Laboratorium 5

Mateusz Markowski gr. 7.4/7

Zadanie 1

Wykorzystując przedstawiony wyżej obraz minimalnego systemu Alpine, utwórz obraz dla serwera Apache+PHP (przykładowy Dockerfile dla takiego serwera jest przedstawiony w części A niniejszej instrukcji). Uzupełnił Dockerfile o ewentualne, niezbędne komponenty.

W sprawozdaniu:

- podaj zawartość Dockerfile (from scratch) plik ma być umieszczony na prywatnym
- repozytorium na DockerHub,
- wykorzystane polecenia do budowy i uruchomienia tego obrazu (polecenie + wynik jego działania) oraz zrzut ekrany dowodzący, że serwer działa,
- wynik porównania wielkości utworzonego obrazu z analogicznym serwerem zbudowanym na bazie systemu Ubuntu:latest

Utworzenie obrazu oraz pliku Dockerfile na bazie obrazu Alpine:latest.

```
LABEL maintainer="Mateusz Markowski"
ENV PHPVERSION=7
RUN apk add --update apache2 php${PHPVERSION}-apache2 php${PHPVERSION} && \
rm -rf /var/cache/apk/* && \
mkdir /run/apache2/ && \
rm -rf /var/www/localhost/htdocs/index.html && \
echo "<php phpinfo(); ?>" > /var/www/localhost/htdocs/index.php && \
chmod 755 /var/www/localhost/htdocs/index.php
EXPOSE 80/tcp
ENTRYPOINT ["httpd"]
CMD ["-D","FOREGROUND"]
```

```
student@student-PwCh0:~/Desktop/zadanie1$ docker build . -t="mateusz-serwer"
Sending build context to Docker daemon 2.048kB
Step 1/7 : FROM alpine:latest
 ---> 196d12cf6ab1
Step 2/7 : LABEL maintainer="Mateusz Markowski"
 ---> Using cache
 ---> 43ee2480db74
Step 3/7 : ENV PHPVERSION=7
 ---> Running in 99655859ab63
Removing intermediate container 99655859ab63
 ---> 177e642747fe
Step 4/7 : RUN apk add --update apache2 php${PHPVERSION}-apache2 php${PHPVERSION} && rm -rf /var/cache/apk/* && mkdir /run/apache2/ && rm -rf /var/www/localhost/htdocs/index.html && echo "<php phpinfo(); ?>" > /var/www/localhost/htdocs/
index.php && chmod 755 /var/www/localhost/htdocs/index.php
 ---> Running in f55efcb3566a
fetch http://dl-cdn.alpinelinux.org/alpine/v3.8/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.8/community/x86_64/APKINDEX.tar.g
(1/14) Installing libuuid (2.32-r0)
(2/14) Installing apr (1.6.3-r1)
```

```
Executing busybox-1.28.4-r1.trigger
DK: 26 MiB in 27 packages
Removing intermediate container f55efcb3566a
---> 070dc88a88b1
Step 5/7 : EXPOSE 80/tcp
---> Running in 4648aa7522d3
Removing intermediate container 4648aa7522d3
---> 57b4f69d6042
Step 6/7 : ENTRYPOINT ["httpd"]
---> Running in d1b9e2f454b9
Removing intermediate container d1b9e2f454b9
---> 17541ea59103
Step 7/7 : CMD ["-D","FOREGROUND"]
---> Running in fac82f21ae97
Removing intermediate container fac82f21ae97
---> 55d4d9c11beb
Successfully built 55d4d9c11beb
Successfully tagged mateusz-serwer:latest
```

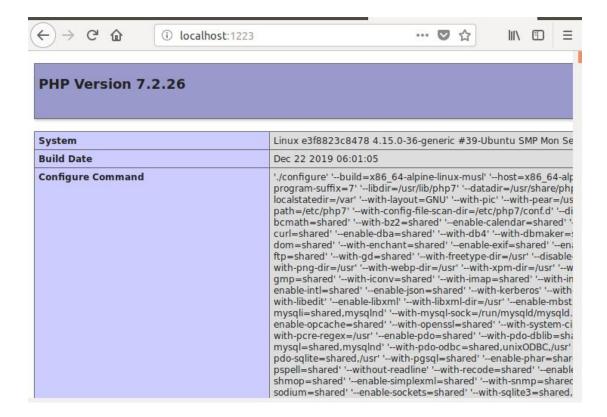
```
student@student-PwCh0:~$ docker images

REPOSITORY TAG IMAGE ID CREATED

SIZE

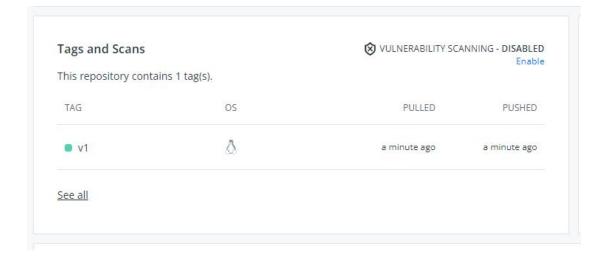
mateusz-serwer latest 55d4d9c11beb 5 minutes
ago 21.2MB
```

```
student@student-PwCh0:~/Desktop/zadanie1$ docker run -d -p 1223:80 mateusz-serw
er
018f80be8bdcc7e33796371626e54be28695b6d942db4ac5bc1855a2fc59ead5
student@student-PwCh0:~/Desktop/zadanie1$
```



Wypchnięcie obrazu Dockerfile na prywatne repozytorium DockerHub z wykorzystaniem konsoli.

```
student@student-PwChO:~/Desktop/zadanie1$ cd ...
student@student-PwChO:~/Desktop$ cd ..
student@student-PwCh0:~$ docker push mateuszmar/mateusz-serwer:v1
The push refers to repository [docker.io/mateuszmar/mateusz-serwer]
tag does not exist: mateuszmar/mateusz-serwer:v1
student@student-PwCh0:~$ docker tag mateuszmar/mateusz-serwer mateuszmar/mateus
z-serwer:v1
student@student-PwCh0:~$ docker push mateuszmar/mateusz-serwer:v1
The push refers to repository [docker.io/mateuszmar/mateusz-serwer]
8ea699053342: Pushing 6.147MB/16.83MB
df64d3292fd6: Mounted from library/alpine
student@student-PwChO:~$ docker push mateuszmar/mateusz-serwer:v1
The push refers to repository [docker.io/mateuszmar/mateusz-serwer]
8ea699053342: Pushed
df64d3292fd6: Mounted from library/alpine
v1: digest: sha256:5dca65f1b77b38910717640ac83d0acd40d4537f97fab5af11cb713faac3
86df size: 739
```

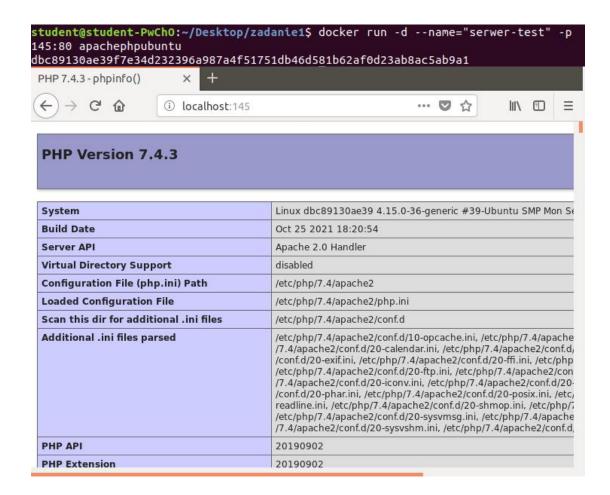




Zbudowanie serwera apache z PHP z wykorzystaniem obrazu Ubuntu:latest.

```
FROM ubuntu:latest
LABEL maintainer="Mateusz Markowski"
ENV DEBIAN_FRONTEND=nointeractive
RUN apt-get update &&\
    apt-get install -y apache2 &&\
    apt-get install -y apache2-utils&&\
    apt-get clean &&\
    apt-get install -y php libapache2-mod-php&&\
    service apache2 restart &&\
    rm -f /var/www/html/index.html&&\
    echo "<?php phpinfo();?>" > /var/www/html/index.php
CMD ["apache2ctl","-D","FOREGROUND"]
EXPOSE 80
```

student@student-PwCh0:~/Desktop/zadanie1\$ docker build . -t="apachephpubuntu"
Sending build context to Docker daemon 2.048kB



Porównanie wielkości obrazów Apache+PHP stworzonych z obrazów Ubuntu:latest oraz Alpine:latest.

apachephpubuntu		latest	48d0f978be5b	5 minutes
ago	236MB			
<none></none>		<none></none>	e384ad6c9227	11 hours
ago	72.8MB			
mateusz-serwer		latest	1d6f7bf528f7	16 hours
ago	21.2MB			

Wnioski:

Miałem problemy z pobraniem minimalnej wersji alpine z oficjalnej strony alpine. Strona nie ładowała się. Dlatego zdecydowałem się na stworzenie obrazu wykorzystując wersje alpine:latest. W celach porównawczych z obrazem ubuntu jest ona wystarczająco lekka, aby zauważyć różnicę w wielkości obrazów. Samo

zbudowanie oraz uruchomienie kontenera na bazie utworzonego obrazu przebiegło bez większych problemów. Cały proces tworzenia kontenera z obrazem przedstawiono na powyższych zrzutach ekranowych. Przy tworzeniu obrazu na bazie ubuntu napotkałem kilka problemów z instalacją php. Jednak po przeszukaniu najlepszego źródła wiedzy(StackOverflow) udało się rozwiązać problem z budowaniem obrazu oraz uruchomić kontener na porcie 145. Schemat postępowania w tworzeniu obrazu Ubuntu przedstawiono na powyższych zrzutach ekranowych. Po porównaniu obrazów widać, że ten sam kontener z tą samą stroną internetową zajmuje zdecydowanie więcej dla obrazu ubuntu niż serwera alpine.