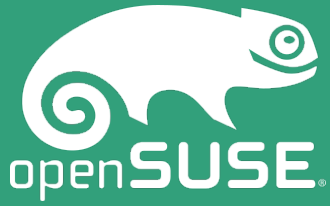


Resurrecting dinosaurs, what can possibly go wrong?

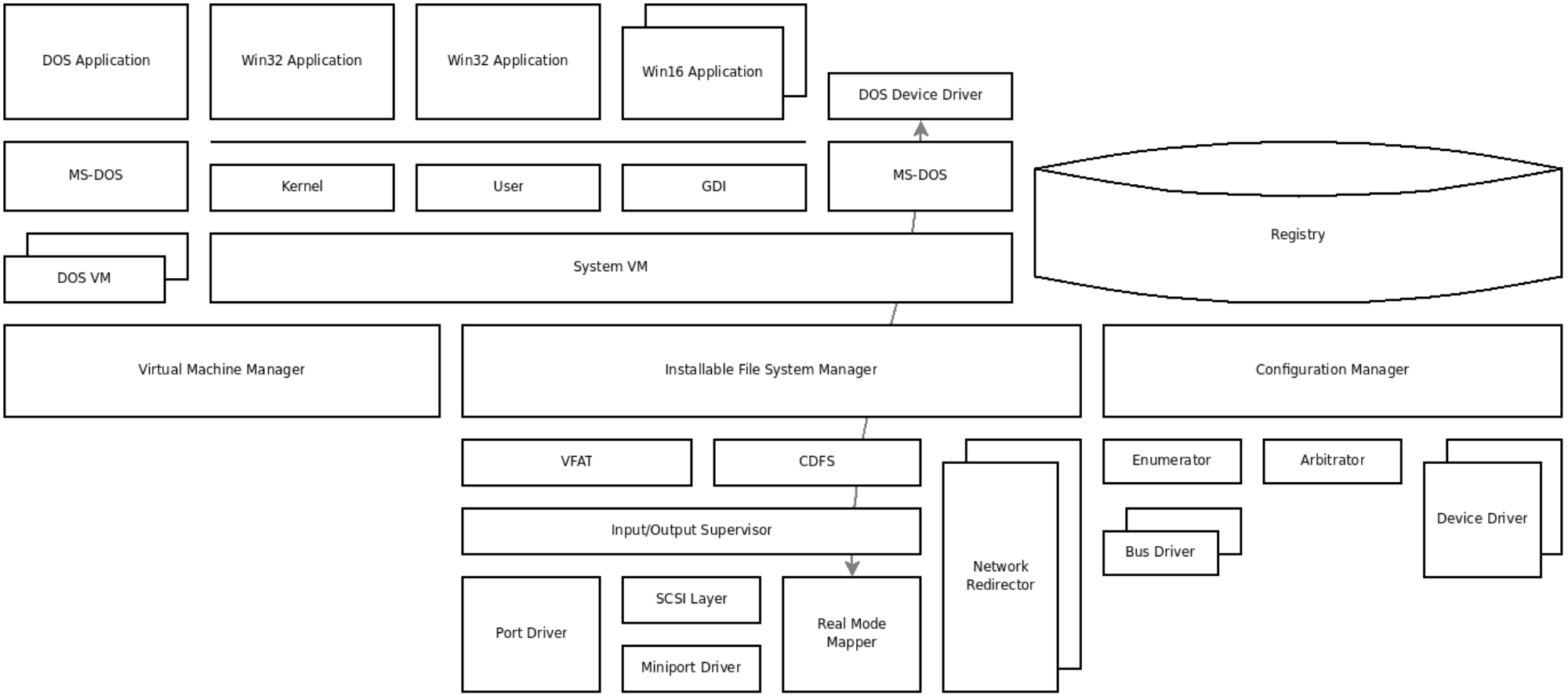
How Containerised Apps could eat our users.

Richard Brown
openSUSE Chairman
rbrown@opensuse.org

“Those who cannot remember the past are condemned to repeat it”

- George Santayana

In the beginning



Windows 3.1/95 - DLL Hell

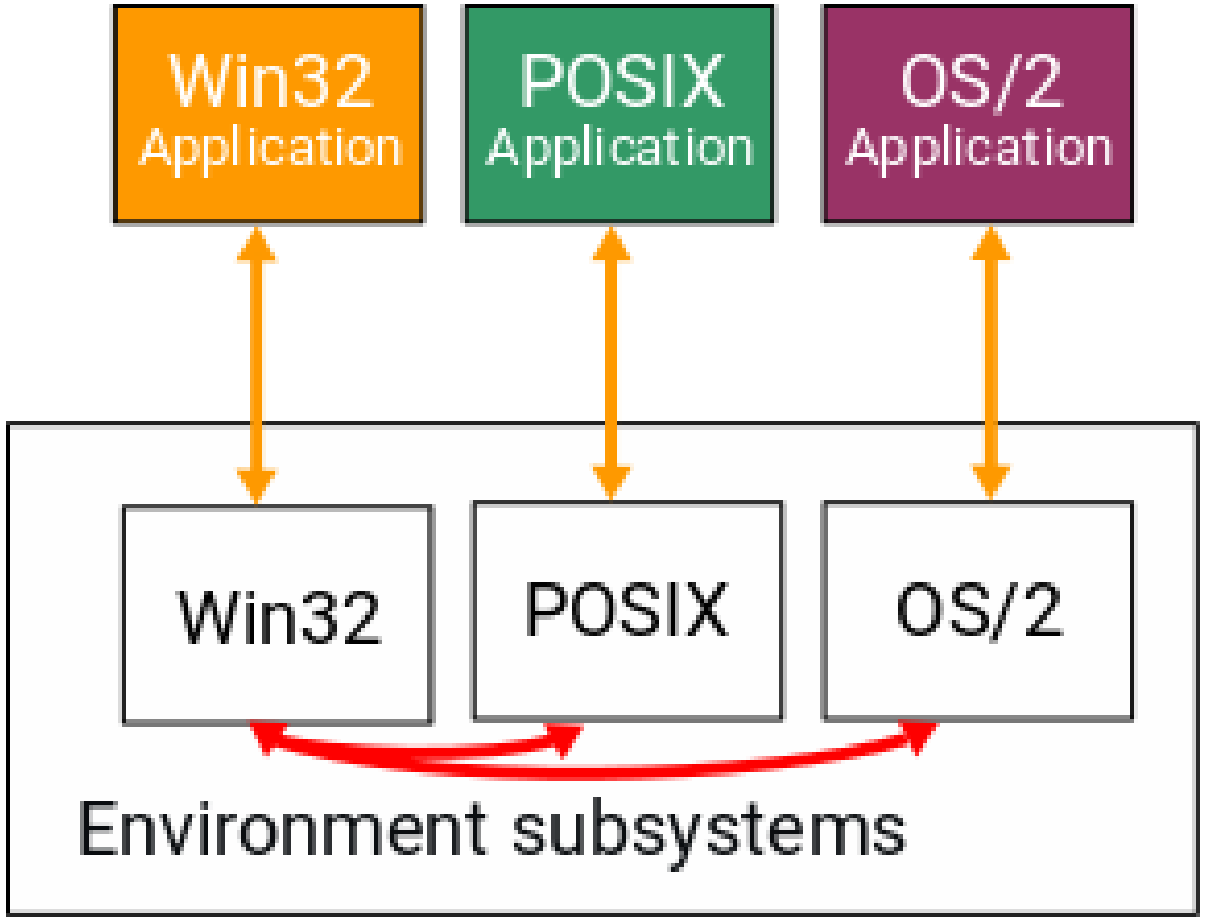
- No ABI backwards compatibility
- Most DLLs installed in C:\WINDOWS or C:\WINDOWS\SYSTEM
- Global COM Class IDs
- Service/Maintenance Nightmare

DLL Hell in Real Terms

- Developers had to dev & test Apps on every possible DLL combination
- Then retest every App patch on every possible DLL combination
- AND test every DLL patch on every possible App & DLL combination
- Then cry when it all broke anyway

Windows 2000 to the Rescue

- Side-by-side (SxS) assembly – DLL “Containerisation”
  - Separate Memory Space for each App and its DLLs
  - ‘Private DLLs’ loaded from the Application Directory
- Windows File Protection (WFP) – Disk Isolation of System DLLs
- DLL Universal Problem Solver (DUPS) – Audit all the DLLs in use and help migrate ‘legacy’ applications to SxS bundles



Problem Solved? Right?

- Security nightmare
  - Security relevant DLLs lurking in countless application folders
- Maintenance nightmare
  - How are we going to update our app? Oh we'll ship an updater!
- Legal nightmare
  - Can we legally redistribute all the DLLs we need to?
- Storage vendor dream
  - More disk consumption, everyone buying bigger disks!

Meanwhile in Linuxland

Distributions – Solving Real Problems

- Security
  - Security Teams auditing packages, monitoring CVEs & embargoed lists
- Maintenance
  - Maintainers packaging applications & keeping them updated
- Legal
  - Lawyers auditing licenses and ensuring compatibility/compliance

In Defence of Shared Libraries/Dependencies

- Not just about using less space on disk
- Distributing fewer libraries have broad benefits
  - Fewer INSECURE libraries, more easily patched
  - Less manpower required to maintain/update
  - Easier to review/ensure legal compliance

Mission Accomplished?

- Compatibility
- Portability
- Pace of Change vs “It just works”

Windows 3.1/95 - DLL Hell

- No ABI backwards compatibility
- Most DLLs installed in C:\WINDOWS or C:\WINDOWS\SYSTEM
- Global COM Class IDs
- Service/Maintenance Nightmare

Compatibility

- Many distributions with many different libraries and apps
- Different apps require different libraries
- Application developers don’t want to worry about what other application developers have chosen as their dependencies

Compatibility

- Many distributions with many different libraries and apps
- Different apps require different libraries
- Application developers don’t want to worry about what other application developers have chosen as their dependencies

- But application developers don’t (often) worry about this
- Distro Maintainers work on this for F/OSS licensed apps

Portability

- Many distributions with many different libraries and toolsets
- Application Developers don’t want to learn dozens of toolsets, nor rebuild & retest their application on a dozen platforms

Portability

- Many distributions with many different libraries and toolsets
- Application Developers don’t want to learn dozens of toolsets, nor rebuild & retest their application on a dozen platforms

- But application developers don’t (often) worry about this
- Distro Maintainers solve the problem for F/OSS licensed apps

Pace of Change vs “It just works”

- Many distributions with fixed release schedules
- Distributions freeze package/library versions to aid ‘stability’
- Holds back new application versions from users

Pace of Change vs “It just works”

- Many distributions with fixed release schedules
- Distributions freeze package/library versions to aid ‘stability’
- Holds back new application versions from users

- But application developers don’t need to worry about this
- Rolling Distributions resolve this with increasing efficiency

Back to the Future!

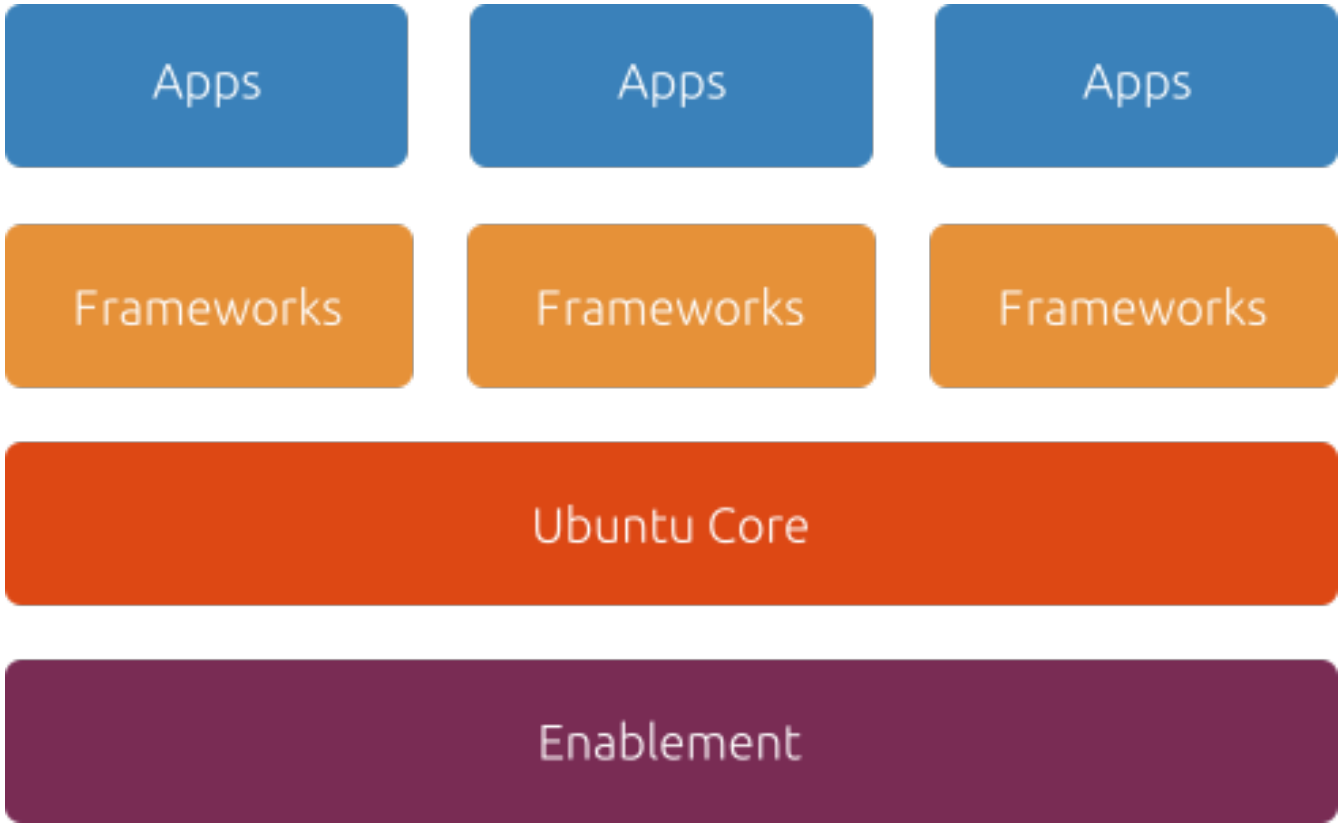
Containerised Applications to the Rescue

- AppImage, FlatPak, Snappy
- Provides uses with a “Bundle” containing App + Libraries
- Runs the App in some kind of Sandbox or Container

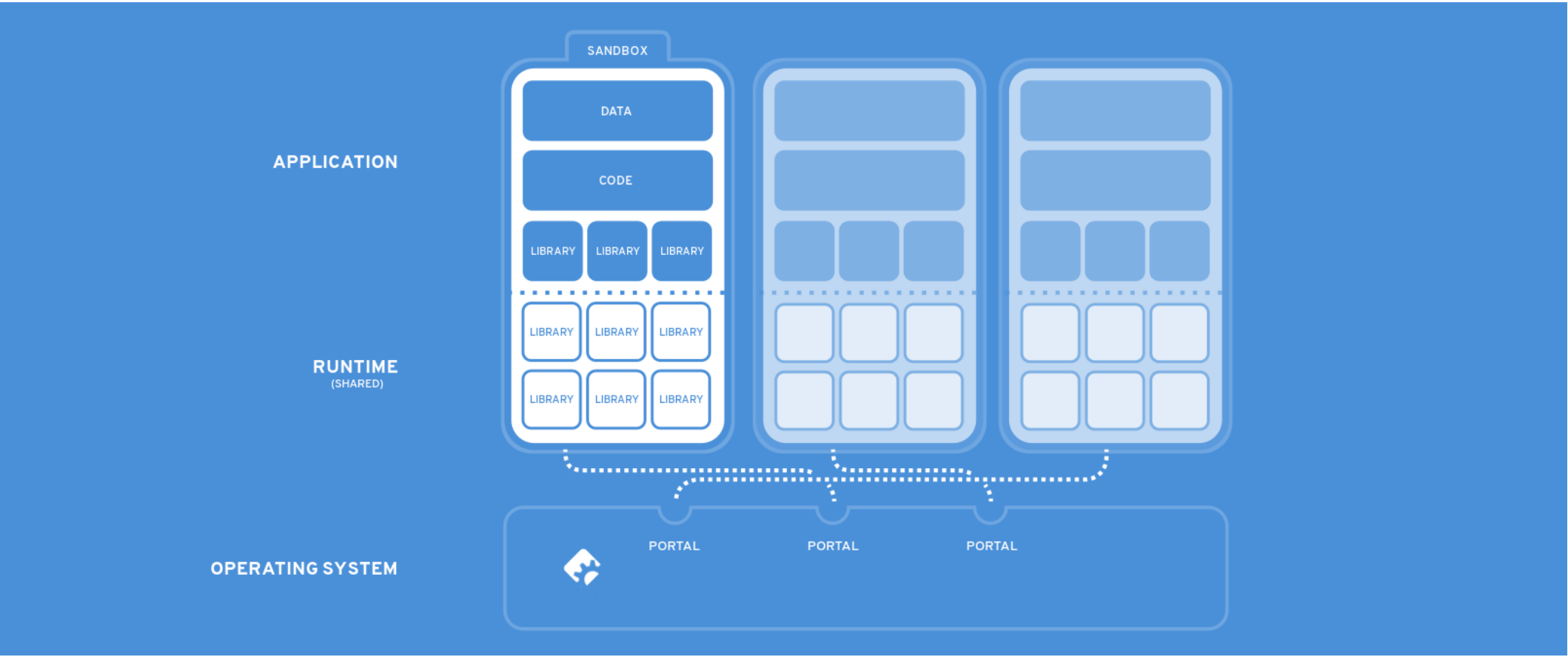
The Big Promises

- Compatibility – SOLVED
  - Only compatible libraries in the bundle
- Portability – SOLVED
  - All dependencies in the bundle
- Pace of Change – SOLVED
  - App developers can distribute at their pace, not a distro pace
- “It just works” - SOLVED

Compatibility & Portability



Compatibility & Portability



Compatibility & Portability

- Containerised Apps at some point make assumptions of a common standard base provided by the Distribution
- No such common base exists in a practical sense

Compatibility & Portability

not required.

2. **Gather suitable binaries of all dependencies** that are not part of the base operating systems you are targeting. For example, if you are targeting Ubuntu, Fedora, and openSUSE, then you need to gather all libraries and other dependencies that your app requires to run that are not part of Ubuntu, Fedora, and openSUSE.
3. **Create a working AppDir** from your binaries. A working AppImage runs your app when you execute its AppRun file

Compatibility & Portability

- For a Containerised App to be portable, it must contain ALL compatible dependencies which MIGHT not be provided by ANY distribution
- If not, expect crashes

So it’s hopeless?

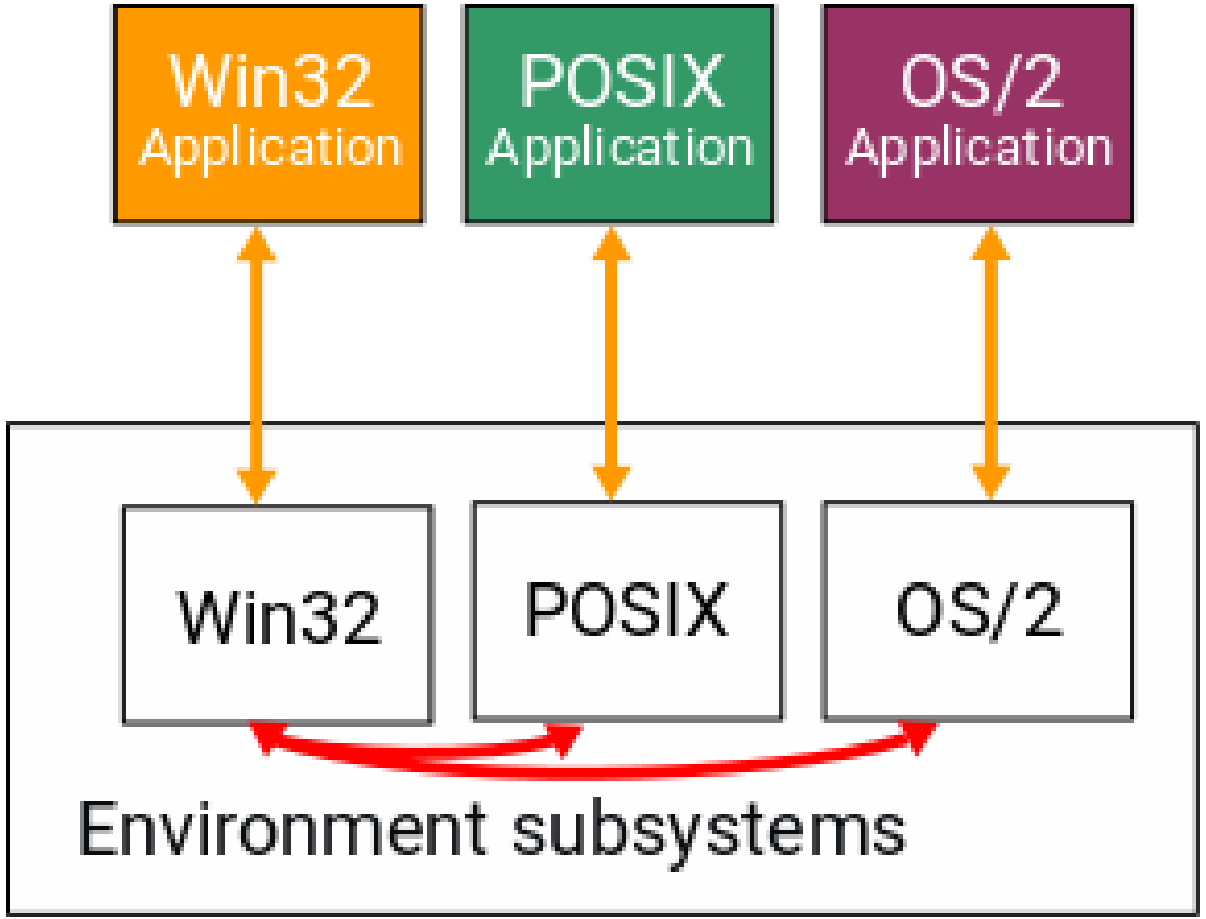
If everything is still liable to break, what is the point?

- Frameworks/Runtimes attempt to mitigate by providing curated ‘Middledistros’ to build Applications for
- The “Real” Solution: A well defined Linux Standard Base?

The Big Promises - Reality

- Compatibility – ~~SOLVED~~
  - Only compatible libraries in the bundle
- Portability – ~~SOLVED~~
  - All dependencies in the bundle
- Pace of Change – SOLVED
  - App developers can distribute at their pace, not a distro pace
- “It just works” - ?

Wait a second...



History Repeating?

- Security nightmare?
  - Security relevant libs lurking in countless application bundles
- Maintenance nightmare?
  - How are we going to update our app and every single lib?
- Legal nightmare?
  - Can we legally redistribute all the libs we need to?
- Storage vendor dream
  - More disk consumption, everyone buying bigger disks!

“With Great Power…”

“… Comes Great Responsibilities”

- AppImage/FlatPak/Snappy are tools that enable App Developers to directly distribute software without the ‘need’ for Distributions
- Therefore, they must adopt the responsibilities which come with being a distributor of software

Compatibility & Portability

Consider everything an App needs that isn't in the Bundle

- Can this break my App if the ABI changes?
  - If YES, then move it to the Bundle
- Can I rely on it being there on ALL systems?
  - If NO, then move it to the Bundle

Compatibility & Portability in Real Teams

Application Developers will still need to

- Dev & test Apps on every possible distro
- Then retest every App patch on every possible distro
- Then cry when it all breaks anyway

Broader Responsibilities

- Security – Monitor & rapidly react to CVEs. Audit libraries. Do not assume sandboxing is enough.
- Maintenance – Update all bundled dependencies in a timely manner
- Legal – Review licences of all bundled dependencies and ensure compliance & compatibility

Distributions can be part of the solution

- Distributions should like the promise of Containerised Applications
- Less work & responsibility for us is always good
- Should not be fearful of the transfer of responsibility, but should not encourage it blindly either

Distributions can be part of the solution

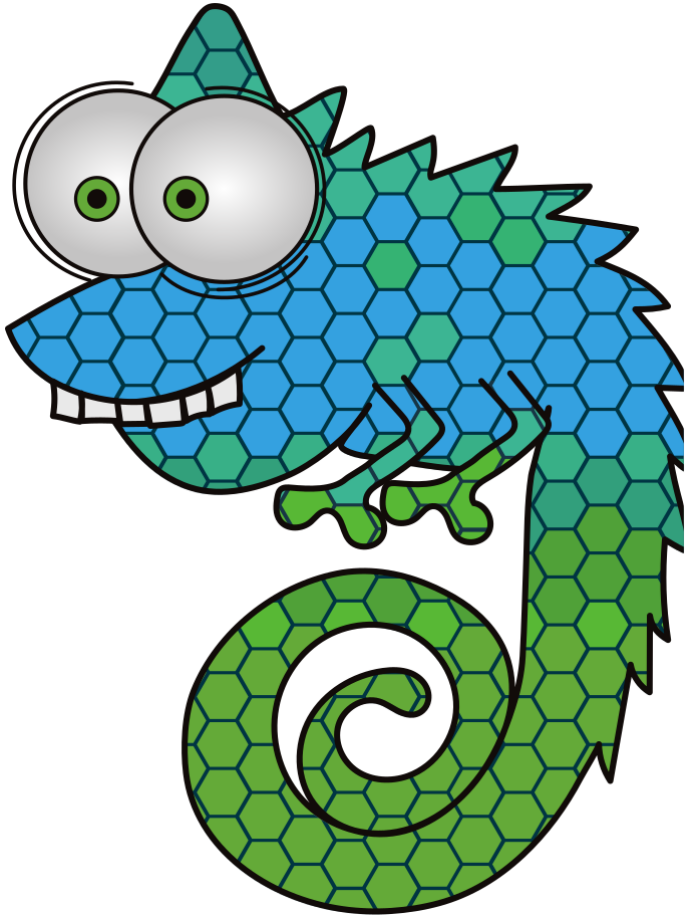
- A Common Base (“LSB for the Container Age”) must be considered
  - Without one, the portability promise is unachievable

- Distributions have decades of tools and talent for dealing with the broader issues. USE THEM
- Don’t reinvent every wheel just because we can

One more thing

Rolling Releases for Everyone?

- To get Applications in the hands of users fast, what model beats a rolling distribution?
- Users can be guaranteed an integrated “built together” experience
- Security/Maintenance burdens less broadly distributed, fewer points of failure, Devs don’t need to be security engineers
- “It just works” can be reached with good tools – OBS & openQA



Join Us at www.opensuse.org

License

General Disclaimer

Credits

Template
Richard Brown
rbrown@opensuse.org

Design & Inspiration
openSUSE Design Team
http://opensuse.github.io/branding-guidelines/

docker-library / **wordpress**

Watch 24 | Star 195 | Fork 144

Code | Issues 30 | Pull requests 8 | Pulse | Graphs

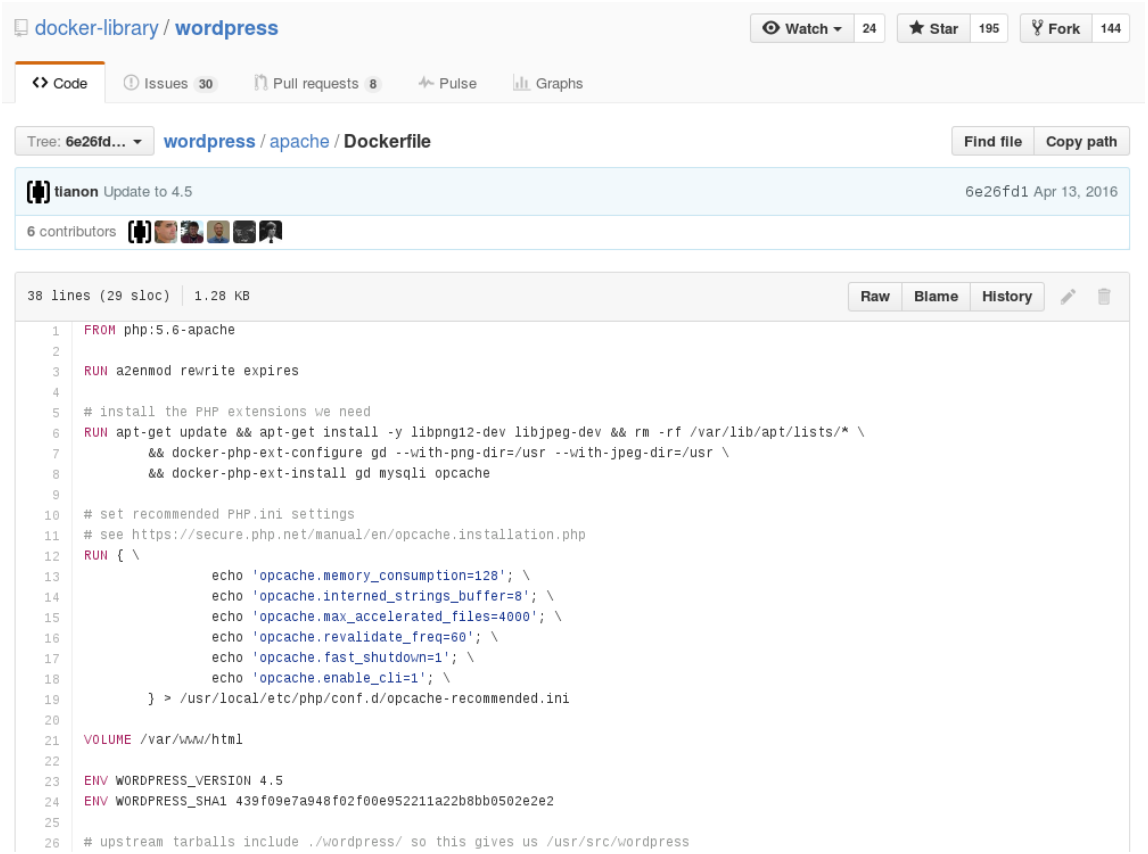
Tree: 6e26fd... | **wordpress** / apache / **Dockerfile** | Find file | Copy path

tianon Update to 4.5 — 6e26fd1 Apr 13, 2016

6 contributors

38 lines (29 sloc) | 1.28 KB | Raw | Blame | History

```
FROM php:5.6-apache

RUN a2enmod rewrite expires

# install the PHP extensions we need
RUN apt-get update && apt-get install -y libpng12-dev libjpeg-dev && rm -rf /var/lib/apt/lists/* \
        && docker-php-ext-configure gd --with-png-dir=/usr --with-jpeg-dir=/usr \
        && docker-php-ext-install gd mysqli opcache

# set recommended PHP.ini settings
# see https://secure.php.net/manual/en/opcache.installation.php
RUN { \
                echo 'opcache.memory_consumption=128'; \
                echo 'opcache.interned_strings_buffer=8'; \
                echo 'opcache.max_accelerated_files=4000'; \
                echo 'opcache.revalidate_freq=60'; \
                echo 'opcache.fast_shutdown=1'; \
                echo 'opcache.enable_cli=1'; \
        } > /usr/local/etc/php/conf.d/opcache-recommended.ini

VOLUME /var/www/html

ENV WORDPRESS_VERSION 4.5
ENV WORDPRESS_SHA1 439f09e7a948f02f00e952211a22b8bb0502e2e2

# upstream tarballs include ./wordpress/ so this gives us /usr/src/wordpress
RUN curl -o wordpress.tar.gz -SL https://wordpress.org/wordpress-${WORDPRESS_VERSION}.tar.gz \
        && echo "$WORDPRESS_SHA1 *wordpress.tar.gz" | sha1sum -c - \
        && tar -xzf wordpress.tar.gz -C /usr/src/ \
        && rm wordpress.tar.gz \
        && chown -R www-data:www-data /usr/src/wordpress

COPY docker-entrypoint.sh /entrypoint.sh

# grr, ENTRYPOINT resets CMD now
ENTRYPOINT ["/entrypoint.sh"]
CMD ["apache2-foreground"]
```



docker-library / wordpress

Watch 24 | Star 195 | Fork 144

Code | Issues 30 | Pull requests 8 | Pulse | Graphs

Tree: 6e26fd... | wordpress / apache / Dockerfile | Find file | Copy path

tianon Update to 4.5 — 6e26fd1 Apr 13, 2016

6 contributors

38 lines (29 sloc) | 1.28 KB | Raw | Blame | History

```
FROM php:5.6-apache

RUN a2enmod rewrite expires

# install the PHP extensions we need
RUN apt-get update && apt-get install -y libpng12-dev libjpeg-dev && rm -rf /var/lib/apt/lists/* \
        && docker-php-ext-configure gd --with-png-dir=/usr --with-jpeg-dir=/usr \
        && docker-php-ext-install gd mysqli opcache

# set recommended PHP.ini settings
# see https://secure.php.net/manual/en/opcache.installation.php
RUN { \
                echo 'opcache.memory_consumption=128'; \
                echo 'opcache.interned_strings_buffer=8'; \
                echo 'opcache.max_accelerated_files=4000'; \
                echo 'opcache.revalidate_freq=60'; \
                echo 'opcache.fast_shutdown=1'; \
                echo 'opcache.enable_cli=1'; \
        } > /usr/local/etc/php/conf.d/opcache-recommended.ini

VOLUME /var/www/html

ENV WORDPRESS_VERSION 4.5
ENV WORDPRESS_SHA1 439f09e7a948f02f00e952211a22b8bb0502e2e2

# upstream tarballs include ./wordpress/ so this gives us /usr/src/wordpress
```

```
RUN curl -o wordpress.tar.gz
         -SL https://wordpress.org/wordpress-$WORDPRESS_VERSION}.tar.gz
```

```
COPY docker-entrypoint.sh /entrypoint.sh

# grr, ENTRYPOINT resets CMD now
ENTRYPOINT ["/entrypoint.sh"]
CMD ["apache2-foreground"]
```

docker-library / **wordpress**

Watch 24 | Star 195 | Fork 144

Code | Issues 30 | Pull requests 8 | Pulse | Graphs

Tree: 6e26fd... | **wordpress** / apache / **Dockerfile** | Find file | Copy path

tianon Update to 4.5 — 6e26fd1 Apr 13, 2016

6 contributors

FROM php:5.6-apache

Raw | Blame | History

```
RUN a2enmod rewrite expires

# install the PHP extensions we need
RUN apt-get update && apt-get install -y libpng12-dev libjpeg-dev && rm -rf /var/lib/apt/lists/* \
        && docker-php-ext-configure gd --with-png-dir=/usr --with-jpeg-dir=/usr \
        && docker-php-ext-install gd mysqli opcache

# set recommended PHP.ini settings
# see https://secure.php.net/manual/en/opcache.installation.php
RUN { \
                echo 'opcache.memory_consumption=128'; \
                echo 'opcache.interned_strings_buffer=8'; \
                echo 'opcache.max_accelerated_files=4000'; \
                echo 'opcache.revalidate_freq=60'; \
                echo 'opcache.fast_shutdown=1'; \
                echo 'opcache.enable_cli=1'; \
        } > /usr/local/etc/php/conf.d/opcache-recommended.ini

VOLUME /var/www/html

ENV WORDPRESS_VERSION 4.5
ENV WORDPRESS_SHA1 439f09e7a948f02f00e952211a22b8bb0502e2e2

# upstream tarballs include ./wordpress/ so this gives us /usr/src/wordpress
RUN curl -o wordpress.tar.gz -SL https://wordpress.org/wordpress-${WORDPRESS_VERSION}.tar.gz \
        && echo "$WORDPRESS_SHA1 *wordpress.tar.gz" | sha1sum -c - \
        && tar -xzf wordpress.tar.gz -C /usr/src/ \
        && rm wordpress.tar.gz \
        && chown -R www-data:www-data /usr/src/wordpress

COPY docker-entrypoint.sh /entrypoint.sh

# grr, ENTRYPOINT resets CMD now
ENTRYPOINT ["/entrypoint.sh"]
CMD ["apache2-foreground"]
```

docker-library / php

Watch 58 Star 334 Fork 205

Code Issues 44 Pull requests 7 Pulse Graphs

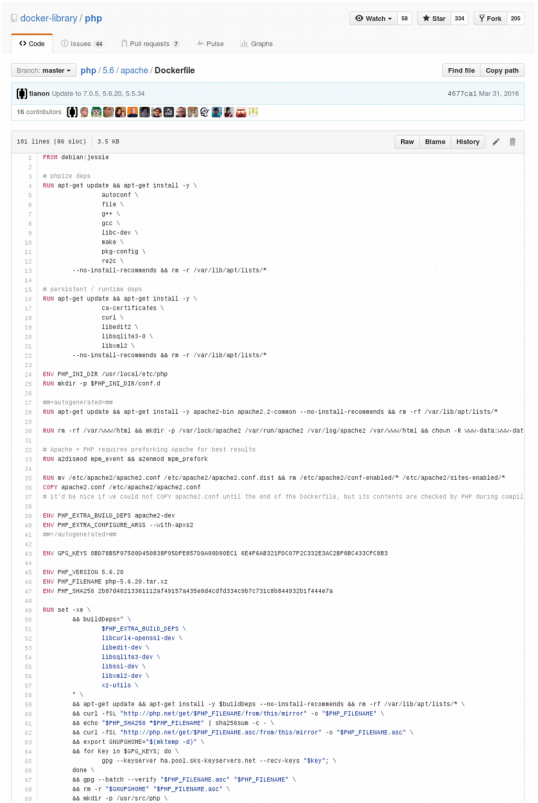
Branch: master php / 5.6 / apache / Dockerfile Find file Copy path

tianon Update to 7.0.5, 5.6.20, 5.5.34 4677ca1 Mar 31, 2016

16 contributors

101 lines (86 sloc) 3.5 KB Raw Blame History

```
FROM debian:jessie

# phpize deps
RUN apt-get update && apt-get install -y \
		autoconf \
		file \
		g++ \
		gcc \
		libc-dev \
		make \
		pkg-config \
		re2c \
	--no-install-recommends && rm -r /var/lib/apt/lists/*

# persistent / runtime deps
RUN apt-get update && apt-get install -y \
		ca-certificates \
		curl \
		libedit2 \
		libsqlite3-0 \
		libxml2 \
	--no-install-recommends && rm -r /var/lib/apt/lists/*

ENV PHP_INI_DIR /usr/local/etc/php
RUN mkdir -p $PHP_INI_DIR/conf.d

##<autogenerated>##
RUN apt-get update && apt-get install -y apache2-bin apache2.2-common --no-install-recommends && rm -rf /var/lib/apt/lists/*

RUN rm -rf /var/www/html && mkdir -p /var/lock/apache2 /var/run/apache2 /var/log/apache2 /var/www/html && chown -R www-data:www-dat

# Apache + PHP requires preforking Apache for best results
RUN a2dismod mpm_event && a2enmod mpm_prefork

RUN mv /etc/apache2/apache2.conf /etc/apache2/apache2.conf.dist && rm /etc/apache2/conf-enabled/* /etc/apache2/sites-enabled/*
COPY apache2.conf /etc/apache2/apache2.conf
# it'd be nice if we could not COPY apache2.conf until the end of the dockerfile, but its contents are checked by PHP during compil

ENV PHP_EXTRA_BUILD_DEPS apache2-dev
ENV PHP_EXTRA_CONFIGURE_ARGS --with-apxs2
##</autogenerated>##

ENV GPG_KEYS 0BD78B5F97500D450838F95DFE857D9A90D90EC1 6E4F6AB321FDC07F2C332E3AC2BF0BC433CFC8B3

ENV PHP_VERSION 5.6.20
ENV PHP_FILENAME php-5.6.20.tar.xz
ENV PHP_SHA256 2b87d4022133611124f491573443500d4cdfd334c9b7c731c8b844932b1f444e7a

RUN set -xe \
	&& buildDeps=" \
		$PHP_EXTRA_BUILD_DEPS \
		libcurl4-openssl-dev \
		libedit-dev \
		libsqlite3-dev \
		libssl-dev \
		libxml2-dev \
		xz-utils \
	" \
	&& apt-get update && apt-get install -y $buildDeps --no-install-recommends && rm -rf /var/lib/apt/lists/* \
	&& curl -fSL "http://php.net/get/$PHP_FILENAME/from/this/mirror" -o "$PHP_FILENAME" \
	&& echo "$PHP_SHA256 *$PHP_FILENAME" | sha256sum -c - \
	&& curl -fSL "http://php.net/get/$PHP_FILENAME.asc/from/this/mirror" -o "$PHP_FILENAME.asc" \
	&& export GNUPGHOME="$(mktemp -d)" \
	&& for key in $GPG_KEYS; do \
		gpg --keyserver ha.pool.sks-keyservers.net --recv-keys "$key"; \
	done \
	&& gpg --batch --verify "$PHP_FILENAME.asc" "$PHP_FILENAME" \
	&& rm -r "$GNUPGHOME" "$PHP_FILENAME.asc" \
	&& mkdir -p /usr/src/php \
	&& tar -xf "$PHP_FILENAME" -C /usr/src/php --strip-components=1 \
	&& rm "$PHP_FILENAME" \
	&& cd /usr/src/php \
	&& ./configure \
		--with-config-file-path="$PHP_INI_DIR" \
		--with-config-file-scan-dir="$PHP_INI_DIR/conf.d" \
		$PHP_EXTRA_CONFIGURE_ARGS \
		--disable-cgi \
# --enable-mysqlnd is included here because it's harder to compile after the fact than extensions are (since it's a plugin for serve
		--enable-mysqlnd \
# --enable-mbstring is included here because otherwise there's no way to get pecl to use it properly (see https://github.com/docker
		--enable-mbstring \
		--with-curl \
		--with-libedit \
		--with-openssl \
		--with-zlib \
	&& make -j"$(nproc)" \
	&& make install \
	&& { find /usr/local/bin /usr/local/sbin -type f -executable -exec strip --strip-all '{}' + || true; } \
	&& make clean \
	&& apt-get purge -y --auto-remove -o APT::AutoRemove::RecommendsImportant=false -o APT::AutoRemove::SuggestsImportant=false

COPY docker-php-ext-* /usr/local/bin/

##<autogenerated>##
COPY apache2-foreground /usr/local/bin/
WORKDIR /var/www/html

EXPOSE 80
CMD ["apache2-foreground"]
##</autogenerated>##
```



docker-library / php

Watch 58 ★ Star 334 Fork 205

Code Issues 44 Pull requests 7 Pulse Graphs

Branch: master php / 5.6 / apache / Dockerfile Find file Copy path

tianon Update to 7.0.5, 5.6.20, 5.5.34 4677ca1 Mar 31, 2016

16 contributors

101 lines (86 sloc) 3.5 KB Raw Blame History

```
FROM debian:jessie

# phpize deps
RUN apt-get update && apt-get install -y \
		autoconf \
		file \
		g++ \
		gcc \
		libc-dev \
		make \
		pkg-config \
		re2c \
	--no-install-recommends && rm -r /var/lib/apt/lists/*

# persistent / runtime deps
RUN apt-get update && apt-get install -y \
		ca-certificates \
		curl \
		libedit2 \
		libsqlite3-0 \
		libxml2 \
	--no-install-recommends && rm -r /var/lib/apt/lists/*

ENV PHP_INI_DIR /usr/local/etc/php
RUN mkdir -p $PHP_INI_DIR/conf.d

##<autogenerated>##
RUN apt-get update && apt-get install -y apache2-bin apache2.2-common --no-install-recommends && rm -rf /var/lib/apt/lists/*

RUN rm -rf /var/www/html && mkdir -p /var/lock/apache2 /var/run/apache2 /var/log/apache2 /var/www/html && chown -R www-data:www-dat

# Apache + PHP requires preforking Apache for best results
RUN a2dismod mpm_event && a2enmod mpm_prefork

RUN mv /etc/apache2/apache2.conf /etc/apache2/apache2.conf.dist && rm /etc/apache2/conf-enabled/* /etc/apache2/sites-enabled/*
COPY apache2.conf /etc/apache2/apache2.conf
# it'd be nice if we could not COPY apache2.conf until the end of the Dockerfile, but its contents are checked by PHP during compil

ENV PHP_EXTRA_BUILD_DEPS apache2-dev
ENV PHP_EXTRA_CONFIGURE_ARGS --with-apxs2
##</autogenerated>##

ENV GPG_KEYS 0BD78B5F97500D450838F95DFE857D9A90D90EC1 6E4F6AB321FDC07F2C332E3AC2BF0BC433CFC8B3

ENV PHP_VERSION 5.6.20
ENV PHP_FILENAME php-5.6.20.tar.xz
ENV PHP_SHA256 2b87d4022133611128af49157443500d4cdfd334c9b7c731c8b844932b1f444e7a

RUN set -xe \
	&& buildDeps=" \
		$PHP_EXTRA_BUILD_DEPS \
		libcurl4-openssl-dev \
		libedit-dev \
		libsqlite3-dev \
		libssl-dev \
		libxml2-dev \
		xz-utils \
	" \
	&& apt-get update && apt-get install -y $buildDeps --no-install-recommends && rm -rf /var/lib/apt/lists/* \
	&& curl -fSL "http://php.net/get/$PHP_FILENAME/from/this/mirror" -o "$PHP_FILENAME" \
	&& echo "$PHP_SHA256 *$PHP_FILENAME" | sha256sum -c - \
	&& curl -fSL "http://php.net/get/$PHP_FILENAME.asc/from/this/mirror" -o "$PHP_FILENAME.asc" \
	&& export GNUPGHOME="$(mktemp -d)" \
	&& for key in $GPG_KEYS; do \
		gpg --keyserver ha.pool.sks-keyservers.net --recv-keys "$key"; \
	done \
	&& gpg --batch --verify "$PHP_FILENAME.asc" "$PHP_FILENAME" \
	&& rm -r "$GNUPGHOME" "$PHP_FILENAME.asc" \
	&& mkdir -p /usr/src/php \
```

```
	&& curl -fSL "http://php.net/get/$PHP_FILENAME/from/this/mirror" -o "$PHP_FILENAME" \
	&& echo "$PHP_SHA256 *$PHP_FILENAME" | sha256sum -c - \
	&& curl -fSL "http://php.net/get/$PHP_FILENAME.asc/from/this/mirror" -o "$PHP_FILENAME.asc" \
	&& export GNUPGHOME="$(mktemp -d)" \
	&& for key in $GPG_KEYS; do \
		gpg --keyserver ha.pool.sks-keyservers.net --recv-keys "$key"; \
	done \
	&& gpg --batch --verify "$PHP_FILENAME.asc" "$PHP_FILENAME" \
	&& rm -r "$GNUPGHOME" "$PHP_FILENAME.asc" \
	&& mkdir -p /usr/src/php \
```

```
&& make -j"$(nproc)" \
&& make install \
```

```
	&& { find /usr/local/bin /usr/local/sbin -type f -executable -exec strip --strip-all '{}' + || true; } \
	&& make clean \
	&& apt-get purge -y --auto-remove -o APT::AutoRemove::RecommendsImportant=false -o APT::AutoRemove::SuggestsImportant=false

COPY docker-php-ext-* /usr/local/bin/

##<autogenerated>##
COPY apache2-foreground /usr/local/bin/
WORKDIR /var/www/html

EXPOSE 80
CMD ["apache2-foreground"]
##</autogenerated>##
```


docker-library / php
Watch 58
Star 334
Fork 205
Code
Issues 44
Pull requests 7
Pulse
Graphs
Branch: master
php / 5.6 / apache / Dockerfile
Find file
Copy path
2016
FROM debian:jessie
phpize deps
RUN apt-get update && apt-get install -y \
autoconf \
file \
g++ \
gcc \
libc-dev \
make \
pkg-config \
re2c \
--no-install-recommends && rm -r /var/lib/apt/lists/*
persistent / runtime deps
RUN apt-get update && apt-get install -y \
ca-certificates \
curl \
libedit2 \
libsqlite3-0 \
libxml2 \
--no-install-recommends && rm -r /var/lib/apt/lists/*
ENV PHP_INI_DIR /usr/local/etc/php
RUN mkdir -p $PHP_INI_DIR/conf.d
##<autogenerated>##
RUN apt-get update && apt-get install -y apache2-bin apache2.2-common --no-install-recommends && rm -rf /var/lib/apt/lists/*
RUN rm -rf /var/www/html && mkdir -p /var/lock/apache2 /var/run/apache2 /var/log/apache2 /var/www/html && chown -R www-data:www-dat
Apache + PHP requires preforking Apache for best results
RUN a2dismod mpm_event && a2enmod mpm_prefork
RUN mv /etc/apache2/apache2.conf /etc/apache2/apache2.conf.dist && rm /etc/apache2/conf-enabled/* /etc/apache2/sites-enabled/*
COPY apache2.conf /etc/apache2/apache2.conf
it'd be nice if we could not COPY apache2.conf until the end of the dockerfile, but its contents are checked by PHP during compil
ENV PHP_EXTRA_BUILD_DEPS apache2-dev
ENV PHP_EXTRA_CONFIGURE_ARGS --with-apxs2
##</autogenerated>##
ENV GPG_KEYS 0BD78B5F97500D450838F95DFE857D9A90D90EC1 6E4F6AB321FDC07F2C332E3AC2BF0BC433CFC8B3
ENV PHP_VERSION 5.6.20
ENV PHP_FILENAME php-5.6.20.tar.xz
ENV PHP_SHA256 2b87d4022133611124af49157443560d4cdfd334c9b7c731c8b844932b1f444e7a
RUN set -xe \
&& buildDeps=" \
$PHP_EXTRA_BUILD_DEPS \
libcurl4-openssl-dev \
libedit-dev \
libsqlite3-dev \
libssl-dev \
libxml2-dev \
xz-utils \
" \
&& apt-get update && apt-get install -y $buildDeps --no-install-recommends && rm -rf /var/lib/apt/lists/* \
&& curl -fSL "http://php.net/get/$PHP_FILENAME/from/this/mirror" -o "$PHP_FILENAME" \
&& echo "$PHP_SHA256 *$PHP_FILENAME" | sha256sum -c - \
&& curl -fSL "http://php.net/get/$PHP_FILENAME.asc/from/this/mirror" -o "$PHP_FILENAME.asc" \
&& export GNUPGHOME="$(mktemp -d)" \
&& for key in $GPG_KEYS; do \
gpg --keyserver ha.pool.sks-keyservers.net --recv-keys "$key"; \
done \
&& gpg --batch --verify "$PHP_FILENAME.asc" "$PHP_FILENAME" \
&& rm -r "$GNUPGHOME" "$PHP_FILENAME.asc" \
&& mkdir -p /usr/src/php \
&& tar -xf "$PHP_FILENAME" -C /usr/src/php --strip-components=1 \
&& rm "$PHP_FILENAME" \
&& cd /usr/src/php \
&& ./configure \
--with-config-file-path="$PHP_INI_DIR" \
--with-config-file-scan-dir="$PHP_INI_DIR/conf.d" \
$PHP_EXTRA_CONFIGURE_ARGS \
--disable-cgi \
--enable-mysqlnd is included here because it's harder to compile after the fact than extensions are (since it's a plugin for seve
--enable-mysqlnd \
--enable-mbstring is included here because otherwise there's no way to get pecl to use it properly (see https://github.com/docker
--enable-mbstring \
--with-curl \
--with-libedit \
--with-openssl \
--with-zlib \
&& make -j"$(nproc)" \
&& make install \
&& { find /usr/local/bin /usr/local/sbin -type f -executable -exec strip --strip-all '{}' + || true; } \
&& make clean \
&& apt-get purge -y --auto-remove -o APT::AutoRemove::RecommendsImportant=false -o APT::AutoRemove::SuggestsImportant=false
COPY docker-php-ext-* /usr/local/bin/
##<autogenerated>##
COPY apache2-foreground /usr/local/bin/
WORKDIR /var/www/html
EXPOSE 80
CMD ["apache2-foreground"]
##</autogenerated>##

tianon / **docker-brew-debian**

Watch 18 | Star 64 | Fork 38

Code | Issues 3 | Pull requests 0 | Wiki | Pulse | Graphs

Tree: **04fb8b...** **docker-brew-debian** / jessie / **Dockerfile** — Find file | Copy path

tianon 2016-03-01 debootstraps (CVE-2016-0800, CVE-2016-0705, CVE-2016-0798,... d431f09 Mar 1, 2016

1 contributor

4 lines (3 sloc) | 51 Bytes — Raw | Blame | History

```
1 FROM scratch
2 ADD rootfs.tar.xz /
3 CMD ["/bin/bash"]
```