php

编辑Dockerfile文件

|  |
| --- |
| FROM debian:jessie  # persistent / runtime deps  ENV PHPIZE\_DEPS \  autoconf \  file \  g++ \  gcc \  libc-dev \  make \  pkg-config \  re2c  RUN apt-get update && apt-get install -y \  $PHPIZE\_DEPS \  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>##  ENV PHP\_EXTRA\_CONFIGURE\_ARGS --enable-fpm --with-fpm-user=www-data --with-fpm-group=www-data  ##</autogenerated>##  ENV GPG\_KEYS 0BD78B5F97500D450838F95DFE857D9A90D90EC1 6E4F6AB321FDC07F2C332E3AC2BF0BC433CFC8B3  ENV PHP\_VERSION 5.6.22  ENV PHP\_FILENAME php-5.6.22.tar.xz  ENV PHP\_SHA256 c96980d7de1d66c821a4ee5809df0076f925b2fe0b8c362d234d92f2f0a178e2  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" \  $PHP\_EXTRA\_CONFIGURE\_ARGS \  --disable-cgi \  # --enable-mysqlnd is included here because it's harder to compile after the fact than extensions are (since it's a plugin for several extensions, not an extension in itself)  --enable-mysqlnd \  # --enable-mbstring is included here because otherwise there's no way to get pecl to use it properly (see https://github.com/docker-library/php/issues/195)  --enable-mbstring \  --with-curl \  --with-libedit \  --with-openssl \  --with-zlib \  && make -j"$(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 $buildDeps  COPY docker-php-ext-\* /usr/local/bin/  ##<autogenerated>##  WORKDIR /var/www/html  RUN set -ex \  && cd /usr/local/etc \  && if [ -d php-fpm.d ]; then \  # for some reason, upstream's php-fpm.conf.default has "include=NONE/etc/php-fpm.d/\*.conf"  sed 's!=NONE/!=!g' php-fpm.conf.default | tee php-fpm.conf > /dev/null; \  cp php-fpm.d/www.conf.default php-fpm.d/www.conf; \  else \  # PHP 5.x don't use "include=" by default, so we'll create our own simple config that mimics PHP 7+ for consistency  mkdir php-fpm.d; \  cp php-fpm.conf.default php-fpm.d/www.conf; \  { \  echo '[global]'; \  echo 'include=etc/php-fpm.d/\*.conf'; \  } | tee php-fpm.conf; \  fi \  && { \  echo '[global]'; \  echo 'error\_log = /proc/self/fd/2'; \  echo; \  echo '[www]'; \  echo '; if we send this to /proc/self/fd/1, it never appears'; \  echo 'access.log = /proc/self/fd/2'; \  echo; \  echo 'clear\_env = no'; \  echo; \  echo '; Ensure worker stdout and stderr are sent to the main error log.'; \  echo 'catch\_workers\_output = yes'; \  } | tee php-fpm.d/docker.conf \  && { \  echo '[global]'; \  echo 'daemonize = no'; \  echo; \  echo '[www]'; \  echo 'listen = [::]:9000'; \  } | tee php-fpm.d/zz-docker.conf  EXPOSE 9000  CMD ["php-fpm"] |

构建镜像

|  |
| --- |
| docker build -t litongjava/php . |

启动容器

可以对外映射的目录

|  |  |
| --- | --- |
| /usr/local/etc/php | 配置文件目录 |

-p 9000:9000 :将容器的9000端口映射到主机的9000端口

--name myphp-fpm :将容器命名为myphp-fpm

-v ~/nginx/www:/www :将主机中项目的目录www挂载到容器的/www

-v $PWD/logs:/phplogs ：将主机中当前目录下的logs目录挂载到容器的/phplogs

此处是通过nginx+php实现web服务，nginx配置文件的fastcgi\_pass应该配置为myphp-fpm容器的IP。

fastcgi\_pass 172.17.0.4:9000;

容器IP的查方法

docker inspect 容器ID或容器名 |grep '"IPAddress"'