

Sprawozdanie z laboratorium 6

Jakub Zelmanowicz, IO 6.10, 93075

Zadanie 1:

Celem zadania było utworzenie pliku docker-compose.yml w taki sposób, by tworzył on aplikację Apache + PHP + MySQL.

```
version: '3.7'

services:
  php:
    build: './php-fpm/'
    container_name: 'php'
    networks:
      - backend
    volumes:
      - ./public_html:/var/www/html/
  apache:
    build: './apache/'
    depends_on:
      - php
      - mysql
    networks:
      - frontend
      - backend
    ports:
      - "6666:80"
    volumes:
      - ./public_html:/var/www/html/
  mysql:
    image: mysql
    networks:
      - backend
    environment:
      - MYSQL_ROOT_PASSWORD=rootpassword

networks:
  frontend:
  backend:
```

Powyższy kod tworzy dwie sieci:

- frontend
- backend

Dzięki temu, poszczególne kontenery „wiedzą”, z którymi kontenerami powinny móc się łączyć.

Tworzymy także wolumen, który będzie kopiował zawartość katalogu public_html do katalogu kontenera /var/www/html (tam znajdują się pliki naszej strony).

Zawartość Dockerfile dla /php-fpm/:

```
FROM php:7.2.7-fpm-alpine3.7
RUN apk update; \
    apk upgrade;

RUN docker-php-ext-install mysqli
```

Na bazie dystrybucji alpine pobieramy obraz php-fpm, który następnie rozszerzamy o sterownik mysqli.

Zawartość Dockerfile dla /apache/:

```
FROM httpd
RUN apt-get update; \
    apt-get upgrade;

COPY php.apache.conf /usr/local/apache2/conf/php.apache.conf
RUN echo "Include /usr/local/apache2/conf/php.apache.conf" >> /usr/local/apache2/conf/httpd.conf

ServerName localhost

LoadModule deflate_module /usr/local/apache2/modules/mod_deflate.so
LoadModule proxy_module /usr/local/apache2/modules/mod_proxy.so
LoadModule proxy_fcgi_module /usr/local/apache2/modules/mod_proxy_fcgi.so

<VirtualHost *:80>
    # Proxy .php requests to port 9000 of the php-fpm container
    ProxyPassMatch ^/(.*\.php(/.*)?)$ fcgi://php:9000/var/www/html/$1
    DocumentRoot /var/www/html/
    <Directory /var/www/html/>
        DirectoryIndex index.php
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    # Send apache logs to stdout and stderr
    CustomLog /proc/self/fd/1 common
    ErrorLog /proc/self/fd/2
</VirtualHost>
```

Do kontenera httpd wstawiamy plik konfiguracyjny informujący serwer apache, by każda próba wywołania kodu PHP była delegowana do kontenera php.

Zawartość pliku PHP: index.php:

```
<?php
phpinfo();
?>
```

Proste wywołanie informacji o serwerze PHP.

Działająca strona:

PHP Version 7.2.7

System	Linux 970589a6c02 4.19.0-18-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) x86_64
Build Date	Jul 6 2018 16:25:42
Configure Command	'./configure' '--build=x86_64-linux-musl' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-sodium=shared' '--with-curl' '--with-libc' '--with-openssl' '--with-zlib' '--enable-fpm' '--with-fpm-user=www-data' '--with-fpm-group=www-data' '--disable-cgi' 'build_alias=x86_64-linux-musl'
Server API	FFM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-mysql.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies

zendengine

Aktywuj system Windows
Przejdź do ustawień, aby aktywować system Windows.

UWAGA!

Aby strona działała na porcie 6666, na niektórych przeglądarkach może być konieczne zezwolenie w ustawieniach przeglądarki na użycie tego portu.

Wnioski:

Dzięki docker-compose w bardzo łatwy i intuicyjny sposób możemy tworzyć złożone sieci kontenerów, co przekłada się na łatwiejsze tworzenie dużych, wydajnych oraz skalowalnych aplikacji wszelkiego typu.