<i>VEOCPUUNITS</i>	Вес CPU , определяемый для Hardware Node.	<i>1000</i>

Параметры загрузки (*Logging parameters*) влияют на утилиту загрузки *vzctl*.

<b>Параметр</b>	<b>Описание</b>	<b>Ценность по умолчанию</b>
LOGGING	Этот параметр определяет должен ли <i>vzctl</i> регистрировать свои действия.	Да
LOGFILE	Файл, где <i>vzctl</i> регистрирует свои действия.	<i>/var/log/vzctl.log</i>
LOG_LEVEL	Существуют 3 стадии регистрации, определенные версией 0 Open VZ.	0

Таблица ниже показывает возможные ценности параметра *LOG\_LEVEL* и их значения:



<b>Уровень регистрации</b>	<b>Подлежащая регистрации информация</b>
<b>0</b>	Действия <i>vzctl</i> на серверах – запуск, остановка, создание, разрушение, установка, демонтаж.
<b>1</b>	Этот уровень регистрирует события, обращается к вспомогательным <i>vzctl</i> скриптам, находящимся в <i>/etc/sysconfig/vz-scripts</i> (таким как <i>vz-запуск</i> и <i>vz-отключение</i> ), а также регистрирует ситуации, когда основной процесс VPS прекращается при закрытии VPS после небольшой паузы.
<b>2</b>	Содержит регистрации 0 и 1 уровней, плюс версию шаблона, используемую для создания VPS, и обращается к <i>mount</i> и <i>quota</i> операциям с параметрами.

*Параметры дисковой доли (Disk quota parameters)* позволяют Вам контролировать использование диска несколькими VPS:

<b>Параметр</b>	<b>Описание</b>	<b>Ценность по умолчанию</b>
DISK_QUOTA	DISK_QUOTA определяет включать ли дисковые доли для серверов VPS. Если «нет», тогда место на диске и <i>inodes accounting</i> (подсчет) будут заблокированы.	Да (Yes)
VZFASTBOOT	Если установить «нет», то дисковая доля будет реинициализирована для каждого конкретного VPS, если после неправильного выключения не будет запущен Hardware Node. Все это повлечет за собой увеличение времени загрузки. Если же установить «да», все серверы VPS будут запущены без реинициализации доли, но затем последовательно приостановлены. То есть один за другим серверы VPS начнут реинициализацию их долей.	Нет (No)

*Параметры шаблона (Template parameters)* позволяют формировать местоположение области шаблона.

<b>Параметр</b>	<b>Описание</b>	<b>Ценность по умолчанию</b>
-----------------	-----------------	------------------------------

TEMPLATE	В этой директории можно найти шаблоны. Не рекомендуется менять опцию, так как все шаблоны SWsoft используют директорию по умолчанию (default).	/vz/template
----------	---	--------------

Параметры по умолчанию сервера VPS (*Virtual Private Server default parameters*) либо влияют на создание нового VPS, или же представляют не принятые во внимание параметры (*overridden*) VPS в файле конфигурации VPS:

<b>Параметр</b>	<b>Описание</b>	<b>Ценность по умолчанию</b>
<i>VE ROOT</i>	Это путь к корневой директории VPS, где установлена частная область.	<i>/vz/root/\$VEID</i>
<i>VE PRIVATE</i>	Путь к частной области VPS. Реализация Open VZ требует, чтобы <i>VE_PRIVATE</i> находился в пределах одного физического деления.	<i>/vz/private/\$VEID</i>
<i>CONFIGFILE</i>	Образец файла конфигурации по умолчанию для создания VPS, может быть отклонен с помощью опции <code>--config</code> команды <i>vzctl create</i> .	<i>vps.basic</i>
<i>DEF OSTEMPLATE</i>	Шаблон ОС по умолчанию для создания VPS, может быть отклонен с помощью командной построчной опции (command line option) <code>--ostemplate</code> для <i>vzctl create</i> .	<i>fedora-core-4</i>
<i>DEF OSTEMPLATE</i>	На VPS, находящимся на Hardware Node, будут загружены только те модули <i>iptables</i> , которые обозначены как ценность данного параметра и только те, которые находятся непосредственно на Node. Список модулей пригодных для загрузки: p_conntrack, ip_conntrack_ftp, ip_conntrack_irc, iptable_fliter, ipt_length, ipt_limit, ipt_LOG, iptable_mangle, ipt_conntrack, ipt_helper, ipt_state, ipt_tcpmss, ipt_tos, pt_multiport, iptable_nat, ip_nat_ftp, ip_nat_irc, ipt_REJECT, ipt TCPMSS, ipt TOS, ipt ttl.	<i>ipt_REJECT ipt_tos ipt_limit ipt_multiport iptable_fliter iptable_mangle ipt_TCPMSS ipt_tcpmss ipt_ttl ipt_length</i>

Дополнительные параметры определяют другие параметры настройки OpenVZ:

<b>Параметр</b>	<b>Описание</b>	<b>Ценность по умолчанию</b>
VZWDOG	Определяет, загружен ли на startup Open VZ модуль vzwdog. Этот модуль ответственен за получение ядерных сообщений. Он необходим, если Вы формируете serial Monitor Node для Open VZ.	Нет

### Файл конфигурации VPS

Каждый сервер имеет свой файл конфигурации, хранящийся в директории */etc/sysconfig/vz-scripts* под названием *vpsid.conf*. Этот файл имеет тот же самый формат, что и глобальный файл конфигурации. Параметры настройки, определенные в этом файле, могут быть подразделены на следующие категории: разнообразные параметры, параметры работы сети, и параметры управления ресурсом.

#### Разнообразные параметры

VERSION	Определяет версию Open VZ, к которой прибегает файл конфигурации.
ONBOOT	Определяет, должен ли VPS загружаться автоматически вместе с системой. Open VZ загружает автоматически все серверы, где стоит значок «да».
ALLOWREBOOT	Определяет, может ли VPS быть перезагружен с помощью внутренней команды "перезагрузки" (reboot). Если не указано или стоит значок "да", перезагрузка разрешается.
CAPABILITY	Определяет внутренние возможности \ способности VPS. Разрешается настройка следующих способностей: CHOWN, AC_OVERRIDE, AC_READ_SEARCH, FOWNER, FSETID, KILL, SETGID, SETUID, SETPCAP, LINUX_IMMUTABLE, NET_BIND_SERVICE, NET_BROADCAST, NET_ADMIN, NET_RAW, IPC_LOCK, IPC_OWNER, SYS_MODULE, SYS_RAWIO, SYS_CHROOT, SYS_PTRACE, SYS_PACCT, SYS_ADMIN, SYS_BOOT, SYS_NICE, SYS_RESOURCE, SYS_TIME, SYS_TTY_CONFIG, MKNOD, LEASE.

OSTEMPLATE	Название шаблона ОС, который использовался для создания VPS. Не нужно менять его параметры. Команда <code>vzctl</code> установит все за Вас, используя команду <code>vzctl create vpsid --ostemplate template</code> (или используя <code>defaults</code> глобального файла конфигурации).
VE_ROOT	Убирает (overrides) параметр <code>VE_ROOT</code> из глобального файла конфигурации.
VE_PRIVATE	Убирает параметр <code>VE_PRIVATE</code> из глобального файла конфигурации.

*Параметры управления ресурса* контролирует количество потребляемых VPS ресурсов. Они детально описываются в главе *Управление ресурсами*. Здесь дается лишь список параметров, разрешенных в файле конфигурации VPS.

Для удобства все параметры управления ресурса могут быть подразделены на общие, дисковые, и системные. Любой параметр может быть установлен с помощью команды `vzctl set` и соответствующего option name (названия выбора) (строчными буквами это будет так - `kmemsize` для `KMEMSIZE`, и т.д.). За более детальной информацией обращайтесь к разделу *OpenVZ Command Line Interface*. Колонка Обычная ценность (Typical value) определяет диапазон ценностей параметра для разных приложений, от легко - до сильнозагруженных VPS (потребляющих 1/8 часть Hardware Node с памятью 2 ГБ). Если используются пограничные и предельные области, даются диапазоны для 2 порогов.

<i>Параметр</i>	<i>Описание</i>	<i>Обычная ценность (Typical value)</i>
<i>ORIGIN SAMPLE</i>	Образец конфигурации, на основе которого был создан VPS.	
<i>CONFIG CUSTOMIZED</i>	Касательно оригинального шаблона конфигурации, указывает были ли какие-нибудь из параметров конфигурации VPS изменены. Если параметр пропущен, ценность приравнивается к значку «нет»	

<i>CPUUNITS</i>	Гарантируемая мощность CPU. Это - положительное целое число, которое определяет гарантируемую минимальную долю CPU, которую получает VPS. Полная мощность CPU в CPUUNITS – число Vopomips, умноженное на 25. Отчетные инструменты Open VZ (reporting tools) приравнивают 1 GHz PNI Intel processor к 50 000 единицам CPU.	250...1000
<i>CPULIMIT</i>	Разрешенная мощность CPU. Это - положительное число, обозначающее долю времени CPU в %, которую VPS никогда не превысит. Вы можете рассчитать эту долю так - (разрешенная мощность VPS CPUUNITS/CPU power)* 100%.	1...4
<i>DISKSPACE</i>	Полный размер дискового пространства, потребляемого VPS, блоками по 1 Кб	204800...10485760 - 204800...11534340
<i>DISKINODES</i>	Общее количество inodes диска (файлы, директории, символические связи), которые может разместить VPS.	80000...400000- 88000...440000
<i>QUOTATIME</i>	Grace период дисковой доли. Он рассчитан в секундах. Серверу VPS разрешается временно превысить quota soft limits, но не больше чем на период QUOTATIME.	0...604800
<i>QUOTAUGIDLIMIT</i>	Определенное количество user/group IDs для внутренней дисковой доли. Если стоит 0, UID/GID quota не будет задействована.	0...500
<b><i>Системные параметры</i></b>		
<i>NUMPROC</i>	Разрешено определенное количество процессов и тредов. До достижения этого предела, у VPS не будет возможности начать новый процесс или тред.	40...400

AVNUMPROC	Среднее число предполагаемых процессов, идущих на сервере VPS. AVNUMPROC - информационный параметр, используемый такими утилитами как <i>vzcfgvalidate</i> для того, чтобы гарантировать правильность конфигурации.	0...NUMPROC
NUMTCP SOCK	Число гнезд TCP (PF_INET family, 40...500 SOCK_STREAM type). Этот параметр ограничивает число связей TCP и, таким образом, число пользователей, которых параллельно может обслужить приложение сервера.	40...500
NUMOTHERSOCK	Число гнезд отличных от TCP. Местные (UNIX- domain) гнезда используются для коммуникаций внутри системы. Например, гнезда UDP используются для запросов Domain Name Service (DNS). UDP и другие гнезда могут также использоваться в некоторых специфических приложениях (SNMP agents and others)	40...500
VMGUARPAGES	Гарантия распределения памяти, постранично. Приложения гарантированно смогут уместить память, так как объем памяти под аккаунтом <i>privvmpages</i> не превышает установленную границу параметра <i>vmguarpages</i> . Размещение объема памяти выше границы не гарантируется и может быть неудачным.	1725...107520
KMEMSIZE	Объем unswappable (неменяемой) ядерной памяти, ассигнованной для внутренних ядерных структур для процессов конкретного VPS. Обычный объем ядерной памяти - 16 ... 50 КБ на процесс.	798720...13148160
TCP SNDBUF	Полный размер send буферов для гнезд TCP – объем ядерной памяти, предназначенной для данных, посланных от приложения в гнездо TCP, но еще не признанной удаленной стороной.	159744...5365760 - 262144...10458760

TCPRCVBUF	Полный размер receive буферов для TCP гнезд. Объем корневой памяти, полученной от отдаленной стороны, но не используемой местным приложением.	159744...5365760- 262144...10458760
OTHERSOCKBUF	Общий размер буферов гнезда UNIX-domain, размер UDP и datagram protocols send буферов.	61440...1503232- 163840...4063232
DGRAMRCVBUF	Общий размер receive буферов UDP и других протоколов datagram	32768...262144
OOMGUARPAGES	Гарантия out-of-memory, постранично. Процессы VPS не будут прекращены даже в случае острой нехватки памяти, если текущее потребление (включая физическую память и swap) не достигло барьера <i>oomguarpages</i> .	1725...107520
LOCKEDPAGES	Память, которую не разрешается обменивать (запертая с помощью mlock () system call), постранично (в страницах 1 стр. - 4 Кб)	4...4096
SHMPAGES	Полный размер совместно используемой памяти (включая IPC, совместно используемые упаковки (mappings) и объекты tmpfs), размещенной процессами конкретного VPS, в страницах.	512...16384
PRIWMPAGES	Размер частной (или потенциально частной) памяти, распределяемой приложением. Память, которая всегда делится между приложениями, не включена в ресурс этого параметра.	3072...151200- 3450...1612800
NUMFILE	Количество файлов, открываемых любыми процессами VPS.	512...8192
NUMFLOCK	Число замков для файлов, создаваемых всеми процессами VPS.	50...200 - 60...220
NUMPTY	Число псевдо-терминалов. Например, сессия ssh, экран (screen), приложение xterm потребляют ресурсы псевдо-терминала.	4...64
NUMSIGINFO	Количество структур siginfo (важно, что этот параметр ограничивает размер очереди поставки сигнала (signal delivery queue)).	256...512

DCACHESIZE	Полный размер dentry и inode структур, хранящихся в памяти. Например, приложение, сначала открывая файл <i>/etc/passwd</i> , закрывает входы, относящиеся к <i>etc</i> и <i>passwd inodes</i> . Если второе приложение открывает файл <i>/etc/shadow</i> – загружаются только входы, относящиеся к <i>shadow</i> .	184320...3932160-196608...4194304
PHYSPAGES	Полный объем RAM, используемого процессами. В настоящее время этот параметр используется как отчетно – учетный. Показывает использование RAM серверами VPS. Что касается страниц памяти, используемых разными серверами, только часть страницы (fraction) принадлежит конкретному VPS. Для всех VPS сумма использования <i>physpages</i> соответствует общему количеству страниц, используемых зарегистрированными в системе пользователями.	Не ограничена
NUMIPTENT	Число IP packet filtering entries (входов)	12... 128

*Параметры сети, или каким-либо образом связанные с сетью*, позволяют Вам устанавливать bandwidth параметры, hostname и IP адреса, которые могут использоваться серверами VPS, чтобы установить загружаемые на iptables модули VPS:

HOSTNAME	Если этот параметр определен, то <i>vzctl</i> установит ценность hostname до начала следующего запуска VPS. Этот параметр может быть опущен. В этом случае, администратор VPS должен сформировать hostname вручную.
IP_ADDRESS	Это - список IP адресов, которые могут использоваться в интерфейсах сети VPS. Этот список – действительный параметр VPS start call (запроса запуска VPS). Невозможно создать IP адрес изнутри VPS, если адрес не находится в списке. Любой IP адрес, созданный изнутри системы VPS будет доступен только в пределах VPS.
NAME SERVER	IP адрес сервера DNS для VPS. В отдельном формате может быть определено больше одного сервера.
SEARCHDOMAIN	Домены поиска DNS для VPS. Может быть определено больше одного домена.
NETDEV	Названия физических адаптеров сети, которые были перемещены сHardware Node на данный VPS.



IPTABLES

Убирает параметр IPTABLES с глобального файла конфигурации.

## Управление скриптами Open VZ

В этом разделе дается информация относительно скриптов Open VZ, используемых чтобы автоматизировать и выполнять некоторые операции и процедуры в пределах Вашей системы.

### Краткий обзор

Наряду с файлами конфигурации Open VZ, ответственными за конфигурацию системы, существует много скриптов Open VZ, позволяющих различными путями настроить деятельность VPS. Список скриптов:

<i>Название</i>	<i>Описание</i>
<i>/etc/sysconfig/vz-scripts/VPS_ID.action</i>	Скрипты частного действия VPS. Они позволяют выполнять определенные пользователем действия. Обычно этими действиями являются – запуск, остановка, установка, демонтаж.
<i>/var/lib/vzctl/ scripts/...</i>	Скрипты, которые должны быть исполнены при выполнении операций, связанных с VPS (например, добавление VPS нового IP адреса). Эти файлы должны быть определены в соответствующем файле конфигурации распределения.
<i>/etc/re.d/init.d/vz</i>	Скрипт запуска \ остановки системы V Open VZ. Этот скрипт является ответственным за надлежащий запуск и остановку системы Open VZ, включая модули загрузки.

### Скрипты действия Open VZ

Могут быть случаи, когда необходимо предпринять дополнительные действия в случае остановки или выключения конкретного VPS. Например, если Вы хотите получить доступ к системе файлов Host ОС (или к части файлов) с сервера VPS 101, Вы можете построить связь внутри VPS вручную от Host ОС. Однако после того как Вы перезагрузите VPS, ваша связь исчезнет, и Вы должны будете вручную набрать эту команду снова.

Система Open VZ позволяет Вам автоматизировать процедуры такого типа, используя скрипты действия системы Open VZ. В данной версии определено 6 скриптов:

<i>глобальный mount</i>	Этот скрипт начинает работать сразу после того, как vzctl установит частную область VPS. Сам же VPS пока еще не работает и скрипт работает только в контексте Host ОС.
<i>mount</i>	Этот скрипт начинает работать сразу после скрипта глобального mount. Сам VPS пока все еще не работает и скрипт упоминается только в контексте Host ОС.
<i>start</i>	После того как vzctl запустил VPS, начинает свою работу текущий скрипт. Скрипт работает в контексте VPS.
<i>stop</i>	Этот скрипт действует непосредственно перед закрытием VPS, в контексте VPS.
<i>umount</i>	После прекращения работы VPS, действует текущий скрипт и уже в контексте Host ОС.
<i>глобальный umount</i>	Он работает непосредственно перед тем, как vzctl удалит частную область VPS. Работает также в контексте Host ОС.

Обычный порядок выполнения скриптов действия «action scripts» показан на рисунке ниже. Скрипты *mount* и *umount* действуют скорее в контексте Host ОС, чем в контексте VPS.

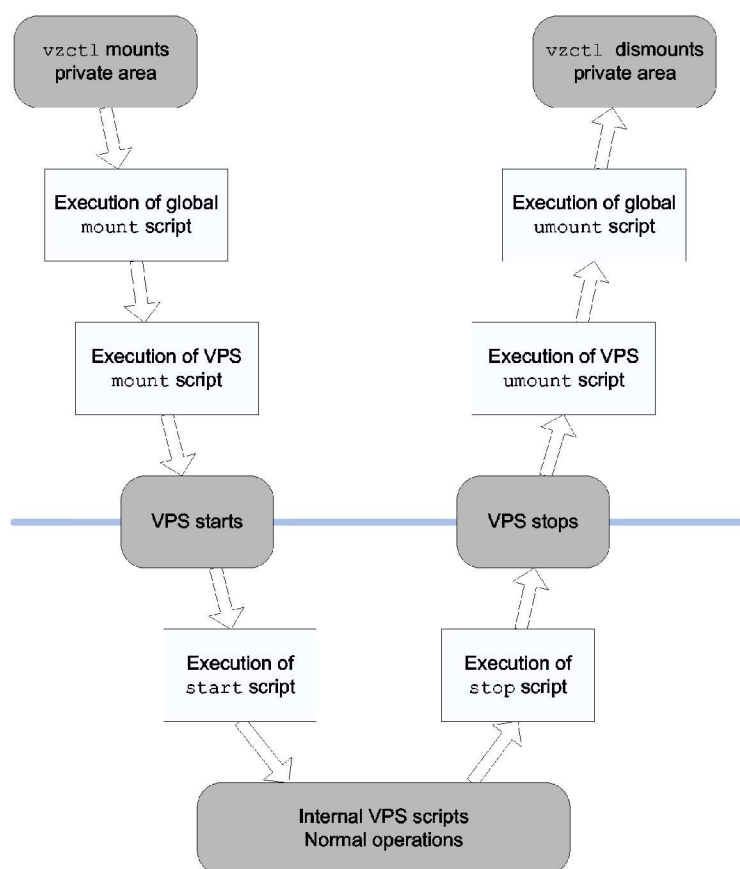


Рисунок 6 Последовательность исполнения скриптов действия

Важно понять, как `vzctl` обращается с выходными кодами скриптов действия. Если код выхода отличен от нуля, тогда `vzctl` постарается отменить действие скриптов `mount` и `start`. Другими словами, если скрипт запуска (`start`) возвратит ошибку, `vzctl` остановит VPS, и если будет допущена ошибка в одном из `mount` скриптов, то `vzctl` демонтирует частную область VPS. Необходимо отметить, что в этом случае `vzctl` не выполнит скрипты `stop` и `umount` вообще.

**Предостережение:** Выполняя `vzctl start`, действуют скрипты и `start` и `mount`. Однако, если что-то случится со скриптом `start`, не будут действовать ни `stop` ни `umount`. В результате `vzctl` может стать неспособным демонтировать частную область VPS, если Вы настроите дополнительные крепления в `mount` скриптах и демонтируете их в `umount` скриптах.

Та же ситуация со скриптами `mount` и `stop`. Если скрипт возвращает ошибку, то действие не будет предпринято. Будьте осторожны, так как это позволяет создавать серверы VPS, не останавливаемые командой `vzctl`.

Скрипты действия расположены в той же директории, что и файлы конфигурации VPS, под названием `vpsid.action`. Глобальные скрипты называются `vps.mount` и `vps.umount`, другие содержат в имени соответствующий ID VPS. Имена глобальных скриптов зафиксированы, к ним обращаются лишь при запуске и остановке VPS. Поэтому в этих скриптах необходимо выполнять только приемлемые для всех VPS команды же, оставляя специфические команды скриптам конкретных серверов. Так, например, для сервера VPS 101 названия скриптов будут следующими:

- `/etc/sysconfig/vz-scripts/vps.mount`
- `/etc/sysconfig/vz-scripts/101.mount`
- `/etc/sysconfig/vz-scripts/101.start`
- `/etc/sysconfig/vz-scripts/101.stop`
- `/etc/sysconfig/vz-scripts/101.umount`
- `/etc/sysconfig/vz-scripts/vps.umount`

Для скриптов `mount` и `umount`, бывшая среда (параметры) является `parent` (родной) стандартной средой (то есть `vzctl`) с двумя дополнительными переменными: `$VEID` и `$VE_CONFFILE`. Первая устанавливает ID VPS (запускает, останавливает, демонтирует), вторая имеет полный доступ к файлу конфигурации VPS. Это немного избыточно, но SWsoft ввел эти две переменные для удобства. Вы можете использовать следующий фрагмент кода в `bash` скриптах, чтобы получить доступ к дополнительной информации о VPS, такой как местоположение `$VE_PRIVATE` или `$VE_ROOT`:

```
#!/bin/bash
#
# This script source VPS configuration files in the same
# order as vzctl does

# if one of these files does not exist then something is
# really broken
[ -f /etc/sysconfig/vz ] || exit 1
[ -f $VE_CONFFILE ] || exit 1

# source both files. Note the order, it is important
. /etc/sysconfig/vz
. $VE_CONFFILE
```

Скрипты *start* и *stop* выполняются в контексте VPS. Если эти скрипты обращаются к каким-либо внешним командам, команды берутся непосредственно из VPS. Имейте в виду, что скрипт *start* начинает свою работу еще до выполнения каких-либо операций VPS (включая *init*). Таким образом, в этот момент, система файлов */proc* на VPS еще не установлена и приложения, использующие */proc*, могут просто не функционировать.

## 2. *Интерфейс командных строк Open VZ (OpenVZ Command Line Interface)*

OpenVZ содержит ряд инструментов для командных строк. Эта глава рассматривает утилиты, которые поддерживаются OpenVZ. Для каждой из утилит описаны все доступные варианты командных строк и их выключатели.

### **Матрица утилит командных строк OpenVZ.**

Эта таблица содержит полный список утилит командных строк OpenVZ.

*Общие утилиты* предназначены для того, чтобы выполнить ежедневные задачи обслуживания:

#### ***vzctl***

Утилита для управления VPS .

#### ***vzlist***

Утилита для рассмотрения списков (с дополнительной информацией) VPS, существующих на Node.

#### ***vzquota***

Утилита для управления дисковыми квотами OpenVZ.

*Инструменты управления шаблонами*, для создания шаблонов, обслуживания и установки приложений на VPS:

#### ***vzpkgls***

Утилита для получения списка шаблонов, доступных на hardware node и VPS .

### ***vzpkgcache***

Создание/обновление ряда шаблонов cache .

### ***vzrpm***

Простая обертка (wrapper) rpm для использования rpm на VPS.

### ***vzyum***

Простая обертка (wrapper) yum для использования yum на VPS .

*Дополнительные инструменты* выполняют множество задач и используются другими утилитами OpenVZ:

### ***vzdqcheck***

Печатает информацию об использовании текущего файлового места с позиций дисковой квоты.

### ***vzdqdump u vzdqload***

Утилиты, чтобы переместить лимиты долей пользователя/группы VPS и периоды grace из ядра или файла квоты или же для их загрузки на файл квоты.

### ***vzcpucheck***

Утилита для проверки использования центрального процессора VPS

### ***vzmemcheck***

Утилита для проверки текущих параметров памяти HN и VPS.

### ***vzcalc***

Утилита для вычисления использования ресурсов VPS

### ***vzpid***

Утилита для печати Id VPS, к которому принадлежит процесс.

### ***vzsplit***

Утилита для создания образцов файлов конфигурации VPS путем деления Node на равные части

### ***vzcfgvalidate***

Утилита для проверки правильности составления файлов конфигурации VPS.

## **vzctl**

vzctl - основной инструмент для управления VPS. Чтобы использовать его вам нужно войти на HN в качестве корневого пользователя. Синтаксис vzctl таков:

```
vzctl [verbosity-options] command vpsid [command-specific-options]
```

Где command может быть одной из следующих:

<i>Create</i>	Используется для создания VPS и базы для нескольких объединенных VPS
---------------	--

<i>Destroy</i>	Используется для разрушения VPS
<i>Mount</i>	Позволяет устанавливать частные области VPS и извлечь скрипт mount
<i>Umount</i>	Позволяет демонтировать частные области VPS и извлекать скрипт umount
<i>Start</i>	Запускает VPS
<i>Stop</i>	Останавливает VPS
<i>Restart</i>	Перезагружает VPS
<i>Status</i>	Отображает статус VPS
<i>Set</i>	Используется для установки параметров VPS, включая настройки контроля ресурсов, местоположение частных областей, VPS hostname, IP адресов и пользовательских паролей VPS
<i>Enter</i>	Обеспечивает беспарольный вход администратора hardware node на VPS. Используйте эту команду с осторожностью и никогда не прибегайте к ней на подозрительных VPS.
<i>Exec, exec 2</i>	Эти две команды позволяют управлять произвольными командами в VPS без подключения к VPS. Различием между двумя является – статус возврата.

Опции *verbosity* могут использоваться с любой из вышеупомянутых команд, и к *verbosity* относятся следующие:

- - verbose (многословный)	Отвергает установку LOG_LEVEL из глобального файла конфигурации OpenVZ <code>/etc/sysconfig/vz</code> и устанавливает уровень на максимально возможную ценность для сессии <code>vzctl</code> .
- - quiet (тихий)	Обнуляет процесс регистрации на экране и в файле регистрации

### **vzctl create**

Эта команда используется для создания нового VPS, она имеет следующий вид:

```
vzctl create vpsid [--ostemplate name] [--config name]
[--private path] [--root path]
```

С помощью этой команды Вы можете создать VPS. Идентификационный номер VPS `vpsid`, который требуется для выполнения этой команды, будет уникальным для всего hardware node.

---

**Замечание:** VPS с ID от 1 до 100 сохраняются для внутренних потребностей OpenVZ. Не используйте номера от 1 до 100 для ваших Частных Виртуальных Серверов.

---

Аргументы команд выглядят следующим образом:

-- osteamplate name	Обозначает набор пакетов (шаблонов ОС), для использования при создании VPS. В случае опущения, данные будут взяты из основного файла OpenVZ (DEF_OSTEMPLATE parameter)
-- config name	По выбору. В случае использования, vzctl копирует данные с типового файла конфигурации VPS, расположенного в /etc/sysconfig/vz-scripts, называя их <i>ve-name.conf-sample</i> .  Типичные файлы конфигурации обычно имеют множество ограничений контроля ресурсов для VPS. Если Вы пропускаете эту функцию и имя файла конфигурации по умолчанию остается неопределенным в глобальном файле конфигурации, Вам придется дополнительно устанавливать параметры контроля ресурсов для VPS при помощи команды <code>vzctl set</code> прежде, чем запускать VPS.
-- private path	По выбору. При использовании определяет путь к частной области VPS. Эта опция используется для пересечения пути по умолчанию к частной области VPS из файла конфигурации /etc/sysconfig/vz (Переменная VE_PRIVATE). Аргумент может содержать строку \$VEID, которая впоследствии будет заменена на числовую составляющую идентификационного номера VPS.
--root path	По выбору. При использовании определяет путь к mount point конечной директории VPS. Эта опция используется для пересечения пути по умолчанию к корневой директории из файла конфигурации /etc/sysconfig/vz (переменная VE_ROOT). Аргумент может содержать строку \$VEID, которая впоследствии будет заменена на числовую составляющую идентификационного номера VPS.

Создавая новый VPS, вам необходимо определить его идентификационный номер. Каких-то ограничений здесь нет, кроме уникальности с точки зрения vzctl. Однако, желательно использовать номера, варьирующиеся в разных средах hardware node. Например, Вы можете использовать номера от 101 до 2000 на первом узле, номера от 2001 до 4000 на втором и так далее. Это поможет Вам в поисках nodes, на которых был создан данный VPS, и устранил возможные противоречия между номерами VPS и между nodes.

### **vzctl destroy**

Вид этой команды следующий:

```
vzctl destroy vpsid
```

Эта команда используется, чтобы удалить ненужный VPS . Она физически удаляет все файлы, расположенные в частной области VPS (определенные как переменная `VE_PRIVATE` в файле конфигурации VPS), и переименовывает файл конфигурации VPS в `/etc/sysconfig/vz-scripts/` с `vpsid.conf` на `vpsid.conf.destroyed`. Она в подобной манере переименовывает и скрипты действия VPS.

Эта команда не нуждается в дополнительных аргументах, останавливая работу VPS и его частных областей.

### **vzctl start, vzctl stop, vzctl restart, and vzctl status**

Эти четыре команды имеют сходный вид и не требуют обязательных аргументов

```
vzctl start vpsid  
vzctl stop vpsid [--fast]  
vzctl restart vpsid  
vzctl status vpsid
```

Первая команда используется для запуска сервера. Она настраивает интерфейс сети, инициализирует квоту VPS, если нужно, запускает процесс `init` на VPS.

Запуская VPS, `vzctl` может выполнять типичные скрипты, расположенные в директории `/etc/sysconfig/vz-scripts`, в порядке их следования друг за другом.

<code>vpsid.mount</code>	Дополнительный (по выбору) установочный скрипт VPS. Если он существует, тогда он извлекается сразу же после установки частной области VPS. Если он имеет отличный от нуля статус тогда <code>vzctl</code> демонтирует частную область VPS и выводит ошибку
<code>vpsid.start</code>	Дополнительный скрипт запуска VPS. Если он существует, тогда он приводится в исполнение только в случае запущенного VPS.

`vzctl stop` останавливает работу VPS. Если VPS не закрывается после двухминутного перерыва,



например, из-за какой-то ошибки в приложениях, `vzctl` в любом случае остановит идущие на VPS процессы, но с частичным разрушением сервера. Чтобы избежать двухминутного ожидания и фактического разрушения сервера Вы можете использовать опцию `--fast` этой команды. Нормальная последовательность закрытия `vzctl stop` описана ниже в порядке ее выполнения:

<code>vpsid.stop</code>	Дополнительный скрипт закрытия VPS .Если он существует, тогда он приводится в действие в первую очередь на VPS. Если он имеет при этом статус, отличный от нуля, тогда <code>vzctl</code> не останавливает VPS.
<code>vpsid.umount</code>	Дополнительный <code>umount</code> скрипт VPS. Если он существует, тогда он исполняется после того, как VPS был остановлен, но прежде, чем его частная область демонтируется.

Вам следует использовать скрипты (`mount/umount` и `start/ stop`), если бы Вы хотите произвести какие-то действия с запуском или закрытием VPS.

`vzctl restart vpsid` команда последовательно выполняет закрытие и запуск соответствующих VPS.

`vzctl status vpsid` команда отражает текущее состояние VPS. Она выводит следующую информацию: существует ли частная область VPS, установлена ли она и работает ли сам VPS. Пример ниже показывает как это может выглядеть

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

### **vzctl mount and vzctl umount**

Эти команды не нуждаются в дополнительных аргументах:

```
vzctl mount vpsid
vzctl umount vpsid
```

Первая команда устанавливает частную область VPS на корневую директорию VPS (`/vz/root/vpsid/на hardware node`) без запуска самой области. Обычно Вам не нужно прибегать к этой команде, поскольку команда `vzctl start` устанавливает частную область VPS автоматически.

`vzctl umount` команда не устанавливает частную область VPS. Обычно также нет никакой потребности в использовании этой команды, поскольку `vzctl stop` автоматически демонтирует частную область VPS.

### **vzctl set**

Эта команда используется для того, чтобы установить параметры VPS. Она имеет следующий вид:

```
vzctl set vpsid --setting_name value [...] [ --save ]
```

Дополнительный переключатель `--save` указывает `vzctl` сохранить ли изменения в файле конфигурации VPS `/etc/sysconfig/vz-scripts/vpsid.conf`. Фактически все параметры настройки VPS могут быть изменены в процессе работы без перезагрузки VPS. Исключения составляют `--onboot`, `--quotauidlimit`, `--capability`, `--private`, and `--root`.

Параметры настройки, определенные в этом файле могут быть подразделены на следующие категории:

- смешанные параметры,
- параметры сетевой организации
- параметры управления ресурсами.

**Смешанные** параметры настройки:

- onboot yes no	Этой настройке требуется переключатель <code>-save</code> . Если Вы ставите его в позицию "да", тогда OpenVZ автоматически запускает VPS при следующем запуске системы.
- userpasswd user:password	Это урегулирование создает на VPS нового пользователя с указанным паролем, или изменяет пароль уже существующего пользователя. Эта команда изменяет не сам файл конфигурации VPS, а файлы <code>/etc/passwd</code> и <code>/etc/shadow</code> внутри VPS. В случае, если корень VPS не установлен, он устанавливается для внесения изменений, а затем демонтируется.
- noatime yes no	Устанавливает <code>noatime</code> флажок (не обновляйте inode access times) на файловой системе VPS. - <b>yes</b> по умолчанию для VPS Класса 1, и <b>no</b> для всех остальных серверов.
- devnodes device:r w rw none	Разрешает доступ VPS к указанным устройствам указанным способом - только чтением, только записью, или чтением - записью- или отрицает любой доступ. Например: <code>- devnodes hda1:rw</code> Устройство должно присутствовать в VPS <code>/dev</code> директории, в противном случае, будет автоматически создано новое устройство.
- netdev_add name	Перемещает указанное сетевое устройство с hardware Node на данный VPS. Например: <code>- netdev add eth0</code>

- netdev_del name	Перемещает указанное сетевое устройство с данного конкретного VPS на hardware Node.
- capability:on off	<p>Определяет возможности VPS. На сервере разрешена настройка следующих возможностей:</p> <p>AC_OVERRIDE,  AC_READ_SEARCH, CHOWN, FOWNER, FSETID,  IPC_LOCK, IPC_OWNER, УБИВАЮТ, АРЕНДУЮТ,  LINUX_IMMUTABLE, MKNOD, NET_ADMIN,  NET_BIND_SERVICE, NET_BROADCAST,  NET_RAW, SETGID, SETPCAP, SETUID,  SYS_ADMIN, SYS_BOOT, SYS_CHROOT,  SYS_MODULE, SYS_NICE, SYS_PACCT,  SYS_PTRACE, SYS_RAWIO, SYS_RESOURCE,  SYS_TIME, SYS_TTY_CONFIG.</p>
- root path	Эта настройка не перемещает root mount пункт вашего сервера на новый путь. Она просто отвергает/не принимает параметр VE_ROOT в файле конфигурации VPS.
- private path	<p>Эта настройка не перемещает частную область вашего сервера на новую дорожку.</p> <p>Она просто не принимает параметр VE_PRIVATE в файле конфигурации VPS. Вам следует пользоваться этой командой только в том случае, когда вам нужно вручную переместить частную область вашего сервера в новую директорию, или в том случае, когда вы хотите обновить файл конфигурации вашего сервера.</p>
- setmode restart ignore	Эта команда решает, должна ли утилита перезагружать VPS после применения всех параметров, которые фактически требуют перезагрузки сервера для того, чтобы самим вступить в силу.

**Параметры управления ресурсами** контролируют количество ресурсов, которое может потреблять VPS. Если настройка имеет bar:lim после себя, то очевидно требуется дополнительная спецификация свойств как bars так и lims, разделенных двоеточием.

<p><code>--applyconfig</code> <i>name</i></p>	<p>Эта опция позволяет Вам устанавливать параметры ресурсов для VPS не один за другим, а просто считывая их с файла-образца конфигурации VPS. Все файлы-образцы конфигурации VPS расположены в директории <code>/etc/sysconfig/vz-scripts</code> и обычно называются : <code>ve-name.conf-sample</code>, таким образом все, что вам необходимо сделать - это определить только часть <code>name</code> соответствующего названия образца после опции – <code>applyconfig</code>. Заметьте, что пробелов в файлах-образцах конфигурации содержаться не должно.</p> <p>Опция - <code>applyconfig</code> осуществляет перенос всех параметров с файла-образца на данный VPS. Исключением являются <code>OSTEMPLATE</code>, <code>VE_ROOT</code>, и <code>VE_PRIVATE</code> параметры в случае, если они вообще существуют в файле-образце.</p>
<p><code>--numproc</code> <i>bar:lim</i></p>	<p>Число разрешенных процессов и тредов. В случае превышения этого предела VPS не сможет запустить новый процесс. В этой версии Open VZa, должен быть установлен предел, равный по величине барьеру.</p>
<p><code>--numtcpsock</code> <i>bar:lim</i></p>	<p>Число гнезд TCP (PF_INET Family, SOCK_STREAM types). Этот параметр ограничивает количество TCP соединений и, таким образом, количество клиентов сервера, с которыми может одновременно работать данное приложение. В этой версии OpenVZ, следует установить величину предела равную величине барьера.</p>
<p><code>--numothersock</code> <i>bar:lim</i></p>	<p>Число гнезд, отличных от TCP. Местные гнезда (UNIX-domain), используются для осуществления связей внутри системы. Гнезда UDP используются для вопросов, связанных, например, с Обслуживанием Названия Областей (DNS). В этой версии OpenVZ, следует установить величину предела равную величине барьера</p>

<p>--vmguarpages <i>bar:lim</i></p>	<p>Гарантия распределения памяти, измеряемая в страницах (одна страница - 4 КБ). Приложения, как указано, могут распределять память, до тех пор пока количество памяти, подсчитанной на <code>privvmpages</code> не превышает барьера, формируемый на основе <code>vmguarpages</code> параметра. В случае превышения барьера, а также при полной нехватки доступной к распределению памяти, ее распределение может не удалиться. В этой версии OpenVZa, должен быть установлен предел, равный по величине барьеру.</p>
<p>--kmemsize <i>bar:lim</i></p>	<p>Размер <code>unswappable</code> ядерной памяти (в байтах), распределяемой для процессов внутренних ядерных структур отдельно взятого VPS. Обычное количество ядерной памяти - 16-50 КБ для каждого из процессов.</p>
<p>--tcpsndbuf <i>bar:lim</i></p>	<p>Полный объем (в байтах) <code>send</code> буферов, посланного на TCP гнезда - количество ядерной памяти, распределяемой для данных, посылаемых приложениями на TCP гнездо, но еще не признанное удаленной стороной.</p>
<p>--tcprecvbuf <i>bar:lim</i></p>	<p>Общий объем (в байтах) <code>receive</code> буферов для TCP гнезд. Количество ядерной памяти, полученной удаленной стороной, но все еще не прочитанной местным приложением.</p>
<p>--othersockbuf <i>bar:lim</i></p>	<p>Полный размер (в байтах) буферов гнезда <code>UNIX-domain</code>, <code>UDP</code> и других датаграмных протоколов <code>receive</code> буферов.</p>
<p>--dgramrecvbuf <i>bar:lim</i></p>	<p>Полный размер в байтах полученных буферов <code>UDP</code> и других датаграмных протоколов</p>
<p>--oomguarpages <i>bar:lim</i></p>	<p>Гарантии, не зависящие от памяти, в страницах в размере 4 КБ. Любой процесс VPS не будет остановлен даже в случае острой нехватки памяти, если текущее потребление памяти (включая физическую память и <code>swap</code>), не достигнет барьера <code>oomguarpages</code>. В этой версии OpenVZ, величина установленного предела должна равняться величине барьера.</p>

<p>--lockedpages <i>bar:lim</i></p>	<p>Память, которая не подлежит функции замены- swar (локализуются в mlock () по запросу системы), измеряется в страницах в размере 4 КБ</p>
<p>--shmpages <i>bar:lim</i></p>	<p>Общий объем памяти, предназначенной для совместного использования (включая IPC, совместно используемые анонимные картографии и объекты tmpfs). Память распределяется по процессам отдельно взятого VPS, измеряется в страницах на 4 КБ.</p>
<p>--privvmpages <i>bar:lim</i></p>	<p>Размер в 4 КБ страницах частной (или потенциально частной) памяти, ассигнованной приложениям VPS. Память, которая обычно является общей для всех приложений не включена в этот параметр ресурсов.</p>
<p>--numfile <i>bar:lim</i></p>	<p>Число файлов, открытых для всех процессов VPS. В этой версии OpenVZ, величина установленного предела должна равняться величине барьера.</p>
<p>--numflock</p>	<p>Число файловых замков, созданных всеми процессами VPS.</p>
<p><i>bar:lim</i>  numpty <i>bar:lim</i></p>	<p>Число псевдотерминалов. Например, приложения ssh сессии, screen, xterm потребляет ресурсы псевдотерминалов. В этой версии OpenVZ, величина установленного предела должна равняться величине барьера.</p>
<p>numsiginfo <i>bar:lim</i></p>	<p>Число структур siginfo (по существу этот параметр ограничивает размер очередности в доставке сигналов). В этой версии OpenVZ, величина установленного предела должна равняться величине барьера</p>
<p>--dcache <i>bar:lim</i></p>	<p>Полный размер (в байтах) структур dentry и inode, заключенных в памяти. Это отдельный параметр, существующий для определения предела, который бы сообщал о нехватке памяти файловым процессам и возвращал сигнал об ошибке приложениям, защищая таким образом, от чрезмерного потребления памяти.</p>

<p><code>--cpuunits</code> <i>units</i></p>	<p>Допускаемая мощность CPU -центрального процессора. Это - положительное целое число, которое определяет минимальную гарантируемую долю Центрального процессора, которую получит VPS. Вы можете рассчитать эту долю по формуле: <math>((VPS\ CPUUNITS)/(сумма\ CPU\ UNITS\ на\ всех\ работающих\ Virtual\ Private\ Servers))*100\%</math>.</p> <p>Общая мощность центрального процессора зависит от самого центрального процессора и OpenVZ и соответствующих инструментов, учитывая, что один процессор intel PIII на 1 ГГц является эквивалентом 50 000 единиц центрального процессора.</p>
<p><code>--cpulimit</code> <i>percent</i></p>	<p>Это - положительное число, указывающее в процентах время центрального процессора, которое не позволено превышать соответствующему VPS</p>
<p><code>--diskspace</code> <i>bar:lim</i></p>	<p>Полный размер дискового пространства, потребляемого VPS, в 1 КБ блоках. Когда место, используемое VPS переходит барьер, VPS может ассигновать дополнительное место на диске до предела в течение всего периода, определенного в настройке quotatime.</p>
<p><code>--diskinodes</code> <i>bar:lim</i></p>	<p>Общее количество дисковых inodes (файлы, директории, символные связи), которые может может VPS. Когда число inodes, используемых VPS превышает дозволенный предел / барьер, VPS может создать дополнительные файловые записи, которые будут валидны в течение всего периода, определенного в настройке quotatime.</p>
<p><code>--quotatime</code> <i>seconds</i></p>	<p>Период grace дисковой доли. Измеряется в секундах. VPS позволено временное превышение численных величин барьера для дискового пространства и предела дисковых inodes, но не дольше чем на период, определенный этой настройкой.</p>
<p><code>--quotauidlimit</code> <i>num</i></p>	<p>Число ID пользователя/группы, разрешенных дисковой доле второго уровня VPS (доля пользователя на VPS). Если данный параметр установлен на 0, доля UID/GID не будет разрешена.</p>

**Параметры сетевой организации** позволяют Вам настраивать hostname, области поиска в

том случае, когда используются названия `domain`, которые не были до конца определены, а также настраивать адрес сервера DNS и IP адреса, которые может использовать VPS. Кроме того, эти параметры позволят Вам указывать те модули `iptables`, которые могут загружаться на данный VPS:

<code>hostname name</code>	Меняет\устанавливает <code>hostname</code> на указанное название
<code>- ipadd addr</code>	Добавляет, адрес IP к списку IP адресов, которые может использовать VPS, создавая сетевой интерфейс с указанным именем на данном VPS.
<code>- ipdel addr all</code>	Позволяет Вам анулировать IP адрес на данном VPS. При использовании "all " на месте адреса IP отменяются все адреса IP.
<code>- nameserver addr</code>	Адрес IP сервера DNS для VPS. В формате с использованием пробелов можно определить больше одного сервера.
<code>- searchdomain domain</code>	Области поиска DNS для VPS. Может быть определено несколько областей
<code>- iptables module</code>	Только те модули <code>iptables</code> , которые обозначены, загрузятся на VPS . Вот список возможных модулей:  <code>ip_conntrack,</code> <code>ip_conntrack_ftp, ip_conntrack_irc,</code> <code>iptables_filter, ipt_length, ipt_limit,</code> <code>ipt_LOG, iptable_mangle, ipt_conntrack,</code> <code>ipt_helper, ipt_state, ipt_tcpmss,</code> <code>ipt_tos, ipt_multiport, iptable_nat,</code> <code>ip_nat_ftp, ip_nat_irc, ipt_REJECT,</code> <code>ipt_TCPMSS, ipt_TOS, ipt_ttl.</code>

### **`vzctl exec`, `vzctl exec2`, и `vzctl enter`**

Эти команды используются для выполнения арбитрных (произвольных) команд внутри корневого сервера на Hardware Node. Синтаксис этих команд следующий:

```
vzctl exec|exec2 vpsid command
vzctl enter vpsid
```

`command` в этом случае обозначает строку, которую необходимо исполнить на VPS. Если `command` определена как “-“ команды исполнения будут читаться с обыкновенного входа до конца файла или же пока будет использована команда «`exit`».

Разница между `exec` и `exec2` заключается в коде выхода. `vzctl exec` возвращает 0 в том случае, если `vzctl` был в состоянии выполнить команду - то есть он не принимает во внимание код выхода.



`vzctl exec2` возвращает код выхода команды, выполненной на сервере.

Используя `exec` или `exec2`, необходимо помнить, что оболочка разбирает командную строку, и если у Вашей команды имеются оболочечные мета-характеры, необходимо их отменить. ( по другому это предложение никак не перевести)

`vzctl enter` похож на `vzctl exec /bin/bash`. Разница состоит в том, что `vzctl enter` заставляет интерпретатора оболочки полагать, что `vzctl enter` связан с терминалом. Вы получаете подсказку оболочки, а также возможность выполнять многочисленные операции, будто бы Вы загружены в VPS.

Однако, знайте, что `vzctl enter` - потенциально опасная команда, если на VPS имеются заведомо небезопасные пользователи. Файловые дескрипторы оболочки будут доступны для корня VPS в системе файлов `/proc`, т.е. злонамеренный пользователь сможет управлять `ioctl calls`.

**Никогда не** используйте `vzctl enter` на серверах, которымне доверяете! Именно поэтому, `vzctl enter` является резервным способом соединения с серверами VPS, и не является полным заместителем `ssh`. Поэтому, у него есть ограничения. Например, Вы не можете установить `ssh` соединения пока Вы подключены к VPS через `vzctl enter`.

## vzlist

Утилита `vzlist` используется, чтобы составить список серверов VPS на Hardware Node и узнать дополнительную информацию о них. Вывод и сортировка этой информации могут быть настроены. Утилита имеет следующий синтаксис:

```
vzlist [-a] [-S] [-o parameter[.specifier] \
[,parameter[.specifier]...]] [-s [-]parameter[.specifier]] \
[-H] [-h hostname_pattern] [vpsid ...]

vzlist -L|--list
```

Следует описание доступных опций:

Опция	Описание
<code>-a, --all</code>	Список серверов, существующих на Node. По умолчанию, указаны лишь действующие.
<code>-S, --stopped</code>	Список приостановленных серверов
<code>-o parameter[.specifier]</code>	Данная опция используется для указания лишь конкретной информации о серверах. Параметры и спецификаторы могут быть использованы после того как опция <code>-o</code> будет внесена в список этой подсекции. Чтобы несколько параметров появились на листе вывода одновременно, необходимо отделить каждого из них запятой, так как это показано выше.

<i>-s, --sort</i> <i>[-]parameter[.specifier]</i>	Сортирует серверы по определенному признаку. Если стоит перед названием параметра стоит “-“, порядок сортировки меняет направление.
<i>-h, --hostname</i> <i>hostname_pattern</i>	Отображает только те серверы, которые соответствуют определенному образцу. Могут использоваться следующие обобщающие характеристики: *,?, и [ ]. Замечание: последняя характеристика должна быть устранена, чтобы избежать интерпретации оболочки.
<i>-H, --no-header</i>	Не показывает заголовки колонок.
<i>vpsid</i>	Отображает только серверы с определенным ID. Может быть специфицировано несколько ID VPS, отделенных пробелом. Если ID = -1, утилита выдает список (без дополнительной информации) тех ID, серверы которых расположены на Node.
<i>-L, --list</i>	Выдает список параметров, доступных для использования с помощью опции -o.

### Параметры вывода *vzlist* и их спецификаторы (*vzlist* Output Parameters and Their Specifiers)

Практически любой параметр, который может использоваться после -o и -s переключателей утилиты *vzlist*, может быть определен с помощью комбинации «точка+буква», следующей за параметром и обозначающей следующее:

.m	Максимальное использование соответствующего ресурса данным сервером.
.b	Барьер для VPS, установленный для ограничения использования ресурса.
.l	Лимит для VPS, установленный для ограничения использования ресурса.
.f	Количество раз, когда система не смогла выделить ресурс для VPS.
.s	Soft лимит на использование соответствующего ресурса для VPS.
.h	Hard лимит на использование соответствующего ресурса для VPS.

За более подробной информацией о барьерах и лимитах обращайтесь к разделу **Управление ресурсами**.

Параметры, используемые утилитой:

<i>Параметр</i>	<i>Возможные указатели \ спецификаторы</i>	<i>Колонка вывода</i>	<i>Описание</i>
-----------------	--	-----------------------	-----------------

vpsid	никаких	VEID	ID сервера
hostname	никаких	HOSTNAME	hostname сервера
ip	никаких	IP_ADDR	IP адрес сервера
status	никаких	STATUS	Определяет работает ли сервер или нет
kmemsize	.m, .b, .1, .f	KMEMSIZE	Размер неизменной корневой памяти (в байтах), предназначенной для конкретных ядерных структур (процессов) определенного сервера. Обычный объем памяти – 16...50 Кб на процесс.
lockedpages	.m, .b, .1, .f	LOCKEDP	Память, не подвергающаяся никаким изменениям (блокирована <code>mlock () system call</code> ), на 4 Кб страницах
privvmpages	.m, .b, .1, .f	PRIWMP	Объем частной (или потенциально частной) на 4 Кб страницах, выделенной приложениями VPS. В этот параметр не включена память, которая является общей для нескольких приложений.
shmpages	.m, .b, .1, .f	SHMP	Общий объем совместно используемой памяти (включая IPC, совместно используемые анонимные mappings, и объекты <i>tmpfs</i> ), выделенной процессами частного сервера. Представлена на 4 Кб страницах.
numproc	.m, .b, .1, .f	NPROC	Количество разрешенных процессов и тредов.

physpages	.m, .b, .1, .f	PHYSP	Общий размер RAM, используемого процессами. В настоящее время это – только вычислительный параметр. Отображает использование RAM различными VPS. На страницах памяти, используемой несколькими серверами, только часть страницы принадлежит одному серверу. Сумма использования <i>physpages</i> для всех серверов соответствует общему количеству страниц, используемых зарегистрированными пользователем.
vmguarpages	.m, .b, .1, .f	VMGUARP	Гарантия распределения памяти, представлена в страницах: 1 стр. – 4 Кб). Приложения имеют возможность распределять память, в то время как объем <i>privvmpages</i> памяти не превышает установленного барьера параметра <i>vmguarpages</i> . Превышение барьера: в случае всеобщей нехватки памяти распределение памяти может не состояться,
oomguarpages	.m, .b, .1, .f	OOMGUARP	Гарантия out-of-memory, представлена в виде 4 Кб страниц. Даже в случае жесточайшей нехватки памяти процессы VPS не будут остановлены, если конечно текущее потребление памяти (включая физическую и подкаченную (swap)) не достигнет барьера <i>oomguarpages</i> .

numtcpsock	.m, .b, .l, .f	NTCP SOCK	Количество TCP гнезд (PF_INET family, SOCK_STREAM type). Этот параметр ограничивает количество TCP соединений, и соответственно количество клиентов, которых одновременно может обслуживать приложение сервера.
numflock	.m, .b, .l, .f	NFLOCK	Количество файловых замков, созданных процессами VPS.
numpty	.m, .b, .l, .f	NPTY	Количество псевдотерминалов. Ресурсы псевдотерминалов потребляются <i>ssh</i> сессией, <i>экраном</i> , <i>xterm</i> приложением.
numsiginfo	.m, .b, .l, .f	NSIGINFO	Количество структур <i>siginfo</i> (этот параметр ограничивает очередь сигнальной доставки)
tcpsndbuf	.m, .b, .l, .f	TCPSNDB	Общий размер (в байтах) send буферов для гнезд TCP – объем корневой памяти, выделенной для пересылки данных от приложения гнезду TCP, не еще не признанной отдаленной стороной.
tcprecvbuf	.m, .b, .l, .f	TCPRCVB	Общий размер (в байтах) receive буферов для гнезд TCP. Объем корневой памяти, получаемой отдаленной стороной, но еще не признанной местным приложением.
othersockb	.m, .b, .l, .f	OTHERSOCKB	Общий размер гнездовых UNIX-domain буферов, UDP и других протоколов send буферов.
dgramrecvbuf	.m, .b, .l, .f	DGRAMRCVB	Общий объем (в байтах естественно) receive буферов UDP и других диаграммных протоколов.

nothersock	.m, .b, .l, .f	NOTH SOCK	Количество не NCP гнезд. Местные гнезда (UNIX-domain) для коммуникаций внутри системы. Гнезда UDP используются для запросов Domain Name Service (DNS), например.
dcachesize	.m, .b, .l, .f	DCACHESIZE	Общий размер (в байтах конечно) dentry и inode структур, хранящихся в памяти. Существует в виде отдельного параметра с целью создания операций - ограничителей, чтобы вовремя обнаружить нехватку памяти и вернуть ошибку приложениям, защищающим от избыточного её использования.
numfile	.m, .b, .l, .f	NFILE	Количество файлов, открываемых процессами VPS.
numiptent	.m, .b, .l, .f	NIPTENT	Количество фильтрующих пакеты IP входов.
diskspace	.s, .h	DQBLOCKS	Общий объем дискового пространства, занимаемого VPS, в 1 Кб блоках. Когда место, занимаемое сервером, достигает определенного барьера, VPS может занять дополнительное прост-ранство, но тоже до определенного лимита, учитывая grace период.
diskinodes	.s, .h	DQINODES	Общее количество дисковых inodes (файлов, директорий, символических связей), выделяемое сервером. В случае достижения барьера (количесиво inodes тоже должно быть ограничено), сервер может создать дополнительное количество entries, учитывая grace период.

lverage	никаких	LAVERAGE	Среднее число процессов, готовых к действию в течение последних 1, 5 и 15 минут.
cpulimit	никаких	CPULIM	Положительное число, обозначающее время(%) CPU, которое не должен превысить соответствующий сервер.
cpuunits	никаких	CPUUNI	Допустимая мощность CPU. Это положительное целое число, обозначающее минимально гарантированный размер CPU, который достанется VPS. Число можно вычислить следующим образом: ((VPS CPUUNITS)/(Сумма всех CPU UNITS по всем работающим серверам VPS))* 100%. Общая мощность зависит от самого CPU и OpenVZ reporting tools, приравнивающих 1 GHz PNI Intel processor к 50,000 CPU единицам.

Текущее потребление ресурса будет показано по умолчанию, если параметр, обычно использующийся со спецификатором, стоит в командной строке без него.

### **vzquota**

Эта команда используется, чтобы сформировать и наблюдать за статистикой дисковой доли для серверов. **vzquota** также используется для получения возможности использования квоты per-user (на каждого пользователя)/group (на группу) внутри VPS.

**vzquota** позволяет Вам также формировать внутри VPS квоту per-user (на каждого пользователя)/group (на группу).

vzctl внутренне использует vzquota, чтобы сформировать квоты. И Вам необходима vzquota только, чтобы изучать текущую статистику квот. Синтаксис команды vzquota следующий:

```
vzquota [options] command vpsid [command-options]
```

Основные опции доступные командам vzquota:

-v	Многословный способ. vzquota приходится печатать сообщения о прогрессе наладки поломок в системе. Можете нажать -v дважды, чтобы увеличить количество и качество этих сообщений.
----	--

-q	Немногословный способ. Он подавляет все предупредительные и диагностические сообщения. Отображаются только серьезные, грубые ошибки.
----	--

Квота Open VZ работает в ниже стоящем древе (file system sub-tree) или области системы файлов. Если в этой области уже имеются дополнительные системы файлов, установленные на поддиректориях, квота не последует за этими mount points. Когда Вы инициализируете квоту, Вы также специализируете для нее начало file system sub-tree. Данные о настройках и использовании квоты специально для сервера хранятся в файле `/var/vzquota/quota.vpsid`.

У квоты есть специальный знак для обозначения «загрязненных» файлов. Файл считается таковым, если его контент несовместим с контентом использования реальной квоты. При запуске VPS, квота будет переинициализирована, если Hardware Node был неправильно выключен (например, при отключении электроэнергии, скачке напряжения). Эта операции заметно увеличит время загрузки Node.

Open VZ позволяет устанавливать soft и hard лимиты (и для дискового использования и для inodes использования) так же как и время истечения. По достижению soft лимита Open VZ запускает счетчик времени. Как только время истечет, квота блокирует соответствующее дисковое пространство или inode allocation запросы. Hard лимит не может быть превышен.

**vzquota** понимает следующие команды:

<i>init</i>	До того, как Вы сможете использовать квоту, текущее дисковое пространство и inode должно быть подсчитано использование. Для команды <i>init</i> , необходимо специфицировать все лимиты и дерево файла, где Вы хотите инициализировать квоту.
<i>drop</i>	Забывает о данной квоте ID, опуская существующий файл квоты.
<i>on</i>	Запускает счетчик квоты на определенном квотой ID.
<i>off</i>	Выключает счетчик квоты на определенном квотой ID.
<i>setlimit</i>	Позволяет менять лимиты квот на действующих квотах.
<i>stat</i>	Отображает статистику работающей квоты.
<i>show</i>	Отбражает использование квоты из файла квоты.

### **vzquota init**

Эта команда используется для подсчета текущего пользования дискового пространства и inodes. Она имеет следующий синтаксис:

```
vzquota [options] init vpsid [command-options]
```

Следующие опции принимаются командой *vzquota init*:

<i>-s, --sub-quotas 1 0</i>	По выбору. Если используемая ценность равна 1, тогда per user/group квоты запускаются на сервере VPS. По умолчанию user/group квоты не работают.
-----------------------------	--



<code>-b, —block-softlimit num</code>	Требуется. Disk quota block soft limit – набор 1 Kb блоков, разрешенных для использования сервером. Этот лимит может быть превышен сервером на время, определенное block expiration time (см. ниже). Когда время на исходе, сервер не может больше выделять дисковое пространство даже если hard лимит еще не достигнут.
<code>-B, —block-hardlimit num</code>	Требуется. Определяет disk quota block hard лимит в 1 Kb блоках. Этот лимит не может быть превышен VPS.
<code>-e, —block-exptime time</code>	Требуется. Ограниченное время для достижения inode soft лимита. Время в двух форматах: <ul style="list-style-type: none"> <li>• dd:hh:mm:ss Например: 30 - 30 секунд; 12:00 -12 минут; 20:15:11:00 – 20 дней, 15 часов, 11 минут</li> <li>• xxA, где A - h/H(часы); d/D(дни); w/W(недели); m/M(месяцы); y/Y (годы). Например: 7D - 7 дней; 01 w -1 неделя; 3m — 3 месяца</li> </ul>
<code>-i, —inode-softlimit rium</code>	Требуется. Inodes soft limit – количество inodes, разрешенных для создания для сервера VPS. Этот лимит может быть превышен сервером на время, определенное inode expiration time (см. ниже). Когда время истекло, у VPS больше нет возможности создавать inodes даже если hard лимит еще не достигнут.
<code>-I, —inode-hardlimit num</code>	Требуется. Определяет inodes hard лимит. Этот лимит не может быть превышен сервером.
<code>-n, —inode-exptime time</code>	Требуется. Ограниченное время для достижения inode soft лимита. Время в двух форматах: dd: hh: mm: s s For example: 30 - 30 seconds; 12:00 -12 minutes; 20:15:11:00 - 20 days, 15 hours, 11 minutes <ul style="list-style-type: none"> <li>• xxA, where A - h/H(hour); d/D(day); w/W(week); m/M(month); y/Y (year). For instance: 7D - 7 days; 01 w -1 week; 3m — 3 months</li> </ul>
<code>-p path</code>	Требуется. Определяет направление к частной области VPS.
<code>-c quota_file</code>	По выбору. Определяет файл, чтобы написать вывод о подсчитанном дисковом пространстве и inodes, и вывод о лимитах. Если <code>-c quota_file</code> пропущен, то будет использован файл по умолчанию <code>/var/vzquota/quota.vpsid</code> .

## **vzquota drop**

Перемещает quota файл. Синтаксис этой команды:

```
vzquota [options] drop vpsid [-f] [-c quota_file]
```

Команда проверяет, работает ли для данного VPS квота. Если да, то нет ли ошибки в действиях. Переключатель `-f` может прекратить эти действия и остановить квоту даже если она работает. Вы можете также перекрыть путь к файлу квоты с помощью команды `-c`.

## **vzquota on and vzquota off**

Эти команды используются, чтобы включать и выключать квоту. Они имеют следующий

синтаксис:

```
vzquota [options] on vpsid [command-options]
vzquota [options] off vpsid [-f] [-c quota_file]
```

*vzquota off* выключает долю для дерева системы файла, определенного в файле доли, данном с дополнительный выключателем-*c*. Если этот выключатель пропущен, будет использоваться файл по умолчанию */var/vzquota/quota.vpsid*. Эта команда выведет с ошибкой, если по некоторым причинам к файлу доли нельзя получить доступ и статистика использования была утеряна. Можно повлиять на такое поведение с помощью команды *-f*.

*vzquota on* принимает следующие опции:

<i>-s, --sub-quotas 1 0</i>	По выбору. Если используемая ценность = 1, то per user/group квоты запускаются на сервере VPS. По умолчанию user/group квоты не работают.
<i>-u, --ugid-limit num</i>	По выбору. Определяет максимальное количество пользовательских и групповых ID, для которых на этом сервере будет выдана пользовательская статистика. Ценность по умолчанию = 0.
<i>-p path</i>	Требуется. Определяет путь к частной области VPS.
<i>-f</i>	Эта опция требует пересчета использования квоты, даже если на файле квоты не стоит значка «загрязненный» (dirty).
<i>-c quota_file</i>	По выбору. Определяет файл, чтобы написать вывод о подсчитанном дисковом пространстве и inodes, и вывод о лимитах. Если <i>-c quota_file</i> пропущен, то будет использован файл по умолчанию <i>/var/vzquota/quota.vpsid</i> .
<i>-b, --block-softlimit num</i> <i>-B, --block-hardlimit num</i> <i>-e, --block-exptime time</i> <i>-i, --inode-softlimit num</i> <i>-I, --inode-hardlimit num</i> <i>-n, --inode-exptime time</i>	Эти опции по выбору для команды <i>vzquota on</i> . Они описаны в подсекции <b>vzquota init</b> .

### **vzquota setlimit**

Эта команда обновляет пределы для работающей квоты. Команда требует, чтобы по крайней мере один лимит был определен. Она также обновляет соответствующий файл квоты с помощью новых параметров настройки. Синтаксис этой команды:

```
vzquota [options] setlimit vpsid [command-options]
```

Варианты командных опций:

<i>-u, —ugid-limit num</i>	По выбору. Определяет максимальное количество пользовательских и групповых ID, для которых на этом сервере будет выдана пользовательская статистика. Если ценность = 0, user/group квоты не будут подсчитаны. Ценность по умолчанию + 0.
<i>-b, —block-softlimit num</i> <i>-B, —block-hardlimit num</i> <i>-e, —block-exptime time</i> <i>-i, —inode-softlimit num</i> <i>-I, —inode-hardlimit num</i> <i>-n, —inode-exptime time</i>	Эти опции являются выборочными для команды <i>vzquota on</i> . Тем не менее, по крайней мере одна из этих опций или <i>-u, —ugid-limit num</i> должны быть определены. Они описаны в подсекции <b>vzquota init</b> .
<i>-c quota_file</i>	По выбору. Определяет файл, чтобы написать вывод о подсчитанном дисковом пространстве и inodes, и вывод о лимитах. Если <i>-c quota_file</i> пропущен, то будет использован файл по умолчанию <i>/var/vzquota/quota.vpsid</i> .

### **vzquota stat and vzquota**

Эти команды используются для проверки статистики квоты. Их синтаксис:

```
vzquota [options] show vpsid [-t] [-f] [-c quota_file]
vzquota [options] stat vpsid [-t] [-c quota_file]
```

Различие между этими двумя командами состоит в том, что первая делает отчет об использовании как указано в ядре, в то время как вторая делает отчет соответственно файлу квоты. Тем не менее, *vzquota stat* по умолчанию обновляет файл с последней ядерной статистикой. Если Вы не хотите обновлять файл квоты, добавьте выключатель *-f* к команде.

Вы можете определить альтернативное местоположение к файлу квоты с помощью выключателя *-c quota\_file*. Иначе, будет использован файл по умолчанию */var/vzquota/quota.vpsid*.

Чтобы добавить информацию относительно квоты пользователя/группы к вышеупомянутому выводу команд, используйте в командной строке выключатель *-t*.

Обычный вывод команды *vzquota stat* показан ниже:

```
# vzquota stat 101 -t
resource          usage          softlimit      hardlimit      grace
1k-blocks         113856         2097152        2097152
inodes            42539          200000         220000
User/group quota: on, active
Ugids: loaded 33, total 33, limit 100
Ugid limit was exceeded: no

User/group grace times and flags:
type block_exp_time inode_exp_time hex_flags
user                                     0
group                                    0

User/group objects:
type ID resource usage softlimit hardlimit grace status
user 0 1k-blocks 113672 0 0 loaded
user 0 inodes 42422 0 0 loaded
```

Этот вывод сокращен \ подавлен ради упрощения структуры. Как видно из схемы, VPS 101 имеет те же самые soft и hard лимиты для дискового пространства, и VPS может занять до 2 Гб дискового пространства. Текущее использование - 113 МБ. Серверами используется 42 539 inodes. Soft лимт установлен на цифре 200 000 inodes, hard лимит - 220 000. Пустая grace колонка показывает, что период grace не работает ни для inodes, ни для дискового пространства. Per user/group (на пользователя\группу) квота включена, ею обрабатывается до 100 пользователей и групп. В настоящее время в VPS существует 33 пользователя и группы, отображается также корневая статистика. Сервером не установлены лимиты, и текущее использование для корня составляет 42 422 inodes и 113 Мб дискового пространства.

### Утилиты управления шаблонами

Шаблон (*Template*) - в основном ряд пакетов от некоторого распределения Linux, используемого, чтобы населить VPS. Шаблон ОС состоит из программ системы, библиотек, и скриптов, необходимых для запуска и остановки системы (VPS), основных приложений и утилит. Приложения, также как компилятор и сервер SQL, обычно не включаются в шаблон ОС.

Тайник шаблона (*Template cache*) – это шаблон ОС, установленный на сервере VPS и затем упакованный в gzipped tar архив. Это позволяет ускорить создание нового VPS в несколько раз: вместо того, чтобы устанавливать все пакеты, включающие распределение Linux, vzctl лишь распаковывает архив.

Шаблон metadata (*Template metadata*) - ряд файлов, содержащих информацию, необходимую для воссоздания *Template cache*. Текущий шаблон содержит следующую информацию:

- Список пакетов, которые содержит этот шаблон
- Местоположения (сетевых) репозиторий пакета
- Скрипты, которые должны быть выполнены на различных стадиях установки шаблона
- Общий GPG ключ (и), необходимый для проверки подписей пакетов
- Дополнительные пакеты OpenVZ

**vzpkgls**

Эта утилита содержит списки шаблонов, установленных на Hardware Node или уже установленных на определенный VPS. Утилита имеет следующий синтаксис:

```
vzpkgls [-c|--cached]
vzpkgls vpsid
```

Если Вы определяете ID VPS для этой команды, то *vzpkgls* составляет список шаблонов, относящихся к VPS. Без *vpsid* аргумента, утилита перечислит лишь шаблоны, доступные для VPS на Hardware Node. Другие опции, доступные для *vzpkgls* перечисляются ниже:

-c, <code>--cached</code>	Эта опция не является действенной, если стоит аргумент <i>vpsid</i> . Если же опция используется, чтобы перечислить шаблоны, доступные на Hardware Node, она заставляет <i>vzpkgls</i> пропустить те шаблоны ОС, для которых работающий <i>vzpkgcache</i> еще не создал тайники. Другими словами, если эта опция установлена <i>vzpkgls</i> перечислит лишь готовые для пользования серверами VPS шаблоны.
---------------------------	--

### **vzpkgcache**

Эта утилита создает/обновляет тайники шаблона для шаблонов ОС. Вы должны использовать эту утилиту, прежде чем будете пользоваться только что установленным шаблоном ОС для создания VPS. *vzpkgcache* имеет следующий синтаксис:

```
vzpkgcache [-f|--force] [osname ...]
vzpkgcache -r|--remove osname [...]
```

Данная утилита проверяет metadata для всех шаблонов, установленных на Hardware Node. Если утилита обнаружит шаблон ОС, у которого нет тайника, она начнет загружать и устанавливать все пакеты перечисленные в файле конфигурации и в конце концов создаст тайник. В случае, если тайник уже существует, утилита его обновит, то есть установит все современные обновленные пакеты, которые были выпущены с момента создания или последнего обновления тайника.

Если Вы хотите создать или обновить определенный шаблон(ы) ОС, установите его имя в командной строке.

Обычно Вы управляете *vzpkgcache*, не имея каких-либо вариантов управления. Тем не менее, в этой программе они существуют:

-r, <code>--remove osname [...]</code>	Перемещает тайник для шаблонов, определенных в командной строке ( <i>osname</i> ). Для этой опции необходим точный список шаблонов. Т.е., чтобы переместить тайники нет команды по умолчанию.
--	---

### **vzrpm**

Эта утилита действует как простой RPM wrapper. Её используют для определенных VPS. Она обладает следующим синтаксисом:

```
vzrpm vpsid [rpm_argument ...]
```

Эта утилита управляет *rpm* package manager tool для данного VPS, перекладывая все опции на *rpm*.

### **vzyum**

Эта утилита действует как простой wrapper для утилиты *yum* package manager . Её используют для определенных VPS. Она обладает следующим синтаксисом:

```
vzyum vpsid [yum_argument ...]
```

Эта утилита управляет *yum* package manager tool для данного VPS, имея все необходимые опции. Она также обеспечивает *yum* аргументами, определяющими местоположения хранилищ для распределения, на котором основан данный VPS. Утилита также заставляет *yum* использовать единый тайник - хранилище (single repository cache), располагающийся на Hardware Node.

## ***Дополнительные инструменты***

### **vzdqcheck**

Эта утилита подсчитывает inodes и дисковое, привыкшие использовать тот же алгоритм, как и квота Open VZ. Имеет следующий синтаксис:

```
vzdqcheck [options] path
```

Команда проходит по дереву директории, данному как *path* , и высчитывает пространство, занимаемое всеми файлами, подсчитывает количество inodes. Команда не следует mount пунктам.

Команде *vzdqcheck* доступны следующие команды:

-h	Информация использования
-V	Информация о версии <i>Vzquota</i>
-v	Многословный способ
-q	Немногословный способ

### **vzdqdump and vzdqload**

Утилиты *vzdqdump* и *vzdqload* используются для дампинга (перемещения) лимитов квоты и периода отсрочки из ядра или файла квоты. А также, чтобы загрузить их в файл квоты. *vzdqdump* отображает соответствующие ценности на console экране. *vzdqload* получает информация со стандартного ввода.

Синтаксис команд:

```
vzdqdump [general_options] quota_id [-f] [-c quota_file] -G|-U|-T  
vzdqload [general_options] quota_id [-c quota_file] -G|-U|-T
```

Опции:

-h	Информация использования
-V	Информация о версии <i>Vzquota</i>
-v	Многословный способ
-q	Немногословный способ

Параметр *quota\_id* соответствует ID сервера VPS, для которого Вам необходимо загрузить или переместить (dump).

Другие опции:

<i>-f</i>	Разгружает (перемещает) информацию о квоте пользователя\группы скорее от ядра, чем от файла квоты
<i>-c quota_file</i>	Определяет файл квоты, чтобы обработать файлы, отличные от файлов квоты по умолчанию ( <i>/var/vzquota/quota.vpsid</i> )
<i>-G, --grace</i>	Перемещает или загружает user/group grace times
<i>-U, --limits</i>	Перемещает или загружает user/group disk limits
<i>-T, --exptimes</i>	Перемещает или загружает user/group expiration times

Когда работает утилита *vzdfload*, квоты могут быть выключены. Имейте в виду, что только квоты 2 уровня регулируются утилитами.

### **vzcpucheck**

Утилита отображает текущую утилизацию Hardware Node в отношении установленных единиц CPU и общей способности (вместимости) единиц hardware node CPU. Синтаксис:

```
vzcpucheck [-v]
```

Без аргументов, утилита выводит сумму единиц CPU всех работающих VPS и общую вместимость Hardware Node. Если дается опция *-v*, утилита выводит информацию на каждую Virtual Private Server CPU единицу.

### **vzmemcheck**

Эта утилита отображает параметры памяти Node: low memory утилизацию, low memory commitment (обязательство), RAM утилизацию, memory+swap утилизацию, memory+swap commitment, выделенную для использования память, определенный предел памяти.

Утилита обладает следующим синтаксисом:

```
vzmemcheck [-v] [-A]
```

В командной строке могут быть определены следующие опции:

<i>-v</i>	Отображает информацию для каждого VPS
<i>-A</i>	Отображает абсолютные ценности (в мегабайтах)

Можно использовать 1 опцию, две сразу или вообще обойтись без них.

### **vzcalc**

Эта утилита используется, чтобы вычислить использование ресурса VPS. Синтаксис:

```
vzcalc [-v] vpsid
```

Эта утилита показывает, какую часть ресурсов Node VPS использует *vpsid* . Дополнительный выключатель *-v* делает тщательный (*verbose*) вывод, включая число процессов, *low memory*, выделенную память, а также отображает *memory* и *swap* статистику.

Для приостановленных VPS утилита показывает перспективные (возможные) и максимальные ценности, которые может потребить VPS. Для работающих VPS, отображает текущие ценности.

Высокие ценности использования ресурса означают, что либо Hardware Node переполнен (*overcommitted*) либо конфигурация VPS недействительна.

### **vzpid**

Эта утилита печатает число *id* VPS на данный *id* (*pid*) процесс. Синтаксис:

```
vzpid pid [pid ...]
```

Многочисленные процессуальные *ids* рассматриваются аргументы. Утилита напечатает количество VPS на каждый процесс.

### **vzsplit**

Эта утилита используется, чтобы произвести типовой файл конфигурации VPS с набором контрольных параметров ресурса системы. Синтаксис этой команды следующий:

```
vzsplit [-n num] [-f sample_name] [-s swap_size]
```

Эта утилита используется для деления Hardware Node на равные части. Она вырабатывает полный набор контрольных параметров системы ресурса серверов VPS. Этот набор основан на физической памяти Hardware Node (которую он сам и использует) и на количестве серверов, которыми Hardware Node сможет управлять, если даже они будут поглощать весь ресурс. Утилита сама ( без опций ) подсказывает желательное количество серверов и выводит итоговые контрольные параметры ресурса на экран.

Опции утилиты:

<i>-n num</i>	Рекомендуемое количество серверов, одновременно работающих на Hardware Node.
<i>-f sample name</i>	Имя типовой конфигурации (для создания)
<i>-s swap_size</i>	Размер <i>swap</i> файла на Node. Рекомендуется определить размер <i>swap</i> файла, чтобы принять его во внимание, когда утилита будет создавать типовые конфигурации.

Итоговая типовая конфигурация будет создана в директории */etc/sysconfig/vz-scripts*. Файл будет носить название *ve-sample\_name.conf-sample*. Теперь Вы можете использовать *sample\_name* как



аргумент к опции - *config* команды *vzctl create*. Если образец с таким названием уже существует, утилита пошлет Вам сообщение об ошибке, она не будет заменять существующую конфигурацию.

### **vzcfg validate**

Утилита используется, чтобы проверить последовательность параметров управления ресурса в файле конфигурации VPS. Синтаксис:

```
vzcfgvalidate vps_config_file
```

У утилиты существует множество особенностей, ограничений (constraints), согласно которым она проверяет файл конфигурации. Если constraint не удовлетворена, утилита с severity статусом.

в OpenVZ определено три таких статуса:

<b><i>Recommendation (рекомендация)</i></b>	Это - предложение, которое не является критическим для VPS или Hardware Node. В целом конфигурация действительна; однако, у системы достаточно памяти, лучше согласно рекомендациям увеличить параметры настройки.
<b><i>Warning (предупреждение)</i></b>	Ограничение не удовлетворено, и конфигурация недействительна. Приложения VPS с такой недействительной конфигурацией могут плохо выполнять свою работу или где-то ошибиться.
<b><i>Error (ошибка)</i></b>	Важное ограничение не удовлетворено, и конфигурация недействительна. Приложения VPS с такой недействительной конфигурацией имеют все шансы неожиданно потерпеть неудачу в работе, закрыться или зависнуть.

Предлагается использовать эту утилиту, когда приложения VPS ведут себя неадекватно, и оказывается, что для VPS нет недостатка ресурсов.

## Глоссарий

1. *Hardware Node* (или *Node*) - компьютер, на котором установлена система Open VZ для host серверов VPS.
2. HN - сокращение от Hardware Node.
3. *Host Operating System* (или *Host OS*) - операционная система, установленная на Hardware Node. *Адрес MAC* – это адрес Media Access Control, hardware адрес, идентифицирующий каждый Node в сети. Уровень \ слой MAC напрямую связан с каналами сети (media network). Следовательно, каждому типу канала сети требуется свой слой MAC.
4. *mlock, mlock'ed page* — *mlock()* (сокращенно закрытие памяти) - запрос \ вызов системы. Он делает доступ к особым отделам памяти невозможным. Закрытые этим инструментом страницы (*mlock'ed page*) гарантировано остаются в RAM до того момента пока не будут открыты с помощью *munlock ()* system call. Существуют два приложения memory locking — приложения в режиме реального времени и сверхсекретные сведения. Первому требуется определенная ответная реакция, второму приложению – защитить ценные сведения от попадания в файл подкачки.
5. OS template (или Operating System template) – Шаблон ОС (оперативной системы) используется для создания новых серверов VPS с заранее установленной ОС. См. также Шаблон (Template)
6. *Package set (набор пакетов) = Template (Шаблон)*.
7. *Private area* (Частная область) - часть системы файлов, в которой файлы одного VPS не имеют связи с файлами другого VPS.
8. *siginfo structure* (или просто *siginfo*) – блок информации о выработке сигнала. Если процесс получает сигнал, он может получить также *siginfo*, объясняющее почему система подала такой сигнал. Если процесс контролирует подчиненных, он может получить *siginfo*, объясняющую что случилось с состоянием подчиненного компьютера. В любом случае система возвращает информацию в виде *siginfo\_t*, который включает: номер сигнала, номер ошибки, сигнальный код.
9. *SSH* - сокращение от Secure Shell. Это - протокол для загрузки на отдаленный компьютер и выполнения на нем операций. Он обеспечивает безопасное зашифрованное соединение между двумя небезопасными хост-машинами в пределах сети.
10. *TCP (TCP/IP)* - Transmission Control Protocol/Internet Protocol. Этот набор коммуникационных протоколов, используемый для связи хост-машин с Интернетом.
11. *Template (или package set)* – набор оригинальных пакетов распределения, предназначенных для установки на VPS.
12. *UBC* – сокращение от *User Beancounter*.

13. *User Beancounter* (или *UBC*) - подсистема OpenVZ, занимающаяся управлением памятью VPS и некоторыми системными ресурсами.
14. *VPS* – сокращение от *Virtual Private Server*.
15. *VENET device* – виртуальное сетевое устройство, защищающее VPS от внешнего воздействия.
16. *Virtual Private Server* (или *VPS*) - виртуальный частный сервер, функционально идентичный отдельному серверу с его IP адресом, процессами, файлами, пользовательской базой данных, файлами конфигурации, приложениями, системой библиотек и т.д. Все серверы делят общий *Hardware Node* и общее ядро ОС. Тем не менее, серверы отделены друг от друга . *Virtual Private Server* - что-то вроде песочницы для пользователей и процессов. Нулевой сервер обозначает *Hardware Node*.
17. *OpenVZ* - это полностью автоматизированное и виртуализированное устройство, позволяющее Вам создавать множество отдельных серверов на одном физическом сервере, которые бы делили hardware, лицензии и управление с максимальной долей эффективности.

Перевод : Елена Снурницына