

## Docker compose

### Zadanie 1

Ustaw usługę www zawierającą 2 wirtualne serwery mikrousługa oparta na apache2 postawionym na systemie opensuse

1.wirtualny host zewnętrzny dowołujący się do treści umieszczonej na hoście w katalogu strona1

2. wirtualny host z autoryzacją użytkownika geeko hasło novell treść strony w pliku index.html można zmieniać treść w pliku index z hosta głównego w katalogu strona1

Rozwiązanie:

Szukamy oficjalnego obrazu opensuse

```
docker-lab-0:~ # docker search opensuse
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
opensuse	DEPRECATED - for current images by the openS...	294	[OK]	
opensuse/portus	Authorization service and frontend for Docke...	75		[OK]
opensuse/leap	Official openSUSE Leap Images	49		

Na bazie tego obrazu będziemy budować obraz pod mikrousługę o nazwie webserwer1

Tworzymy plik Dockerfile na bazie którego będzie pracować mikrousługa w nowo utworzonym katalogu /docker\_examples/przyklad1/docker/

```
docker-lab-0:/docker_examples/przyklad1/docker # vi Dockerfile
```

```
### jaki obraz bazowy wykorzystujemy
```

```
FROM opensuse/leap
```

```
### autor
```

```
MATAINER Lucyna Pyzik
```

```
###instalowanie pakietów apache2 i czyszczenie pamięci podręcznej
```

```
RUN zypper in -y apache2 apache2-example-pages && zypper clean
```

Tworzymy na maszynie hoście katalogi w których umieścimy konfigurację wirtualnych hostów ich pliki index.html oraz plik z hasłem wygenerowanym dla użytkownika geeko Plik konfiguracyjny strony.conf znajdujący się w katalogu /docker\_examples/przyklad1/vhost ma następującą zawartość

```

docker-lab-0:/docker_examples/przyklad1 # cat vhost/strony.conf
<VirtualHost *:80>
ServerAdmin webmaster@dom.pl
ServerName strona1.local
DocumentRoot /srv/www/strona1
ErrorLog /var/log/apache2/strona1-error_log
CustomLog /var/log/apache2/strona1-access_log combined
UseCanonicalName On
ScriptAlias /cgi-bin/ "/srv/www/vhosts/cgi-bin"
<Directory "/srv/www/cgi-bin">
AllowOverride None
Options +ExecCGI +Includes
Require all granted
</Directory>
<Directory "/srv/www/strona1">
Options +Indexes +FollowSymLinks
AllowOverride None
Require all granted
</Directory>
</VirtualHost>
##### konfiguracja drugiego wirtualnego hosta strona2
<VirtualHost *:80>
ServerAdmin webmaster@dom.pl
ServerName strona2.local
DocumentRoot /srv/www/strona2
ErrorLog /var/log/apache2/strona2-error_log
CustomLog /var/log/apache2/strona2-access_log combined
UseCanonicalName On
ScriptAlias /cgi-bin/ "/srv/www/vhosts/cgi-bin"
<Directory "/srv/www/cgi-bin">
AllowOverride None
Options +ExecCGI +Includes
Require all granted
</Directory>
<Directory "/srv/www/strona2">
Options +Indexes +FollowSymLinks
AllowOverride None
AuthType Basic
AuthName "Intranet dla strona2"
#####lokalizacja pliku z hasłami

AuthUserFile /etc/apache2/htpasswd
Require valid-user geeko
#Require all granted
</Directory>
</VirtualHost>

```

W pliku konfiguracyjnym wirtualnych hostów użyliśmy plik z hasłami wirtualnych użytkowników o nazwie htpasswd

Utwórzmy katalog w którym będziemy przechowywać plik z hasłami użytkowników wirtualnych apache2

```

docker-lab-0:/docker_examples/przyklad1 # mkdir hasla
docker-lab-0:/docker_examples/przyklad1 # zypper in apache2-utils

```

Pakiet apache2-utils pozwala na utworzenie haseł dla wirtualnych użytkowników poleceniem htpasswd

```
docker-lab-0:/docker_examples/przyklad1 # htpasswd -c hasla/htpasswd geeko  
New password:  
Re-type new password:  
Adding password for user geeko
```

Parametr -c tworzy nowy plik, jeśli chcemy dodać użytkownika do już utworzonego pliku to nie dodajemy tego parametru.

W katalogach strona1 i strona2 umieszczamy odpowiednio pliki index.html

```
docker-lab-0:/docker_examples/przyklad1 # cat strona1/index.html  
<html>  
<head>  
  <title>Strona 1</title>  
  <meta content="">  
  <style></style>  
</head>  
<body>  
<h1>zmieniona strona pierwsza</h1>  
</body>  
</html>
```

Tworzymy plik docker-compose.yml do uruchomienia odpowiednio usługi serwera www z wirtualnymi hostami

```
version: "3.5"  
  
services:  
  
  webserver1:  
  
    build: ./docker/  
  
    container_name: web1  
  
    command: apachectl -D "FOREGROUND"  
  
    volumes:  
  
      - /docker_examples/przyklad1/strona1:/srv/www/strona1  
      - /docker_examples/przyklad1/strona2:/srv/www/strona2  
      - /docker_examples/przyklad1/vhost:/etc/apache  
      - /docker_examples/przyklad1/hasla/htpasswd:/etc/apache2/htpasswd  
  
    ports:  
  
      - 80:80
```

Wyjaśnienie do poszczególnych linii:

**## wersja dla docker-compose**

version: "3.5"

**##po tym wpisie umieszczamy jakie usługi będziemy uruchamiać**

services:

**## nazwa usługi**

webserver1:

**## budowanie obrazu z pliku Dockerfile znajdującego się w katalogu  
docker**

build: ./docker/.

**##nazwa tworzonego kontenera web1**

container\_name: web1

**##uruchomienie polecenia apachectl**

command: apachectl -D "FOREGROUND"

**## współdzielenie odpowiednich katalogów hosta i kontenera**

volumes:

- /docker\_examples/przyklad1/strona1:/srv/www/strona1

- /docker\_examples/przyklad1/strona2:/srv/www/strona2

- /docker\_examples/przyklad1/vhost:/etc/apache2/vhosts

-

/docker\_examples/przyklad1/hasla/htpasswd:/etc/apache2/htpasswd **##  
przekierowanie portu 80 hosta na 80 kontenera**

ports:

- 80:80

Uruchamiamy poleceniem docker-compose up -d

-d jest opcją uruchomienia usług w tle

**docker-lab-0:/docker\_examples/przyklad1 # docker-compose up -d**

*Building webserver1*

*Step 1/3 : FROM opensuse/leap*

*latest: Pulling from opensuse/leap*

*4f1ebd233674: Pull complete*

*Digest:*

*sha256:8db83c17759a0c802d2fce25a5e79b4be9e1a7cd3e2ea1a7a94b91fef5668d0c*

*Status: Downloaded newer image for opensuse/leap:latest*

*---> b1646e738a8a*

*Step 2/3 : MAINTAINER Lucyna Pyzik*

*---> Running in 58e03482aeb3*

*Removing intermediate container 58e03482aeb3*

*---> d324c96ec3d7*

*Step 3/3 : RUN zypper in -y apache2 apache2-example-pages && zypper clean*

*---> Running in 8dc10eae9b9*

*Retrieving repository 'Non-OSS Repository' metadata [..done]*

*.....*

*.....*

*All repositories have been cleaned up.*

*Removing intermediate container 8dc10eae9b9*

*---> b21c57886661*

*Successfully built b21c57886661*

*Successfully tagged przyklad1\_webserver1:latest*

*WARNING: Image for service webserver1 was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compose up --build`.*

*Creating web1 ... **done***

Wpisujemy w pliku /etc/hosts rozpoznawanie poprzez localhost strona1.local i strona2.local

Czyli linijkę

127.0.0.1 localhost strona1.local strona2.local

Możemy wówczas wywołać stronę programem

w3m strona1.local lub

w3m strona2.local

Można również wprowadzić zmiany w pliku /ect/hosts na linuxie lub C:\\windows\\system32\\drivers\\etc\\hosts na windowsie wpisując linijkę

192.168.100.17 strona1.local strona2.local

Gdzie 192.168.100.17 jest adresem hosta z zbudowanym i wystartowanym kontenerem webserver1

Otwieramy przeglądarkę i wpisujemy strona2.local Strona zgłasza się z okienkiem autoryzacyjnym dla użytkownika.

