



Шаблоны контейнеров в OpenVZ 7

Денис Силаков

Шаблоны?

Откуда берется ОС в контейнере / VM ?

- “Честно” устанавливаем в пустую VM через VNC, netinstall, ...
 - *2-30 минут*
- Контейнер: установка с хоста (mount, chroot)
 - *1-5 минут*
- Клонировем готовую VM или образ ФС для контейнера
 - *2-30 секунд*

Шаблоны контейнеров Linux в OpenVZ

- Предустановленная ОС
 - Готовый образ ФС, подкладываемый новым контейнерам
vzctl create 1000 --ostemplate centos-7-x86_64
- Установка/удаление/обновление стеком приложений с хост-машины
 - Образ ФС с ОС и приложениями + обертки для пакетных менеджеров
vzpkg install 1000 mysql

Немного истории

- OpenVZ “legacy”
 - Шаблоны ОС – готовые образы ФС в репозиториях
 - **vzpkg** “legacy” (vzrpm, vzyum) - скрипты автоматизации
- Virtuozzo < 7
 - Шаблоны ОС и приложений
 - Образы формируются автоматически на стороне сервера с помощью vzpkg (абсолютно другого!)
 - Этот же vzpkg создавал официальные шаблоны OpenVZ
- OpenVZ / Virtuozzo 7
 - Vzpkg из Virtuozzo 6 – open sourced

Типы шаблонов OpenVZ (“EZ Templates”)

- ОС
 - Базовый
 - centos-7-x86_64
 - Кастомизированные
 - centos-7-x86_64-server
- Приложения
 - centos-7-x86_64-mysql
 - centos-7-x86_64-httpd

Структура шаблона

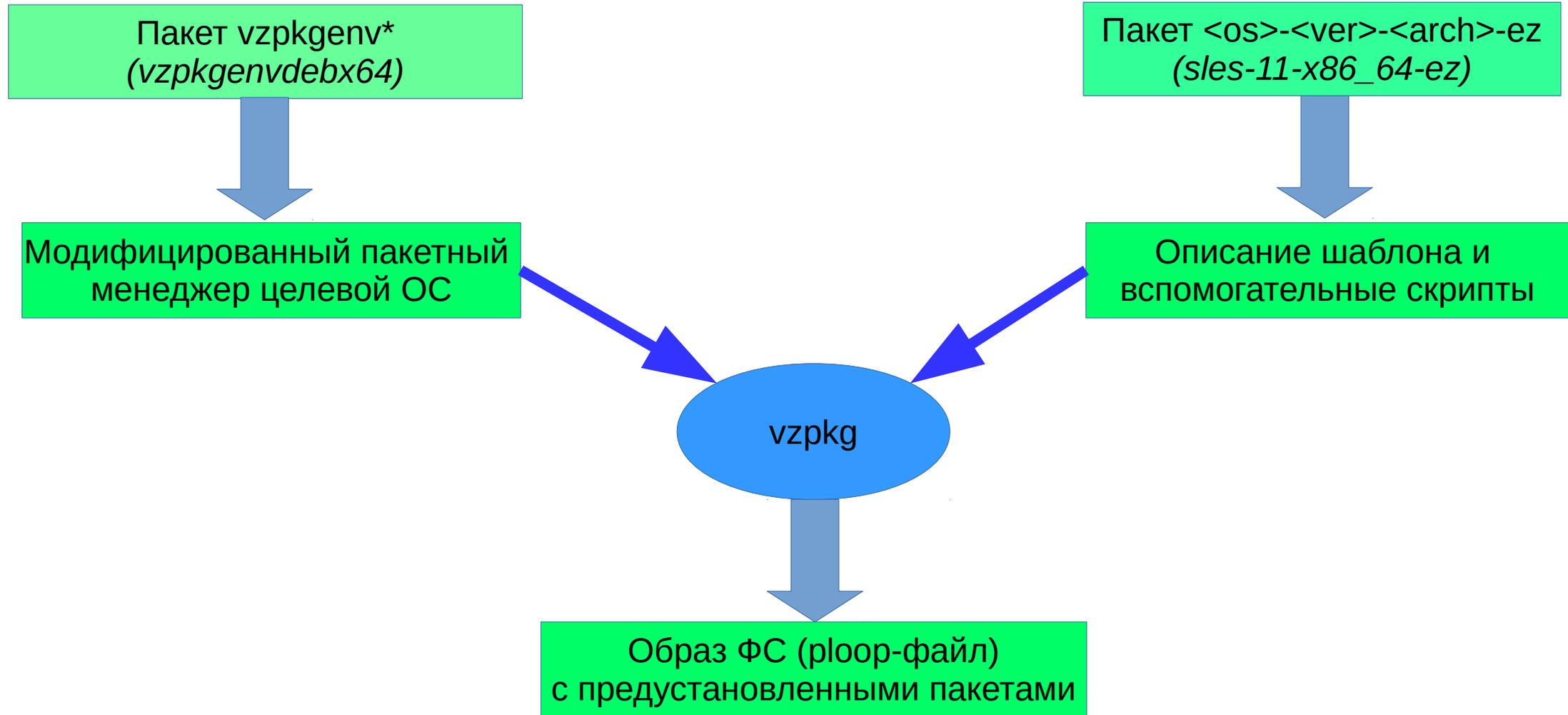
В OpenVZ7, шаблон == набор конфигурационных файлов для vzpkg (не путать с vzpkg из OpenVZ legacy)

- os_distribution
- os_repositories
- os_packages
- os_[pre|post]-[cache|install]
- app1_packages
- ...

Метонахождение в ФС

```
/vz/template/centos/6/x86_64/  
└─ config  
    └─ app  
        ├── cyrus-imap  
        ├── devel  
        ├── jre  
        ├── jsdk  
        ├── mailman  
        ├── mod_perl  
        ├── mod_ssl  
        ├── mysql  
        ├── php  
        ├── postgresql  
        ├── spamassassin  
        ├── tomcat  
        ├── vzftpd  
        └── webalizer  
    └─ os  
        └─ default
```

Создание образа ФС (“кэш” шаблона ОС)



Модификации пакетных менеджеров

Запускаются на хосте, работают с контейнером

- Chroot()
- подмена uid/gid
- вызов команд через vzctl
- игнорирование некоторых ошибок
- ...

обновляются только при радикальных нововведениях в целевых ОС

Поддерживаемые менеджеры

Vzpkg - “обертка” для:

- RPM & Yum / Zypper
- Dpkg / Apt-get

Создание контейнеров

vzpkg - “под капотом”:

```
# vzctl create 1000 --ostemplate centos-7-x86_64
```

```
# prlctl create myCT --vmtype ct \  
--ostemplate centos-7-x86_64
```

(по умолчанию используется DEF_OSTEMPLATE)

Использование vzpkg

- `vzpkg install/update/remove <template>`
(== установка пакетов)
- ... `create/update/remove cache/appcache <template>`
- ... `create/update/remove <ct> <pkg|apptemplate>`
- ... `localinstall/localupdate <ct> <pkg>`
- ... `list [<ct>] [--available]`

Шаблоны приложений

Исторически – аналог “yum groupinstall”

- Установка:
 - развернуть контейнер из кэша ОС с приложением
 - доустановить шаблон в контейнер
- Шаблон установлен == установлены все пакеты из его списка
- При удалении шаблона – удаляются его пакеты, но не очищаются “сироты”

Параметризация

- /etc/vztt/vztt.conf
 - HTTP_PROXY*
 - EXCLUDE (*packages*)
 - ...
- /etc/vztt/url.map
 - Параметры для использования в адресах репозиториев (*например, токены для SLES*)

Преобразование legacy шаблонов

- O vz-template-converter
 - Пытается угадать ОС-версию-архитектуру по имени шаблона
 - Создает минимально необходимые конфигурационные файлы, чтобы vzpkg “видел” шаблон
 - Автоматически вызывается prictl / vzctl при обнаружении legacy-шаблона

Свои шаблоны?

- На основе существующих
- Утилита `vzmktmp1` из пакета `vztt-build`
(см. `/usr/share/vztt/samples`)

Исходный код

- Vzpkg:
<https://src.openvz.org/projects/OVZ/repos/vztt/browse>
- Шаблоны:
<https://src.openvz.org/projects/OVZT>
- Модифицированные пакетные менеджеры:
Ask me:)

dsilakov@virtuozzo.com