

Miskolci Egyetem

Gépészmérnöki és Informatikai Kar

Általános Informatikai Intézeti Tanszék



Rendszerüzemeltetés 3

MSSQL szerver és webszerver kommunikációja konténerizált környezetben

Készítette:

Név: Kiss Bálint

Neptunkód: NXO2PW

Szak: Mérnökinformatikus BSc

Informatikai rendszermérnök szakirány

Bevezetés

A feladat megvalósítása során lokális számítógépen, Docker konténerizált környezetben kerül kialakításra Microsoft SQL, és webszerver. A dokumentum tartalmazza minden szükséges szoftver letöltésének és telepítésének menetét, valamint a teljes implementációt, ideértve az adatbázisok és webszerverek konfigurációját

1. Hozzunk létre egy új mappát, melyben minden szükséges szolgáltatást el fogunk helyezni. A neve legyen a következő: APP
2. Az APP mappán belül hozzunk létre egy almappát BACKEND néven, melyben a MySQL komponensst fogjuk elhelyezni.
3. Az APP mappán belül hozzunk létre egy almappát FRONTEND néven, melyben a webszerver komponensst fogjuk elhelyezni.
4. A BACKEND mappában hozzunk létre egy fájlt index.php néven, melynek a tartalma a következő.

```
<html>
<head>
  <title>Selecting Records</title>
</head>
<body>
  <h1>Table:</h1>
  <?php
    $dbhost = $_ENV["MYSQL_HOST"];
    $dbuser = $_ENV["MYSQL_USER"];
    $dbpass = $_ENV["MYSQL_PASSWORD"];
    $database = $_ENV["MYSQL_DATABASE"];
    $conn = mysqli_connect($dbhost, $dbuser, $dbpass, $database);

    if(! $conn ) {
      die('Could not connect: ' . mysqli_error($conn));
    }

    mysqli_select_db( $conn, 'db' );
    $sql = "SELECT * FROM student";
    $retval = mysqli_query( $conn, $sql );
    if(! $retval ) {
      die('Could not get data: ' . mysqli_error($conn));
    }
    echo "<table class='\"table table-striped\"' ><tr><th>id</th><th>name</th><th>neptuncode</th><th>age</th></tr>";
    while($row = mysqli_fetch_array($retval)) {
      echo "<tr>";
      echo "<td>" . $row['id'] . "</td>";
      echo "<td>" . $row['name'] . "</td>";
      echo "<td>" . $row['neptuncode'] . "</td>";
      echo "<td>" . $row['age'] . "</td>";
      echo "</tr>";
    }
    echo "</table>";

    mysqli_close($conn);
  ?>
</body>
</html>
```

5. Ezután szintén a BACKEND mappában hozzunk létre egy Dockerfile fájlt, melynek a tartalma a következő.

```
FROM php:7.2-apache

ENV MYSQL_HOST=
ENV MYSQL_USER=
ENV MYSQL_PASSWORD=
ENV MYSQL_DATABASE=

RUN docker-php-ext-install mysqli && docker-php-ext-enable mysqli

RUN apt-get update

EXPOSE 80

COPY . /var/www/html/
```

6. A FRONTEND mappában hozzunk létre egy commands.sql nevű fájlt, melynek a tartalma a következő.

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

CREATE DATABASE IF NOT EXISTS `db` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;
USE `db`;

CREATE TABLE `student` (
  `id` int NOT NULL,
  `name` varchar(255) DEFAULT NULL,
  `neptuncode` varchar(6) DEFAULT NULL,
  `age` int DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3 COLLATE=utf8_hungarian_ci;;

INSERT INTO student VALUES(2, 'Gipsz Jakab', 'ABC123', 55 );
INSERT INTO student VALUES(3, 'teszt', 'neptun', 99 );

COMMIT;
```

7. Az APP mappában hozzunk létre egy .env nevű fájlt, mely a szükséges változókat fogja tartalmazni. A fájl tartalma a következő.

```
MYSQL_USER=web_user
MYSQL_PASSWORD=passw0rd
MYSQL_ROOT_PASSWORD=passw0rd
MYSQL_DATABASE=db

MYSQL_HOST=database
```

8. Szintén az APP mappában hozzunk létre egy docker-compose.yaml nevű fájlt, mely a konténer felépítését fogja tartalmazni. A fájl tartalma a következő.

```
version: "3.0"


services:
  webserver:
    build:
      context: ./site
      dockerfile: Dockerfile
    image: beadando:1.0
    environment:
      - MYSQL_USER=${MYSQL_USER}
      - MYSQL_PASSWORD=${MYSQL_PASSWORD}
      - MYSQL_DATABASE=${MYSQL_DATABASE}
      - MYSQL_HOST=${MYSQL_HOST}
    ports:
      - 80:80
    networks:
      - web
    depends_on:
      database:
        condition: service_healthy

  database:
    image: mysql
    command: --default-authentication-plugin=mysql_native_password
    volumes:
      - ./sql:/docker-entrypoint-initdb.d
    environment:
      - MYSQL_USER=${MYSQL_USER}
      - MYSQL_PASSWORD=${MYSQL_PASSWORD}
      - MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD}
      - MYSQL_DATABASE=${MYSQL_DATABASE}
    ports:
      - 3306:3306
    networks:
      - web
    healthcheck:
      test: ["CMD", "mysqladmin" ,"ping", "-h", "localhost"]
      timeout: 3s
      retries: 10

  adminer:
    image: adminer
    networks:
      - web
    ports:
      - 127.0.0.1:8080:8080
    depends_on:
      database:
        condition: service_healthy

networks:
  web:
    driver: bridge
```

9. A parancssorban navigáljunk az APP nevű mappába, majd adjuk ki a *docker-compose up* parancs kiadása után lefut a docker-compose.yaml fájl tartalma, majd megtörténik a konténerekhez szükséges adatok letöltése, és a konténerek elindulnak.
10. Nyissuk meg a böngészőnkbe a localhost oldalt és ellenőrizzük, hogy minden megfelelően működik-e.

A screenshot of a web browser window with the address bar showing 'localhost'. The page content displays a table with the title 'Table:'. The table has four columns: 'id', 'name', 'neptuncode', and 'age'. There are two data rows. The first row has values: '2', 'Gipsz Jakab', 'ABC123', and '55'. The second row has values: '3', 'teszt', 'neptun', and '99'.

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