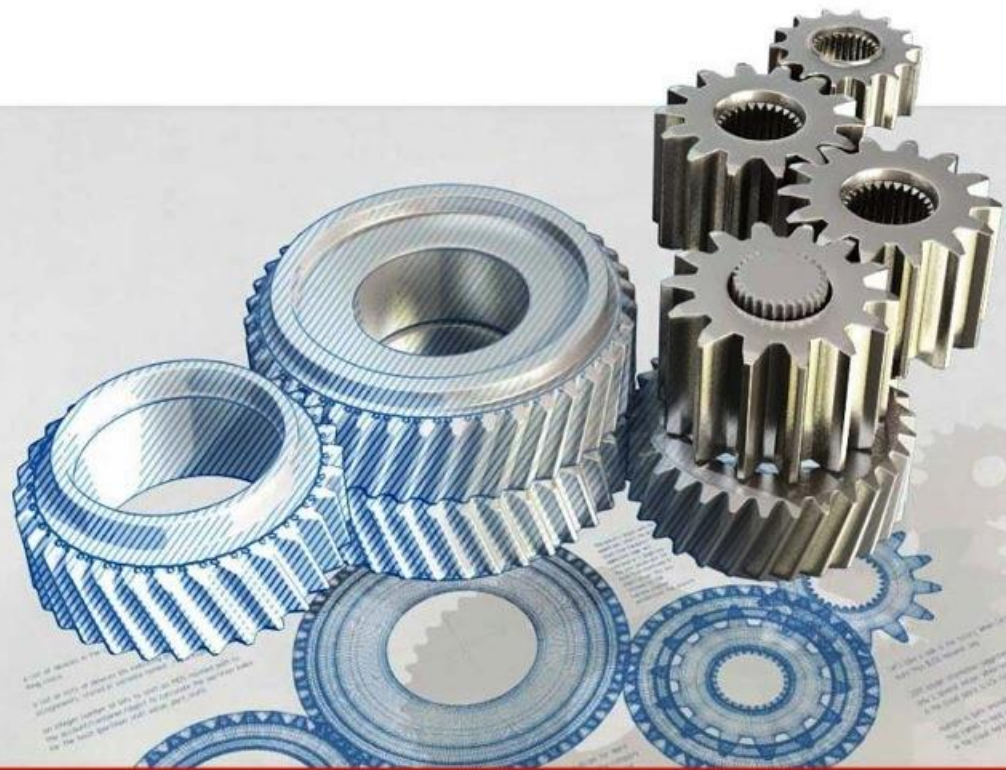




О правильной сборке RPM- пакетов



Matthew Mosesohn
Senior Deployment Engineer

О компании Mirantis



- Лидер в области установки и технической поддержки OpenStack
- Разработала FUEL для автоматизации развертывания OpenStack
- В работе со всеми программными решениями использует принцип Open source

Сборка пакетов? А как же облако?



- OpenStack — кластер систем Linux
- Системы Linux представляют собой компиляцию ПО
- Программы собираются в пакеты
- Пакеты должны быть собраны правильно
- Некачественная сборка приводит к проблемам в работе систем



Что такое программный пакет?



- Пакетом называется специальным образом отформатированный файл,
- содержащий скрипты, правила и файлы,
- для которого определены требования и функции

Getting Started



- Отличный пример:
http://fedoraproject.org/wiki/How_to_create_an_RPM_package
- RPM.org
- Самостоятельная загрузка и доработка SRPM
 - начните с основных поставщиков (Red Hat или CentOS)
 - разработки третьих лиц могут иметь свои особенности/недостатки

Кто собирает программные пакеты?



- Основные поставщики (Red Hat, Ubuntu, SuSE)
- Вторичные поставщики (CentOS, Mint)
- Разработчики
- Сисадмины

Зачем пересобирать пакеты?



- Для установки патчей
- При смене логотипов / другой графики
- При добавлении нового функционала
- Для обеспечения бесшовной сборки

Что делает сборщик пакетов:



- Определяет требования для установки ПО
- Составляет описание программы
- Ведет журнал изменений
- Предварительно компилирует ПО
- Определяет адреса установочных директорий и способ установки ПО
- Добавляет вспомогательные скрипты, чтобы обеспечить успешную установку
- Добавляет скрипты Glue для апгрейда/даунгрейда

Система для сборки пакетов



- Минимальный набор пакетов
- Система должна быть чистой
- Должна быть частью более крупной автоматизированной системы

Чего следует избегать при сборке:



- Доступа к сетям
- Доступа к внешним кодам
- Влияния на другие пакеты и их файлы
- Чрезмерной зависимости от системных сервисов
- Предварительно скомпилированного ПО

Пример sres-файла

- Prep
- Build
- Install
- Pre
- Post



Пример src-файла

Name: python-osnagios

Version: 1.1

Release: 4%{?dist}

Summary: Nagios plugins for OpenStack

Group: System/networking

Source0: %{name}.tar.gz

Requires: python >= 2.6, python-argparse, python-amqplib, libvirt-python, openstack-swift, python-simplejson, python-httplib2, python-setuptools

BuildRequires: python-setuptools

%description

Nagios plugins for OpenStack



Пример sres-файла

`%prep`

`%setup -c`

`%build`

`%{__python} setup.py build`

`%install`

`%{__python} setup.py install -O1 --skip-build --root %{buildroot}`

`%files`

`%defattr(-,root,root,-)`

`%{python_sitearch}/os_nagios-0.2.egg-info`

`%{python_sitearch}/os_nagios/*`

`%changelog`

* Wed Jan 23 2013 Mirantis Product <product@mirantis.com> - 1.1-4

- Initial release

Prep



`%prep`

`%setup -c`

`%patch0 -p1`

- Распаковка исходных файлов
- Использование macro `%setup`
- Применение патчей

Build



```
%{__python} setup.py build
```

```
./configure
```

```
make $BUILD_OPTIONS
```

- Компилирование ПО
- Прописывание опций сборки

Install



```
%{__python} setup.py install -O1 --skip-build --root %{buildroot}
```

```
make install
```

```
/usr/bin/install -d $RPM_BUILD_ROOT %{_datadir}/%{name}
```

```
/usr/bin/install -m 755 $RPM_BUILD_ROOT myapplication %{_datadir}/%{name}/myapplication
```

```
gem install --local --install-dir %{gembuilddir} --force %{SOURCE0}
```

Создание «корневой» папки для установки файлов

- Размещение в ней собранного ПО
- Добавление дополнительных файлов приложений
- Удаление ненужных файлов

- Используется относительно редко
- Все необходимые шаги по подготовке к установке пакета
- Некоторые пакеты создают специальный аккаунт пользователя

- Регистрация разделяемых библиотек (Idconfig)
- Регистрация приложения для запуска
- Модификация sudoers
- Добавление glue-кода для апгрейда (при необходимости)
- Перезагрузить сервисы (при необходимости)

«Нужно» и «нельзя» при сборке



Нужно:



- Обеспечить компактность и логичность секций `%prep` и `%install`
- Добавить комментарии, чтобы другим сборщикам были понятны ваши идеи
- Исходные файлы называть `%{SOURCE0}`, `%{SOURCE1}` и т.д. (для упрощения последующих апдейтов)
- По каждому апдейту составить подробный журнал изменений
- Использовать `%setup -c` для извлечения исходных tar- или zip-файлов
- Использовать `mkdir -p /path` перед установкой файлов в `$RPM_BUILD_ROO`

Нельзя:

- Делать ссылки на другие директории; ваша текущая рабочая папка в `%prep` — новая папка с исходными файлами
- Использовать `'cp'` для перемещения исходных файлов из папки-источника в `$RPM_BUILD_ROOT`.
 - Использовать `install -m 0755 source $RPM_BUILD_ROOT/path/to/target`
- Загружать исходные файлы с URL. Собирайте пакеты в собственной системе в `rpm` с исходной строкой: `Source0: %{name}-%{version}.tar.gz`
- Рассчитывать на сетевое подключение для работы секций `%pre` и `%post` в пользовательской системе
- Пытаться заменить файл, принадлежащий другому RPM

Бэцпортинг



Применяется для:

- Устранения багов
- Добавления новых функций
- Для адаптации пакетов к работе в системах со старыми версиями библиотек
- Интеграции новых, не полностью совместимых с системой пакетов

Пример бэкпортинга

- Программа: dnsmasq версия 2.64
- Сервер DHCP и DNS
- Полезна для управления сетями небольшого масштаба
- Необходимо было обновить программу до 2.64 для устранения бага unicast DHCP для различных сетевых интерфейсов
- Последняя версия в CentOS/RHEL — 2.48

Процедура бэкпортинга

- Загрузите исходный RPM-файл
- `rpm -i dnsmasq-2.64-1.fc18.src.rpm`
- Отредактируйте `dnsmasq.spec`
- Добавьте журнал изменений
- Запустите ``rpmbuild -ba dnsmasq.spec``
- Установите созданный RPM
- При необходимости проведите отладку

Проблемы и недостатки бэкапирования



- Отсутствие поддержки SystemD в CentOS
- Необходимость конвертирования сервисных скриптов в SysV
- При необходимости придется использовать старые копии
- Для исходного скрипта добавляется новый исходный файл
- Исходный скрипт добавляется в секцию %install
- По возможности закомментировать строки

Спасибо за внимание! Есть вопросы?
