

LABORATORIUM 7.

Zadanie 7.1.

Część I:

W zadaniu było trzeba uruchomić docker swarm i postawić na nim usługę nginx

Komendy:

1. `docker swarm init`
2. `docker service create --name ngweb --replicas 1 --publish 8080:80 nginx`

```
File Edit View Search Terminal Help
debian@debian10:~$ sudo docker service ls
ID            NAME      MODE      REPLICAS  IMAGE      PORTS
d0ciw9vs74j2  ngweb     replicated 1/1        nginx:latest *:8080->80/tcp
debian@debian10:~$
```

Część II:

W zadaniu było trzeba przeskalować usługę na 5 instancji.

Komendy:

- `docker service update ngweb --replicas 5`

```
File Edit View Search Terminal Help
debian@debian10:~$ sudo docker service ls
ID            NAME      MODE      REPLICAS  IMAGE      PORTS
d0ciw9vs74j2  ngweb     replicated 5/5        nginx:latest *:8080->80/tcp
debian@debian10:~$
```

Część III:

W zadaniu było trzeba zasymulować jednoczesną awarię 3 instancji.

Komendy:

- `docker container rm -f <id>`

```
debian@debian10:~$ sudo docker container ls
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
9bbcfad14956   nginx:latest "/docker-entrypoint..." 47 seconds ago Up 40 seconds 80/tcp       ngweb.3.xpwkac62ylvcmpw21tquuud
2eff0958b49e   nginx:latest "/docker-entrypoint..." 55 seconds ago Up 48 seconds 80/tcp       ngweb.2.s2hbincxidrryta6j4815t3ce
791f125b000d   nginx:latest "/docker-entrypoint..." About a minute ago Up 53 seconds 80/tcp       ngweb.1.irms77j6csx36bwku7s9glnau
104880dc2959   nginx:latest "/docker-entrypoint..." 5 minutes ago  Up 5 minutes  80/tcp       ngweb.4.uvpsgbvbkpdmllxxw0gyddvln5
2c6ea3e288f4   nginx:latest "/docker-entrypoint..." 5 minutes ago  Up 5 minutes  80/tcp       ngweb.5.iltgjcxdn6nb4j3b4xbvjw8ua
debian@debian10:~$ sudo docker container rm -f ngweb.1.irms77j6csx36bwku7s9glnau && sudo docker container rm -f ngweb.2.s2hbincxidrryta6j4815t3ce
ngweb.1.irms77j6csx36bwku7s9glnau
ngweb.2.s2hbincxidrryta6j4815t3ce
ngweb.3.xpwkac62ylvcmpw21tquuud
debian@debian10:~$ sudo docker service ls
ID            NAME      MODE      REPLICAS  IMAGE      PORTS
d0ciw9vs74j2  ngweb     replicated 2/5        nginx:latest *:8080->80/tcp
debian@debian10:~$
```

Usługi uruchamiają się ponownie po kilku sekundach:

```
debian@debian10:~$ sudo docker service ls
ID                NAME      MODE     REPLICAS  IMAGE          PORTS
d0ciw9vs74j2     ngweb     replicated 5/5        nginx:latest   *:8080->80/tcp
debian@debian10:~$
```

Wniosek:

Usługa Docker Swarm to bardzo narzędzie, które strzeże up-time'u naszej aplikacji.

Zadanie 2:

W zadaniu było trzeba przerobić plik docker-compose usługi LAMP na docker swarm:

Zawartość pliku docker-stack.yml:

```
version: '3.7'

services:
  php:
    image: jz-phpfpm
    networks:
      - backend
    volumes:
      - ./public_html:/var/www/html/
    deploy:
      replicas: 2
      update_config:
        parallelism: 2
        delay: 10s
      restart_policy:
        condition: on-failure

  apache:
    image: jz-apache
    depends_on:
      - php
      - mysql
    networks:
      - frontend
      - backend
    ports:
      - "6666:80"
    volumes:
      - ./public_html:/var/www/html/i
    deploy:
      replicas: 2
      update_config:
        parallelism: 2
        delay: 10s
      restart_policy:
        condition: on-failure

  mysql:
    image: mysql
    networks:
      - backend
    environment:
      - MYSQL_ROOT_PASSWORD=rootpassword

networks:
  frontend:
  backend:
```

Usługi:

- php - przetwarzanie plików PHP
- apache - dostarczanie zawartość strony

Polecenie uruchamiające cały docker-stack: docker stack deploy -c /home/debian/Documents/Docker/Lab_7/LAMP/docker-stack.yml lamp

```
debian@debian10:~/Documents/Docker/Lab_7/LAMP$ sudo docker service ls
ID                NAME          MODE          REPLICAS  IMAGE          PORTS
kbndujvbyc77     lamp_apache   replicated    2/2        jz-apache:latest *:6666->80/tcp
ta5gzivp1dub     lamp_mysql    replicated    1/1        mysql:latest
wci9veb15sd5     lamp_php      replicated    2/2        jz-phpfpm:latest
debian@debian10:~/Documents/Docker/Lab_7/LAMP$
```

Działanie usługi:

PHP Version 7.2.7	
System	Linux dc6dca8ad208 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-libedit' '--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	FPM/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	