

Uni Web Topics Presentation

Felix Pojtinger

October 28, 2021

Introduction

Contributing

License

Overview

Development

Distribution

Basic Distribution Principles

Packaging Overview

Distribution to RedHat Enterprise Linux

Distribution to Debian GNU/Linux

Distribution to Linux (universal)

In Comparison: Distribution to Android, Windows and macOS

Distribution to Kubernetes/the Cloud

Distribution to WebAssembly

Operation

Introduction

Contributing

Contributing

These study materials are heavily based on [professor Heuzeroth's "Spezielle Themen für Web-Anwendungen"](#) lecture at HdM Stuttgart.

Found an error or have a suggestion? Please open an issue on GitHub (github.com/poijntfx/uni-webtopics-notes):



Figure 1: QR code to source repository

If you like the study materials, a GitHub star is always appreciated

License

License



Figure 2: AGPL-3.0 license badge

Uni Web Topics Presentation (c) 2021 Felix Pojtinger and contributors

SPDX-License-Identifier: AGPL-3.0

Overview

Overview

- ▶ What is DevOps?
- ▶ Which parts of the software lifecycle does it cover?
 - ▶ Development
 - ▶ Distribution (I will focus on this today)
 - ▶ Operation
- ▶ What is “cloud native”?
- ▶ Why are “traditional” distribution methods still relevant?

Development

Development

- ▶ DevOps: Also includes development!
- ▶ Modern development should not be bound to any client attributes
- ▶ It should not matter if the client is a RISC-V Linux machine, a locked-down Windows workstation or an Android phone
- ▶ Development should be possible from any platform, for any platform
- ▶ The only truly cross-platform application framework is the web
- ▶ PWAs make it possible for web apps to have all the features native apps have
- ▶ PWAs work offline by default
- ▶ Why not make our development environments PWAs?
- ▶ Virtual machines and user-friendly hypervisors and containers make it possible to run the editor's backend locally too

Distribution

Basic Distribution Principles

Basic Distribution Principles

► Binaries

- Compiled forms of software
- On Linux: ELF binaries, PE binaries on Windows and MACH-O binaries on macOS
- Binaries can be statically or dynamically linked
 - Statically linked: Since the Linux ABIs are stable, one can depend on them not changing - this allows not linking against any specific C library and makes the resulting binary portable across distributions. It also allows including all external dependencies into the binary, effectively making it a “single-file” distribution method
 - Dynamically linked: Thanks to `dlopen` and package management, dynamic linking can also be used. Most of the time (especially on non-Linux OSes), at least the C library and external dependencies (i.e. SQLite) thus need to be available in `LD_LIBRARY_PATH` at runtime; if they are not, the application can't continue. This makes the binaries non-portable across distributions; for example, if a binary is built on a Debian 11 host, it most probably won't run on a Debian 10 host due to the different versions of the GNU C library used. This does however also have a few big

Packaging Overview

Packaging Overview

- ▶ What is a package?
 - ▶ Includes the binary, assets, metadata and signature
 - ▶ Is self-describing
 - ▶ Mostly some form of archive (i.e. RPM, .tar.gz) in combination with a metadata file and signature
- ▶ What is a package manager?
 - ▶ Can install, remove and update packages
 - ▶ Mostly two components: Low-level tool to install and remove package files (dpkg on Debian, rpm on Fedora) and a high-level tool to search, download, install and resolve dependencies (apt on Debian, dnf on Fedora)
 - ▶ Can resolve and install runtime and build-time dependencies (i.e. dependency on C library, SQLite, SDL2, headers for cURL etc.)
 - ▶ Can check GPG signatures of
- ▶ Repository
 - ▶ Can serve packages and their metadata (i.e. versions)
 - ▶ Low-level repository is the provided base distribution (“base”)

Distribution to RedHat Enterprise Linux

Distribution to RedHat Enterprise Linux

- ▶ RHEL is a very popular distribution and serves as the upstream of many other distros (CentOS, Rocky Linux etc.)
- ▶ Fedora Linux is its upstream
- ▶ Is based on the RPM package format and the DNF package manager
- ▶ Commercial
- ▶ Very long support cycles (at least ten years per major release)
- ▶ RPM package format: Demo
- ▶ DNF package manager and repositories: Demo

Distribution to Debian GNU/Linux

Distribution to Debian GNU/Linux

- ▶ Debian is another very popular distribution that also serves as the upstream of many other distros (Ubuntu, Linux Mint, Pop!_OS etc.)
- ▶ Is based on the DEB package format and the APT package manager
- ▶ Community-Driven, completely Free Software
- ▶ 5 years support per major release
- ▶ DEB package format: Demo
- ▶ APT package manager and repositories: Demo

Distribution to Linux (universal)

Distribution to Linux (universal)

- ▶ Flatpak overview
- ▶ Flatpak's build system and differences to traditional packages
- ▶ Flatpak repository

In Comparison: Distribution to Android, Windows and
macOS

In Comparison: Distribution to Android, Windows and macOS

- ▶ APKs in F-Droid repository
- ▶ MSI package with auto-updates
- ▶ DMG package with auto-updates

Distribution to Kubernetes/the Cloud

Distribution to Kubernetes/the Cloud

- ▶ Docker
- ▶ Kubernetes
- ▶ Helm
- ▶ Skaffold

Distribution to WebAssembly

Distribution to WebAssembly

- ▶ WASM-Binary
- ▶ WASI/wasm_exec equivalents

Operation

Operation

- ▶ Sentry
- ▶ OpenTelemetry
- ▶ Prometheus
- ▶ Grafana

Pipelines

Pipelines

- ▶ Bagop
- ▶ Hydrun
- ▶ GitHub Actions
- ▶ Semantic Release