

Sprawozdanie

Programowanie aplikacji w chmurze obliczeniowej

Laboratorium 6

NARZĘDZIE DOCKER COMPOSE

Łukasz Oleksiuk

Grupa: 6.6

Index: 097690

Zadanie 6.1 Budowanie środowiska kontenerowego z wykorzystaniem narzędzia compose

Przygotowane pliki i foldery:



apache



mysql



php



publichtml



docker-compose.yml



README.md

Dockerfile dla apache:

```
1 FROM alpine:latest
2 RUN apk update && \
3     apk add apache2 && \
4     rm -rf /var/cache/apk/* && \
5     mkdir -p /run/apache2
6 EXPOSE 80
7 CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
8
```

Dockerfile dla mysql:

```
1 FROM mysql:latest
2 ENV MYSQL_ROOT_PASSWORD=root
3 EXPOSE 3306
4
```

Dockerfile dla php:

```
1 FROM php:latest
2 WORKDIR /var/www/html
3 COPY . /var/www/html
4 EXPOSE 80
5 CMD ["php", "-S", "0.0.0.0:80"]
6
```

Plik index.html:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Witaj na mojej stronie!</title>
5 </head>
6 <body>
7   <h1>Witaj na mojej stronie!</h1>
8   <p>To jest prosta strona internetowa hostowana na moim serwerze Apache Docker.</p>
9 </body>
10 </html>
11
```

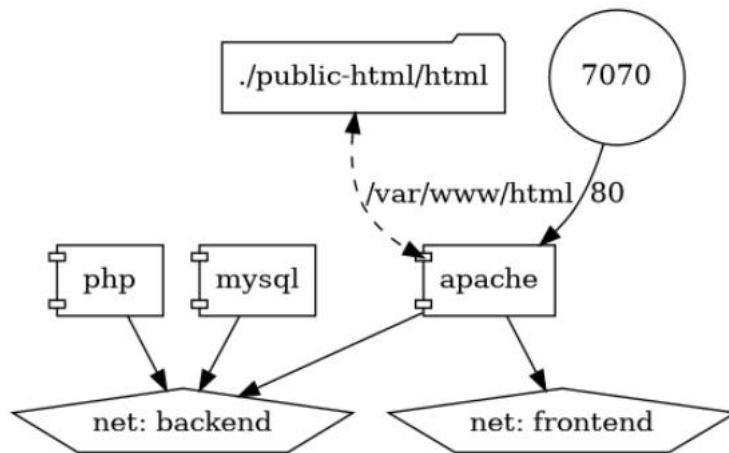
Plik docker-compose.yml:

```
1 version: '3'
2 services:
3   apache:
4     build: ./apache
5     ports:
6       - "7070:80"
7     volumes:
8       - ./public-html/html:/var/www/html
9     networks:
10      - frontend
11      - backend
12   php:
13     build: ./php
14     networks:
15       - backend
16   mysql:
17     build: ./mysql
18     environment:
19       MYSQL_ROOT_PASSWORD: root
20     networks:
21       - backend
22 networks:
23   frontend:
24   backend:
```

Plik Readme.md:

```
1 # Projekt Docker Compose LAMP
2
3 Ten projekt zawiera konfigurację Docker Compose do uruchomienia środowiska LAMP. Składa się z trzech
  usług: Apache, PHP i MySQL. Apache jest dostępny na porcie 6666.
4
5 ## Zawartość plików
6
7 - Dockerfile dla Apache
8 - Dockerfile dla PHP
9 - Dockerfile dla MySQL
10 - index.php (prosta strona internetowa)
11 - docker-compose.yml [konfiguracja usług]
```

Grafika reprezentująca graficznie utworzony plik docker-compose.yml:



Wnioski:

Ważnym jest poprawne skonfigurowanie i mapowanie woluminów pomiędzy hostem z tworzonymi kontenerami, a w razie nie poprawnego ich działania umiejętność debugowanie.

Serwer Apache jest wystawiony na porcie 7070, ponieważ port 6666 jest ograniczony przez przeglądarkę.

Do reprezentacji graficznej użyłem docker-compose-viz, a dokładnie polecenia:

```
docker run --rm -it --name dcw -v $(pwd):/input pmsipilot/docker-compose-viz render -m image docker-compose.yml, które tworzy grafikę ilustrującą plik docker-compose.yml
```