docker-library / **php**

Code **Issues** 24 Pull requests 7 Actions Security Insights

New issue Jump to bottom

PHP 7.4 - Problems on GD with JPEG and FreeType support #926

Closed **prsolucoes** opened this issue on 17 Dec 2019 · 12 comments

Labels question

prsolucoes commented on 17 Dec 2019

Hi,

Im trying use PHP 7.4 with GD with JPEG and FreeType support. But i don't know why it is not enabled.
Code:

```
php -r 'print_r(gd_info());'
Array
(
    [GD Version] => bundled (2.1.0 compatible)
    [FreeType Support] =>
    [GIF Read Support] => 1
    [GIF Create Support] => 1
    [JPEG Support] =>
    [PNG Support] => 1
    [WBMP Support] => 1
    [XPM Support] =>
    [XBM Support] => 1
    [WebP Support] =>
    [BMP Support] => 1
    [TGA Read Support] => 1
    [JIS-mapped Japanese Font Support] =>
)
```

My Dockerfile:

```
######
# See: https://hub.docker.com/_/php/
######

FROM php:7.4.0-fpm
MAINTAINER Paulo Coutinho <paulo@prsolucoes.com>
```

```
######
# You can configure php extensions using docker-php-ext-configure
# You can install php extensions using docker-php-ext-install
######

# define timezone
RUN echo "America/Sao_Paulo" > /etc/timezone
RUN dpkg-reconfigure -f noninteractive tzdata
RUN /bin/echo -e "LANG=\"en_US.UTF-8\"" > /etc/default/local

# install dependencies
RUN apt-get update
RUN apt-get install -y build-essential
RUN apt-get install -y libfreetype6-dev
RUN apt-get install -y libjpeg62-turbo-dev
RUN apt-get install -y libmcrypt-dev
RUN apt-get install -y libpng-dev
RUN apt-get install -y curl
RUN apt-get install -y libcurl4
RUN apt-get install -y libcurl4-openssl-dev
RUN apt-get install -y zlib1g-dev
RUN apt-get install -y libicu-dev
RUN apt-get install -y libmemcached-dev
RUN apt-get install -y memcached
RUN apt-get install -y default-mysql-client
RUN apt-get install -y libmagickwand-dev
RUN apt-get install -y unzip
RUN apt-get install -y libzip-dev
RUN apt-get install -y zip

# memcached
RUN pecl install memcached-3.1.5
RUN docker-php-ext-enable memcached

# mcrypt
RUN pecl install mcrypt-1.0.3
RUN docker-php-ext-enable mcrypt

# configure, install and enable all php packages
RUN docker-php-ext-configure gd --enable-gd --with-freetype --with-jpeg
RUN docker-php-ext-configure pdo_mysql --with-pdo-mysql=mysqlnd
RUN docker-php-ext-configure mysqli --with-mysqli=mysqlnd
RUN docker-php-ext-configure intl
RUN docker-php-ext-configure zip

RUN docker-php-ext-install -j$(nproc) pdo_mysql
RUN docker-php-ext-install -j$(nproc) mysqli
RUN docker-php-ext-install -j$(nproc) pdo
RUN docker-php-ext-install -j$(nproc) gd
RUN docker-php-ext-install -j$(nproc) intl
RUN docker-php-ext-install -j$(nproc) zip

# install xdebug
RUN pecl install xdebug
RUN docker-php-ext-enable xdebug

RUN echo "xdebug.remote_enable=1" >> /usr/local/etc/php/conf.d/docker-php-ext-
xdebug.ini
RUN echo "xdebug.remote_autostart=0" >> /usr/local/etc/php/conf.d/docker-php-ext-
xdebug.ini
RUN echo "xdebug.default_enable=0" >> /usr/local/etc/php/conf.d/docker-php-ext-
```

```
xdebug.ini
RUN echo "xdebug.remote_host=host.docker.internal" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
RUN echo "xdebug.remote_port=9000" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
RUN echo "xdebug.remote_connect_back=0" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
RUN echo "xdebug.profiler_enable=0" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
RUN echo "xdebug.remote_log=\"/tmp/xdebug.log\"" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini

# install imagick
RUN pecl install imagick
RUN docker-php-ext-enable imagick

# install composer
RUN cd /tmp \
    && curl -sS https://getcomposer.org/installer | php \
    && mv composer.phar /usr/local/bin/composer

# clean image
RUN apt-get clean
```

Can anyone help me?

☺

0xbyc commented on 17 Dec 2019

https://www.php.net/manual/en/image.installation.php

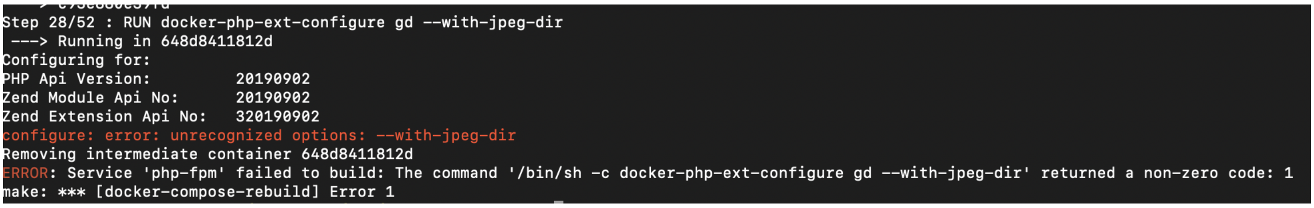
use `--with-jpeg-dir`

☺

prsolucoes commented on 17 Dec 2019

Hi,

Im talking about PHP 7.4. It don't have --with-jpeg-dir, i already tried it:



```
Step 28/52 : RUN docker-php-ext-configure gd --with-jpeg-dir
 ---> Running in 648d8411812d
Configuring for:
PHP Api Version:         20190902
Zend Module Api No:      20190902
Zend Extension Api No:   320190902
configure: error: unrecognized options: --with-jpeg-dir
Removing intermediate container 648d8411812d
ERROR: Service 'php-fpm' failed to build: The command '/bin/sh -c docker-php-ext-configure gd --with-jpeg-dir' returned a non-zero code: 1
make: *** [docker-compose-rebuild] Error 1
```

Im trying migrate the dockerfile from my project to PHP 7.4. Your way is the old way: https://github.com/prsolucoes/yii2-app-advanced/blob/master/extras/docker/php-fpm/Dockerfile#L56

This was changed and already talked about it here:
#912 (comment)

There is another thing that don't work in PHP 7.4 GD to support it.

👍 2 ☺

prsolucoes commented on 18 Dec 2019

My latest Dockerfile without success:

```
######
# See: https://hub.docker.com/_/php/
######

FROM php:7.4.0-fpm-buster
MAINTAINER Paulo Coutinho <paulo@prsolucoes.com>

######
# You can configure php extensions using docker-php-ext-configure
# You can install php extensions using docker-php-ext-install
######

# define timezone
RUN echo "America/Sao_Paulo" > /etc/timezone
RUN dpkg-reconfigure -f noninteractive tzdata
RUN /bin/echo -e "LANG=\"en_US.UTF-8\"" > /etc/default/local

# install dependencies
RUN apt-get update
RUN apt-get install -y --no-install-recommends \
    build-essential \
    libfreetype6-dev \
    libjpeg62-turbo-dev \
    libmcrypt-dev \
    libpng-dev \
    libwebp-dev \
    curl \
    libcurl4 \
    libcurl4-openssl-dev \
    zlib1g-dev \
    libicu-dev \
    libmemcached-dev \
    memcached \
    default-mysql-client \
    libmagickwand-dev \
    unzip \
    libzip-dev \
    zip \
    nano;

# memcached
RUN pecl install memcached-3.1.5
RUN docker-php-ext-enable memcached

# mcrypt
```

```
RUN pecl install mcrypt-1.0.3
RUN docker-php-ext-enable mcrypt

# configure, install and enable all php packages
RUN docker-php-ext-configure gd --enable-gd --with-freetype --with-jpeg --with-webp
RUN docker-php-ext-configure pdo_mysql --with-pdo-mysql=mysqlnd
RUN docker-php-ext-configure mysqli --with-mysqli=mysqlnd
RUN docker-php-ext-configure intl
RUN docker-php-ext-configure zip

RUN docker-php-ext-install -j$(nproc) opcache
RUN docker-php-ext-install -j$(nproc) pdo_mysql
RUN docker-php-ext-install -j$(nproc) mysqli
RUN docker-php-ext-install -j$(nproc) pdo
RUN docker-php-ext-install -j$(nproc) gd
RUN docker-php-ext-install -j$(nproc) intl
RUN docker-php-ext-install -j$(nproc) zip

# install xdebug
RUN pecl install xdebug
RUN docker-php-ext-enable xdebug

RUN echo "xdebug.remote_enable=1" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
RUN echo "xdebug.remote_autostart=0" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
RUN echo "xdebug.default_enable=0" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
RUN echo "xdebug.remote_host=host.docker.internal" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
RUN echo "xdebug.remote_port=9000" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
RUN echo "xdebug.remote_connect_back=0" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
RUN echo "xdebug.profiler_enable=0" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini
RUN echo "xdebug.remote_log=\"/tmp/xdebug.log\"" >> /usr/local/etc/php/conf.d/docker-php-ext-xdebug.ini

# configure opcache
RUN echo "opcache.memory_consumption=128" >> /usr/local/etc/php/conf.d/opcache-recommended.ini
RUN echo "opcache.interned_strings_buffer=8" >> /usr/local/etc/php/conf.d/opcache-recommended.ini
RUN echo "opcache.max_accelerated_files=4000" >> /usr/local/etc/php/conf.d/opcache-recommended.ini
RUN echo "opcache.revalidate_freq=2" >> /usr/local/etc/php/conf.d/opcache-recommended.ini
RUN echo "opcache.fast_shutdown=1" >> /usr/local/etc/php/conf.d/opcache-recommended.ini

# install imagick
RUN pecl install imagick-3.4.4
RUN docker-php-ext-enable imagick

# install composer
RUN cd /tmp \
    && curl -sS https://getcomposer.org/installer | php \
    && mv composer.phar /usr/local/bin/composer

# clean image
RUN apt-get clean
```

I don't understand why it don't compile with correct options.

prsolucoes commented on 18 Dec 2019

The log for command `docker-php-ext-configure gd --enable-gd --with-freetype --with-jpeg --with-webp` is:

```
root@89c47b87b6be:/usr/share/nginx/html# docker-php-ext-configure gd --enable-gd --
with-freetype --with-jpeg --with-webp
Configuring for:
PHP Api Version:         20190902
Zend Module Api No:      20190902
Zend Extension Api No:   320190902
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for a sed that does not truncate output... /bin/sed
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for cc... cc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether cc accepts -g... yes
checking for cc option to accept ISO C89... none needed
checking how to run the C preprocessor... cc -E
checking for icc... no
checking for suncc... no
checking for system library directory... lib
checking if compiler supports -R... no
checking if compiler supports -Wl,-rpath,... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-pc-linux-gnu
checking for PHP prefix... /usr/local
checking for PHP includes... -I/usr/local/include/php -I/usr/local/include/php/main -
I/usr/local/include/php/TSRM -I/usr/local/include/php/Zend -I/usr/local/include/php/ext
-I/usr/local/include/php/ext/date/lib
checking for PHP extension directory... /usr/local/lib/php/extensions/no-debug-non-zts-
20190902
checking for PHP installed headers prefix... /usr/local/include/php
checking if debug is enabled... no
checking if zts is enabled... no
checking for gawk... no
checking for nawk... nawk
checking if nawk is broken... no
checking for GD support... yes, shared
checking for external libgd... no
checking for libwebp... yes
checking for libjpeg... yes
checking for libXpm... no
checking for FreeType 2... yes
checking whether to enable JIS-mapped Japanese font support in GD... no
checking for fabsf... no
checking for floorf... no
checking for zlib... yes
checking for libpng... yes
checking for libwebp... yes
checking for libjpeg... yes
checking for freetype2... yes
checking for a sed that does not truncate output... /bin/sed
checking for ld used by cc... /usr/bin/ld
checking if the linker (/usr/bin/ld) is GNU ld... yes
checking for /usr/bin/ld option to reload object files... -r
checking for BSD-compatible nm... /usr/bin/nm -B
checking whether ln -s works... yes
checking how to recognize dependent libraries... pass_all
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking dlfcn.h usability... yes
checking dlfcn.h presence... yes
checking for dlfcn.h... yes
checking the maximum length of command line arguments... 1572864
checking command to parse /usr/bin/nm -B output from cc object... ok
checking for objdir... .libs
checking for ar... ar
checking for ranlib... ranlib
checking for strip... strip
checking if cc supports -fno-rtti -fno-exceptions... no
checking for cc option to produce PIC... -fPIC
checking if cc PIC flag -fPIC works... yes
checking if cc static flag -static works... yes
checking if cc supports -c -o file.o... yes
checking whether the cc linker (/usr/bin/ld -m elf_x86_64) supports shared libraries...
yes
checking whether -lc should be explicitly linked in... no
checking dynamic linker characteristics... GNU/Linux ld.so
checking how to hardcode library paths into programs... immediate
checking whether stripping libraries is possible... yes
checking if libtool supports shared libraries... yes
checking whether to build shared libraries... yes
checking whether to build static libraries... no

creating libtool
appending configuration tag "CXX" to libtool
configure: patching config.h.in
configure: creating ./config.status
config.status: creating config.h
```

prsolucoes commented on 18 Dec 2019

Other problem tips is that file **GD.SO** is not compiled with jpeg or freetype yet, because it don't have their symbols, only PNG:

```
root@89c47b87b6be:/usr/share/nginx/html# nm -C /usr/local/lib/php/extensions/no-debug-non-zts-20190902/gd.so | grep png
000000000005e4a0 d arginfo_imagecreatefrompng
000000000005e140 d arginfo_imagepng
000000000005c7a0 d arginfo_png2wbmp
                 U php_sig_png
                 U png_create_info_struct@@PNG16_0
                 U png_create_read_struct@@PNG16_0
                 U png_create_write_struct@@PNG16_0
                 U png_destroy_read_struct@@PNG16_0
                 U png_destroy_write_struct@@PNG16_0
                 U png_error@@PNG16_0
                 U png_get_IHDR@@PNG16_0
                 U png_get_PLTE@@PNG16_0
                 U png_get_error_ptr@@PNG16_0
                 U png_get_io_ptr@@PNG16_0
                 U png_get_pHYs@@PNG16_0
                 U png_get_rowbytes@@PNG16_0
                 U png_get_tRNS@@PNG16_0
                 U png_get_valid@@PNG16_0
                 U png_read_end@@PNG16_0
                 U png_read_image@@PNG16_0
                 U png_read_info@@PNG16_0
                 U png_read_update_info@@PNG16_0
                 U png_set_IHDR@@PNG16_0
                 U png_set_PLTE@@PNG16_0
                 U png_set_compression_level@@PNG16_0
                 U png_set_filter@@PNG16_0
                 U png_set_gray_to_rgb@@PNG16_0
                 U png_set_pHYs@@PNG16_0
                 U png_set_packing@@PNG16_0
                 U png_set_read_fn@@PNG16_0
                 U png_set_sig_bytes@@PNG16_0
                 U png_set_strip_16@@PNG16_0
                 U png_set_tRNS@@PNG16_0
                 U png_set_write_fn@@PNG16_0
                 U png_sig_cmp@@PNG16_0
                 U png_write_end@@PNG16_0
                 U png_write_image@@PNG16_0
                 U png_write_info@@PNG16_0
0000000000016030 T zif_imagecreatefrompng
00000000000175a0 T zif_imagepng
0000000000016c50 T zif_png2wbmp
000000000000fc0d t zif_png2wbmp.cold.45
root@89c47b87b6be:/usr/share/nginx/html# nm -C /usr/local/lib/php/extensions/no-debug-non-zts-20190902/gd.so | grep jpeg
root@89c47b87b6be:/usr/share/nginx/html# nm -C /usr/local/lib/php/extensions/no-debug-non-zts-20190902/gd.so | grep freetype
```

prsolucoes commented on 18 Dec 2019 • edited

Hi,

I was able to hack and fix it with the following lines:

```
RUN docker-php-ext-configure gd --enable-gd --with-freetype --with-jpeg --with-webp
RUN cd /usr/src/php/ext/gd && make
RUN cp /usr/src/php/ext/gd/modules/gd.so /usr/local/lib/php/extensions/no-debug-non-zts-20190902/gd.so
RUN docker-php-ext-install -j$(nproc) gd
```

Basically i call make from GD folder and checked that it has correct flags for jpeg and freetype. After, i copied the generated gd.so to PHP extensions folder.

And now the magic:

```
Array
(
    [GD Version] => bundled (2.1.0 compatible)
    [FreeType Support] => 1
    [FreeType Linkage] => with freetype
    [GIF Read Support] => 1
    [GIF Create Support] => 1
    [JPEG Support] => 1
    [PNG Support] => 1
    [WBMP Support] => 1
    [XPM Support] =>
    [XBM Support] => 1
    [WebP Support] => 1
    [BMP Support] => 1
    [TGA Read Support] => 1
    [JIS-mapped Japanese Font Support] =>
)
```

wglambert added the question label on 18 Dec 2019

wglambert commented on 18 Dec 2019 • edited

I'm not sure exactly why but the order of installation effects `gd`. I moved it to the top, and removed the PDO line as it's already installed (although you could keep it). And it installs with the right flags

▶ Dockerfile

▶ build

```
$ docker run --rm php:gd php -r 'print_r(gd_info());'
Array
(
    [GD Version] => bundled (2.1.0 compatible)
    [FreeType Support] => 1
    [FreeType Linkage] => with freetype
    [GIF Read Support] => 1
    [GIF Create Support] => 1
    [JPEG Support] => 1
    [PNG Support] => 1
    [WBMP Support] => 1
    [XPM Support] =>
    [XBM Support] => 1
    [WebP Support] => 1
    [BMP Support] => 1
    [TGA Read Support] => 1
    [JIS-mapped Japanese Font Support] =>
)
```

Also the `RUN apt-get clean` has to be in the same RUN line, previous layers can't be altered in later layers as the size doesn't change, so I added the usual `rm -rf /var/lib/apt/lists/*`

👍 6 🎉 1 ☺

prsolucoes commented on 18 Dec 2019

Hi,

Yes, changing the order make compilation works too.

Thanks.

☺

yosifkit commented on 31 Dec

Closing, seems to be resolved.

☺

⊘ **yosifkit** closed this on 31 Dec

prsolucoes commented on 2 Jan

Yes, was resolved, changing the order too.

Thanks.

👍 1

wglambert mentioned this issue on 20 Jan

How to enable JPEG Support in image 7.4.1-apache #945

Closed

shyamjos commented on 6 Aug

Moving GD to the top fixed my issue too , is this a bug in docker php extension installer script?

shyamjos mentioned this issue on 7 Aug

Order of php-ext-install command affects some features of GD library like freetype and jpeg - Bug PHP 7.4.x - #1047

Closed

tianon commented on 12 Aug

I thought this was odd, but had an epiphany this morning -- it's not a "bug" of `docker-php-ext-install`, but rather a feature:

php/docker-php-ext-install
Line 123 in `098e442`

```
123        docker-php-source delete
```

When it finishes working, `docker-php-ext-install` will delete the source code (which includes all the bits configured by `docker-php-ext-configure`). The intention is that you run `docker-php-ext-configure` some number of times, then run a *single* invocation of `docker-php-ext-install` (because there's really no reason to run those separately); also, it would be good to run them all in the same layer so the image doesn't bloat with the size of the PHP source code:

```
# configure, install and enable all php packages
RUN set -eux; \
        docker-php-ext-configure gd --enable-gd --with-freetype --with-jpeg --with-webp; \
        docker-php-ext-configure intl; \
        docker-php-ext-configure mysqli --with-mysqli=mysqlnd; \
        docker-php-ext-configure pdo_mysql --with-pdo-mysql=mysqlnd; \
        docker-php-ext-configure zip; \
        docker-php-ext-install -j "$(nproc)" \
                gd \
                intl \
                mysqli \
                opcache \
                pdo_mysql \
                zip
```

👍 2 🎉 1 ❤️ 1 👀 1 ☺

monsterxcn mentioned this issue on 24 Aug

GD 库缺少 Freetype Support metowolf/docker-php#5

ⓘ Open

Assignees

No one assigned

Labels

question

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

6 participants