



Онлайн образование

otus.ru

• REC Проверить, идет ли запись

Меня хорошо видно && слышно?



Тема вебинара

Управление пакетами. Дистрибьюция софта



Арифулов Ренат

e-mail: arifulovrenat@gmail.com



Преподаватель



Ренат Арифулов

Около 10 лет в IT индустрии

Ранее занимался администрированием
вычислительного центра научной лаборатории
Затем администрировал рендер-ферму и
Инфраструктуру студии компьютерной графики

Системный администратор linux UCS

Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в Slack
#webinars-2022-04
или #general



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или
задайте вопрос

Маршрут вебинара

Знакомство

Make

RPM

Практика. Работа с RPM

YUM

Практика. Работа с YUM

Пакетные менеджеры других систем

Практика. Работа с SNAP

Docker

Практика. Работа с Docker

Домашнее задание

Рефлексия

Цели вебинара

После занятия вы сможете

1. Познакомиться с работой пакетных менеджеров
2. Изучить создание пакетов RPM
3. Познакомиться с другими способами установки ПО (snap,docker)

Смысл

Зачем вам это уметь

Ваше мнение?

Смысл

Зачем вам это уметь

1. Понять как работают пакетные менеджеры
2. Узнать как производится дистрибьюция софта
3. Научиться создавать пакеты RPM

Make

Make. История

Инструмент контролирующий процесс сборки ПО из исходных файлов. Разработан в 1976 году в Bell Labs.

Версии:

- **BSD** make, основанная на работе Адама де Бура (*Adam de Boor*) над версией make, с возможностью параллельной сборки; в той или иной форме перешла в **FreeBSD**, **NetBSD** и **OpenBSD**.
- **GNU** make — входит в большинство дистрибутивов **GNU/Linux** и часто используется в сочетании с **GNU build system**.

<https://www.gnu.org/software/make/manual/make.html>

Make.

Самый простой метод, который описывается командой

```
./configure && make && make install
```

Суть метода состоит просто в сборке и инсталляции необходимого ПО куда-то в операционную систему

Makefile.

Make ожидает инструкции из текстового файла Makefile. В 99% случаев вы найдете этот файл рядом с исходниками программы.

`make [-f Makefile][цель]`

Стандартные цели для сборки:

- `all` - выполнить сборку пакета
- `install` - установить пакет из дистрибутива (производит копирование исполняемых файлов, библиотек и документации в системные каталоги)
- `uninstal` - удалить пакет
- `clean` - очистить дистрибутив (удалить из дистрибутива объектные и исполняемые файлы, созданные в процессе компиляции)
- `distclean` - очистить все созданные при компиляции файлы и все вспомогательные файлы, созданные утилитой

`./configure --help` - помощь в предварительной конфигурации сборки

Makefile. Синтаксис

```
all: program
```

```
program.o: program.c
```

```
    gcc -c program.c -o program.o
```

```
program: program.o
```

```
    gcc program.o -o program
```

```
clean:
```

```
    -rm -f program.o
```

```
    -rm -f program
```

Makefile. + и -

- Возможность использовать последнюю или, наоборот, устаревшую версию ПО, которой нет в репозиториях
- Возможность собрать ПО с теми модулями/опциями, которые необходимы именно в вашем случае и для ваших целей
- Отсутствие необходимости разбираться с созданием пакета и делать свои сложные “обертки” для сборки

Makefile. + и -

- Возможность использовать последнюю или, наоборот, устаревшую версию ПО, которой нет в репозиториях
- Возможность собрать ПО с теми модулями/опциями, которые необходимы именно в вашем случае и для ваших целей
- Отсутствие необходимости разбираться с созданием пакета и делать свои сложные “обертки” для сборки
- Требует наличия сборочного окружения на сервере
 - ◆ В случае компроментации сервера - облегчает задачу по сборке эксплоита злоумышленнику
 - ◆ Это окружение не является необходимым для выполнения приложения, оно нужно только для подготовки окружения
- Занимает много времени
- Ручное разрешение зависимостей
- Сложность тиражирования
- Возможны затруднения с unit и init-скриптами
- Больше затрат на поддержание системы в целом за счет необходимости самостоятельно следить за обновлениями безопасности

Make.

Самая ключевая проблема этого способа установки ПО заключается в стремлении к хаосу.

Разбор этого и холивар есть в статье:<https://otus.pw/ulZn/>

<https://otus.pw/ulZn/>



RPM

RPM. Обзор

- RedHat Package Manager - написан на C+Perl, первая версия в 1997 году
- В ОС устанавливается уже собранное ПО, поддерживаемое командой поддержки репозитория/дистрибутива.
- RPM-пакет можно рассматривать как умный zip-архив
- Важно понимать, что при установке не происходит и не должно происходить сборки/компиляции ПО

RPM. + и -

- Низкие трудозатраты на установку ПО
 - ПО поддерживает maintainer`ами репозитория/дистрибутива
 - Большинство пакетов разбито на binary/dev-пакеты, нет мусора в ОС
 - Для установки ПО не нужно окружение для сборки, меньше софта в ОС
 - Возможность автоматически выполнять скрипты при установке/удалении
- Замороженные версии софта, не всегда последние
 - Ручное разрешение зависимостей

RPM.

Пакеты представляют из себя архив определенного формата, в нем содержится:

- Метаинформация
 - Имя
 - Версия
 - Релиз
 - Архитектура
 - Зависимости
 - Ресурсы
 - ...
- Файлы
- Скриптлеты
 - pre-install
 - post-install
 - pre-remove
 - post-remove
 - ...

RPM: управление

- `rpm -q {name}` — проверить установлен ли пакет {name}
- `rpm -qi {name}/{pkg}` — показать мета-информацию о пакете
- `rpm -qp --queryformat %{VERSION}-%{RELEASE} {pkg}` — формат вывода информации
- `rpm -e {name}` — удалить пакет
- `rpm -i {pkg}` — установить пакет
- `rpm -ql {name}/{pkg}` — вывести список файлов пакета
- `rpm -q --scripts` — показать скриптлеты
- `rpm -qR` — показать от каких пакетов зависит этот
- `rpm -q --provides {name}` — показать ресурсы предоставляемые пакетом
- `rpm -qf {file}` — показать какому пакету принадлежит {file}.
- `rpm2cpio {pkg} | cpio -idmv` — распаковать содержимое пакета

RPM. Верификация

- 5 - контрольная сумма MD5
- S - размер
- L - символьная ссылка
- T - дата изменения файла
- D - устройство
- U - пользователь
- G - группа
- M - режим (включая разрешения и тип файла)
- ? - файл не удалось прочитать

Создание RPM

RPM. Сборка

```
sudo yum install rpmdevtools rpm-build
```

```
rpmdev-setuptree
```

```
tree -d -L 1 ~/rpmbuild
```

```
~/rpmbuild
```

- |— BUILD # директория в которой происходит сборка
- |— RPMS # директория с собранными пакетами
- |— SOURCES # директория с исходными файлами
- |— SPECS # директория с спес-файлами
- |— SRPMS # директория с SRPM-пакетами

RPM. Spec-file

Это описание ПО вместе с инструкциями как построить пакет и списком файлов для всех устанавливаемых файлов.

имя пакета - тире - номер версии - тире - номер выпуска (релиза) - точка - spec

vim-8.1.2.spec

Заголовок

Summary: Это однострочное описание пакета.

Name: Это должна быть строка имени из имени файла rpm, которое вы планируете использовать.

Version: Это должна быть строка версии из имени файла rpm, которое вы планируете использовать.

Release: Это номер выпуска для пакета с той же самой версией (например, если мы сделали пакет и обнаружили, что он незначительно неисправный и нам необходимо сделать его заново, то следующий пакет будет номер выпуска 2).

Icon: можно установить иконку по умолчанию

Source: указание на файл с исходниками. Можно указывать несколько файлов через Source0, Source1, etc.

Path: указание на файлы с патчами

RPM. Spec-file

Раздел Prep

Здесь располагаются команды для подготовки пакета к команде make (исправления в исходниках, установка доп. пакетов). По сути сюда можно вставить скрипт и он все сделает сам. НО! Есть встроенные макросы для удобства:

%setup - распаковывает исходные тексты и делает cd к ним

%patch - накладывает патчи автоматически

Раздел Build

Раздел для построения ПО. Макросов нет, но так же можно просто запустить скрипт для сбоя

Раздел Install

Раздел для установки ПО в операционную систему. Если есть make install, то можно подставить сюда

- %pre макрос для выполнения предустановочного скрипта.
- %post макрос для выполнения послеустановочного скрипта.
- %preun макрос для выполнения скрипта перед удалением пакета.
- %postun макрос для скрипта выполняемого после удаления пакета.

RPM. Spec-file

Раздел File

Это раздел где вы обязаны перечислить все файлы для двоичного пакета. У RPM нет механизма самостоятельно узнать какие файлы были установлены с помощью `make install`.

`%doc` - обозначение для документации. Будут установлены в директорию `/usr/doc/$NAME-$VERSION-$RELEASE`

`$config` - обозначение для конфигурационных файлов в пакете.

`$dir` - директория, которой “владеет” пакет.

`%files -f <filename>` - можно взять список файлов из заранее подготовленного файла, а не перечислить руками.

RPM. Сборка

`rpmbuild -bb otus.spec --` сборка RPM

`rpmbuild -bs otus.spec --` сборка SRPM

`rpmbuild -ba otus.spec --` сборка RPM+SRPM

RPM. Hint

```
sudo yum install -y yum-utils  
yumdownloader --source <package>  
sudo yum-builddep <package|specfile>
```

RPM. Mock

Позволяет пересобрать пакет под другую версию ОС

- `sudo yum install mock mock-rpmsfusion-free`
- `sudo usermod -a -G mock $(whoami)`
- `newgrp mock`
- `spectool -g -R clogtail.spec`
- `rpmbuild -bs clogtail.spec`
- `ls /etc/mock`
- `mock -r epel-7-x86_64 --rebuild`
`~/rpmbuild/SRPMS/clogtail-0.3.0-2.fc28.src.rpm`

Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет

Практика. Работа с RPM

Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет

YUM

YUM. Обзор

Yum является программой управления и автоматизации работы с пакетным менеджером rpm (Redhat Packet Manager).

Сам по себе rpm способен только проинсталлировать пакет в систему, проверив установлены ли пакеты-зависимости.

Yum позволяет реализовать скачивание пакета, автоматическое разрешение зависимостей, обновления, добавление репозитория.

YUM. Конфигурация

Хранится в /etc/yum.conf

Конфиг в .ini формате:

[main]

cachedir=/var/cache/yum/\$basearch/\$releasever

keepcache=0

debuglevel=2

logfile=/var/log/yum.log

exactarch=1

obsoletes=1

gpgcheck=1

plugins=1

installonly_limit=5

bugtracker_url=http://bugs.centos.org/set_project.php?project_id=23&ref=http://bugs.centos.org/bug_report_page.php?category=yum

distroverpkg=centos-release

Yum: управление

- `search` — поиск пакета
- `install` — установка пакета(ов)
- `update` — обновление (до версии)
- `downgrade` — откат до до версии
- `check-update` — проверка обновлений
- `remove` — удаление пакета
- `info` — информация о пакете
- `provides` — найти из какого пакета файл
- `shell` — CLI

YUM. Конфигурация репо

Хранится в /etc/yum.repos.d/

Представляет собой конфиг в .ini формате:

```
[base]
```

```
name=CentOS-$releasever - Base
```

```
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&re  
po=os&infra=$infra
```

```
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/  
gpgcheck=1
```

```
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

```
enable=1
```

Yum: свой репозиторий

- `sudo yum install createrepo`
- `sudo mkdir -p /repos/CentOS/7/`
- `sudo createrepo /repos/CentOS/7/`
- `rsync -avz rsync://mirror.truenetwork.ru/centos/7.5.1804/
/repos/CentOS/7/`
- `[local-base]`
- `name=Local-Base`
- `baseurl=file:///repos/CentOS/7/$basearch/`
- `enabled=0`

Практика. Работа с YUM

Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет

Пакетные менеджеры других систем

Пакетные менеджеры

- apt – используется в deb-based дистрибутивах
- Pacman (Arch Linux, Madjango, EndeavourOS и др.) - написана на C#, в качестве пакетов использует архивы pkg.tar.gz
- Portage (Gentoo, Calculate Linux) - пакетов нет, все компилируется из исходного кода, можно гибко настроить и полностью контролировать процесс
- ZYPPER (OpenSUSE, SUSE Linux) - написан на C, значительно быстрее чем yum, работает с rpm-пакетами
- SNAP - универсальный пакетный менеджер - рассмотрим далее
- FLATPACK (Fedora) - конкурент snap, более открыт нежели SNAP
- APK (Alpine) - используется в минималистичной версии Linux (часто применяется в Docker)
- DNF - приходит на смену yum - рассмотрим далее

SNAP

Пакетная система создана компанией Canonical, первоначально для Ubuntu.

Основная особенность: в пакет с приложением входит полный набор зависимостей, необходимый для запуска данного приложения.

В данный момент SNAP применяется не только в Ubuntu, но и в Debian, openSUSE, Arch Linux, Gentoo, Fedora

Практика. Работа с SNAP

DNF

Dandified YUM - следующее поколение приложения yum

Решает такие проблемы yum как:

- низкая производительность
- высокое потребление памяти
- низкая скорость итеративного разрешения зависимостей

Использует внешнюю библиотеку libsoft для решения зависимостей

Используется в Fedora (с 22 версии на постоянной основе), RHEL 8, CentOS 8, OEL 8, Mageia 6/7

Вопросы?



Ставим "+",
если вопросы есть

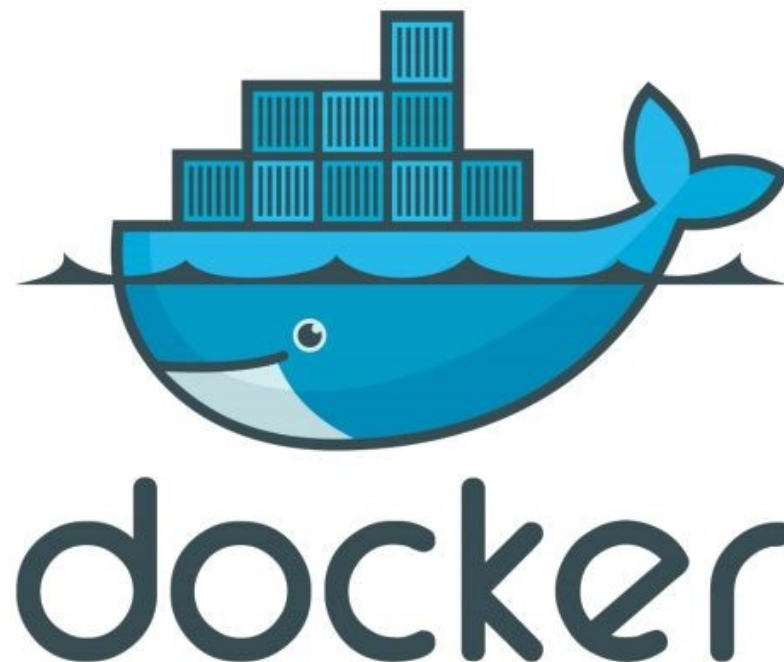


Ставим "-",
если вопросов нет

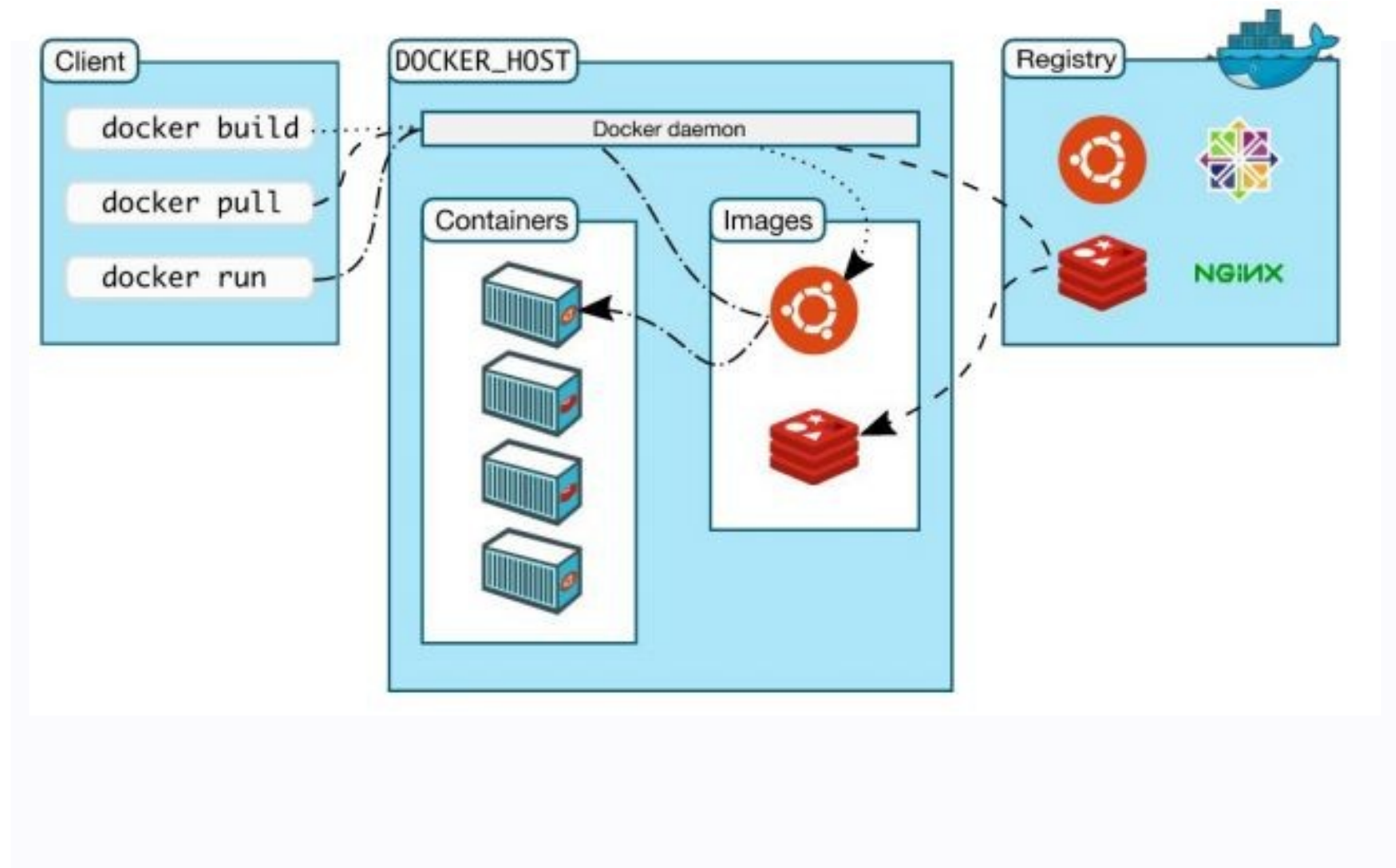
Docker

Docker

- Это про стандартизацию, иммутабельность и воспроизводимость
- Это не виртуальная машина. Это приложение и его окружение упакованные в изолированное окружение
- Есть возможность ограничить ресурсы контейнера



Docker



Docker

```
docker run \  
  --name nginx \  
  -v /srv/nginx/:/usr/share/nginx/html \  
  -p 80:80 \  
  -d nginx
```

Практика. Работа с Docker

Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет

Домашнее задание

Yum: свой репозиторий

- 1) Создать свой RPM пакет (можно взять свое приложение, либо собрать, например, апач с определенными опциями)
- 2) Создать свой репозиторий и разместить там ранее собранный RPM

Реализовать это все либо в Vagrant, либо развернуть у себя через nginx и дать ссылку на репозиторий.

* Написать свой Dockerfile, собрать Image и разместить его в Docker Registry. Как результат прислать ссылку на Image и краткую инструкцию.

Рефлексия

Цели вебинара

Проверка достижения целей

1. Познакомиться с работой пакетных менеджеров
2. Изучить создание пакетов RPM
3. Познакомиться с другими способами установки ПО (snap,docker)

Заполните, пожалуйста,
опрос о занятии
по ссылке в чате

Спасибо за внимание!

Приходите на следующие вебинары



Арифулов Ренат Надирович

Системный администратор linux

e-mail: arifulovrenat@gmail.com

