

### 3. Gyakorlat

#### Docker webszerver, mysql konténerek

Először is létrehoztam egy mappát, amely mindhárom szolgáltatási részt fogja tartalmazni. Ennek a neve *webapp* lett. Ezután létrehoztam két újabb mappát. Egyikben a webszerver lesz található, másikban a mysql információi. Előbbi neve *site*, míg utóbbi neve *sql* lett.

A site mappába lépve létrehoztam 2 fájlt: egy index.php és egy Dockerfile fájlt. Ezekbe a következőket írtam:

```
1 <html>
2 <head>
3   <title>Selecting Records</title>
4 </head>
5 <body>
6   <h1>Table:</h1>
7   <?php
8     $dbhost = $_ENV["MYSQL_HOST"];
9     $dbuser = $_ENV["MYSQL_USER"];
10    $dbpass = $_ENV["MYSQL_PASSWORD"];
11    $database = $_ENV["MYSQL_DATABASE"];
12    $conn = mysqli_connect($dbhost, $dbuser, $dbpass, $database);
13
14    if(! $conn ) {
15      die('Could not connect: ' . mysqli_error($conn));
16    }
17
18    mysqli_select_db( $conn, 'db' );
19    $sql = "SELECT * FROM student";
20    $retval = mysqli_query( $conn, $sql );
21    if(! $retval ) {
22      die('Could not get data: ' . mysqli_error($conn));
23    }
24    echo "<table class='\"table table-striped\"' ><tr><th>id</th><th>name</th><th>neptuncode</th><th>age</th></tr>";
25    while($row = mysqli_fetch_array($retval)) {
26      echo "<tr>";
27      echo "<td>" . $row['id'] . "</td>";
28      echo "<td>" . $row['name'] . "</td>";
29      echo "<td>" . $row['neptuncode'] . "</td>";
30      echo "<td>" . $row['age'] . "</td>";
31      echo "</tr>";
32    }
33    echo "</table>";
34
35    mysqli_close($conn);
36  ?>
37 </body>
38 </html>
```

1.

```
1 FROM php:7.2-apache
2
3 ENV MYSQL_HOST=
4 ENV MYSQL_USER=
5 ENV MYSQL_PASSWORD=
6 ENV MYSQL_DATABASE=
7
8 RUN docker-php-ext-install mysqli && docker-php-ext-enable mysqli
9
10 RUN apt-get update
11
12 EXPOSE 80
13
14 COPY . /var/www/html/
15
```

2.

Ezek után átléptem az sql mappába, ahol létrehoztam egy commands.sql fájlt, amelybe a következőket írta:

```
1 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
2 START TRANSACTION;
3 SET time_zone = "+00:00";
4
5
6 CREATE DATABASE IF NOT EXISTS `db` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;
7 USE `db`;
8
9
10 CREATE TABLE `student` (
11   `id` int NOT NULL,
12   `name` varchar(255) DEFAULT NULL,
13   `neptuncode` varchar(6) DEFAULT NULL,
14   `age` int DEFAULT NULL,
15   PRIMARY KEY (`id`)
16 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3 COLLATE=utf8_hungarian_ci;;
17
18
19 INSERT INTO student VALUES(2, 'Gipsz Jakab', 'ABC123', 55 );
20 INSERT INTO student VALUES(3, 'teszt', 'neptun', 99 );
21
22 COMMIT;
23
```

Ezek után visszaléptem a főmappába, ahol két fájlt hoztam létre. Egyik egy .env nevű (amely tartalmazni fog pár változót, amelyet több helyen is használ a rendszer), másik pedig egy dockercompose.yaml fájl, amely a tényleges konténer felépítést fogja tartalmazni.

A .env-be a következőket írtam:

```
1 MYSQL_USER=web_user
2 MYSQL_PASSWORD=passw0rd
3 MYSQL_ROOT_PASSWORD=passw0rd
4 MYSQL_DATABASE=db
5
6 MYSQL_HOST=database
```

Ezek után átléptem a docker-compose.yaml-be, amelybe a következőket írtam:

```

1  version: "3.0"
2
3  services:
4    webservice:
5      build:
6        context: ./site
7        dockerfile: Dockerfile
8      image: beadando:1.0
9      environment:
10       - MYSQL_USER=${MYSQL_USER}
11       - MYSQL_PASSWORD=${MYSQL_PASSWORD}
12       - MYSQL_DATABASE=${MYSQL_DATABASE}
13       - MYSQL_HOST=${MYSQL_HOST}
14      ports:
15       - 80:80
16      networks:
17       - web
18      depends_on:
19        database:
20          condition: service_healthy
21
22      database:
23        image: mysql
24        command: --default-authentication-plugin=mysql_native_password
25        volumes:
26         - ./sql:/docker-entrypoint-initdb.d
27        environment:
28         - MYSQL_USER=${MYSQL_USER}
29         - MYSQL_PASSWORD=${MYSQL_PASSWORD}
30         - MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD}
31         - MYSQL_DATABASE=${MYSQL_DATABASE}
32        ports:
33         - 3306:3306
34        networks:
35         - web
36        healthcheck:
37         test: ["CMD", "mysqladmin" ,"ping", "-h", "localhost"]
38         timeout: 3s

```

```

39         retries: 10
40
41      adminer:
42        image: adminer
43        networks:
44         - web
45        ports:
46         - 127.0.0.1:8080:8080
47        depends_on:
48          database:
49            condition: service_healthy
50
51
52      networks:
53        web:
54          driver: bridge

```

Mindezek után elmentettem, átnéztem mindent, hogy jól írtam-e, majd megnyitottam a parancssort. Ott elnavigáltam a webapp főmappáig. Itt a docker-compose up parancsot kiadva lefut a dockercompose.yaml fájl tartalma. A 3 konténerhez szükséges adatok letöltődnek, majd elindulnak.

Miután elindultak a konténerek úgy ellenőriztem a konténerek működését, hogy elnavigáltam a localhost:80 –as oldalra, amelyen a következő látható:



id	name	neptuncode	age
2	Gipsz Jakab	ABC123	55
3	teszt	neptun	99