



UNIVERZITET U ZENICI
POLITEHNIČKI FAKULTET
Softversko inženjerstvo



Projektna dokumentacija Operativni sistemi

Tema: Multi-database server uz pomoć Dockera

Članovi tima:

1. Faris Ibrišević
2. Nedim Bašić
3. Hamza Pašanović
4. Kemal Hasanspahić

Mentor: Adin Jahić

Sadržaj

I UVOD I OPIS PROJEKTA	3
II Objašnjenje pojmova.....	4
II.I dockerfile	4
II.II docker-compose.....	4
II.III YAML fajl	5
III. OPIS RADA PROJEKTA	6
IV. ZAKLJUČAK	9

I UVOD I OPIS PROJEKTA

Cilj projekta je podesiti multi-database server koji, kako mu samo ime kaže, koristi više baza podataka.

Multi-database server se koristi za upravljanje sa više baza podataka istovremeno. Ovo je korisno u situacijama kada organizacija ima različite aplikacije ili procese koji koriste različite baze podataka, a želi da centralizuje njihovo upravljanje radi lakšeg održavanja, skaliranja i bolje sigurnosti. Multi-database server omogućava pristup i upravljanje više baza podataka putem jedinstvenog interfejsa. To znači da se podaci mogu lako prenositi između baza podataka i korisniku je omogućen pristup svim podacima koje organizacija poseduje, bez obzira na to u kojoj se bazi podataka nalaze. Takođe, multi-database server omogućava organizacijama da upravljaju velikim količinama podataka na efikasan način, što je posebno korisno u situacijama kada se velike količine podataka obrađuju u realnom vremenu. Konačno, multi-database server može pomoći u smanjenju troškova, jer organizacije ne moraju da kupuju različite softvere i hardvere za upravljanje različitim bazama podataka, već mogu da koriste jedan centralni server.

II Objašnjenje pojmova

Kroz dokumentaciju spominjat će se dosta pojmovi kao što su *dockerfile* i *docker compose*. Ovo poglavlje projektne dokumentacije objašnjava te pojmove.

II.I dockerfile

Dockerfile je tekst koji se koristi za definisanje docker containera. On sadrži sve potrebne korake i upute koje će docker platforma koristiti kako bi se kreirao container. U njemu definišemo osnovnu sliku (base image) koju ćemo koristiti, dodajemo sve potrebne “dependencies” i slične stvari. To je osnovni alat za izgradnju docker slike i definira kako se aplikacija ili usluga moraju konfigurisati, instalirati i pokrenuti unutar docker container-a.

U dockerfile se mogu navesti sljedeće stvari:

- Base image – koristi se za izgradnju docker slike.
- Komande za instalaciju aplikacija i paketa
- Config postavke
- Dodavanje nekog koda u docker image
- Pokretanje naredbi za pokretanje aplikacije ili usluge

Nakon što je file napisan, koristi se komanda *docker build* za izgradnju docker slike iz dockerfile-a. Laički rečeno, dockerfile se može smatrati receptom za izgradnju docker slike. Docker slika, po istoj analogiji, je proizvod koji se dobija iz tog recepta.

II.II docker-compose

Docker compose je alat za definisanje i pokretanje docker aplikacija sa više kontejnera. Ovaj alat vam omogućava da definišete vaše aplikacije, servise, mreže i volumene u jednom fajlu, I pokrenete I zaustavite sve kontejnere u vašoj aplikaciji s jednom komadnom.

Neke ključne karakteristike docker compose-a:

- Jednostavna konfiguracija – svi servisi, volumeni i mreže se definišu u jednom YAML fajlu, što olakšava konfigurisanje i održavanje vaše aplikacije.
- Aplikacija s više kontejnera – docker compose je dizajniran da pokreće aplikacije s više kontejnera, tako da možete definisati i upravljati svim kontejnerima koji čine vašu aplikaciju na jednom mjestu
- Skalabilnost – Lahko je skalirati aplikaciju mijenjajući broj kontejnera za svaki servis

Sveukupno, docker compose je moćan alat koji pojednostavljuje proces implementacije i upravljanja aplikacijama sa više korisnika.

II.III YAML fajl

YAML je ljudski čitljiv jezik. Najčešće se koristi za konfiguracijske fajlove i u svrhe gdje se podaci skladište ili prenose. YAML “cilja” mnoge komunikacijske aplikacije kao i XML, ali ima minimalnu sintaksu koja ga razlikuje od SGML-a. YAML datoteke se obično sastoje od niza parova zvanih “ključ-vrijednost” pri čemu su vrijednosti često nizovi ili objekti koji sadrže druge parove “ključ-vrijednost”. Primjer YAML fajl-a ispod:

```
ime: Faris
prezime: Ibrišević
godine: 20
adrese:
  - grad: Zenica
    ulica: Bulevar Kralja Tvrtka I
  - grad: Sarajevo
    ulica: Skenderija 22
```

Ovaj fajl opisuje osobu s imenom “Faris Ibrišević” koja ima 20 godina i dvije adrese – jednu u Zenici i jednu u Sarajevu. Svaka adresa je opisana kao objekat koji ima “ključeve” “grad” i “ulica”. Ključevi su označeni sa dvotačkom, a vrijednosti se navode nakon toga. Znak minus “-“ ispred adresa označava da su one dio niza vrijednosti koje pripadaju ključu “adrese”.

III. OPIS RADA PROJEKTA

Da bi se najlakše shvatio projekat, potrebno je krenuti ispočetka. Ovdje bi početak značio pravljenje container-a za svaku od baza podataka. I to je barem lahko uraditi. Za svaki container ide jedna komanda, sa različitim parametrima, i to je to.

```
docker run --name my-sqlserver-container -e 'ACCEPT_EULA=Y' -e  
'SA_PASSWORD=MyStrongPassword123' -p 1433:1433 -d  
mcr.microsoft.com/mssql/server:latest
```

--name my-sqlserver-container – je ime našeg docker containera.

-e su environment varijable, u ovom slučaju se prihvata EULA i podešava se šifra sa “sa” user-a. Šifra je u ovom slučaju “MyStrongPassword123”.

-p je broj porta koji se koristi, u našem slučaju 1433 port, odnosno TCP/UDP port

-d znači detached, odnosno da radi u pozadini

mcr.microsoft.com/mssql/server:latest – “povlačenje” slike sa Docker Hub-a

Ovo je MSSQL baza podataka. Njen DBMS je SSMS, odnosno SQL Server Management Studio.

Pošto mi imamo tri baze podataka, od kojih svaka koristi različit DBMS, to znači pisanje tri komande.

Ali, šta ako dođe do nekog upada na server, i neko nam ugasi sve docker container-e (nerealan scenarij, napadač će uraditi mnogo više od toga da samo ugasi containere)? To znači da mi tri puta moramo kucati “*docker up <ime containera ili njegov ID>*”. Odnosno, moramo kucati tu komandu onoliko puta, koliko imamo containera. Ovdje do izražaja dolazi docker compose. Docker compose nam olakšava prethodno spomenuto “podizanje” containera. Princip rada je veoma jednostavan.

Pisanje docker-compose.yaml fajla. Unutar tog yaml fajla, pišemo ono što nam treba. Laički, docker-compose je više dockerfile-ova spojen u jedan. Kada smo zadovoljni svojim docker-compose-om, pokrećemo ga komandom *docker-compose up*. Ako ih želimo ugastiti, *docker-compose stop*.

Primjer docker-compose.yaml fajla za našu svrhu:

```
version: '3.9'
```

```
services:
```

```
  mssql:
```

```
    image: mcr.microsoft.com/mssql/server:2019-latest
```

```
    container_name: mssql-1
```

```
    environment:
```

```
      SA_PASSWORD: MyStrongPassword123
```

```
      ACCEPT_EULA: Y
```

```
    ports:
```

```
      - "1433:1433"
```

```
  mysql:
```

```
    image: mysql:8.0
```

```
    container_name: mysql-compose
```

```
    environment:
```

```
      MYSQL_ROOT_PASSWORD: admin123!
```

```
      MYSQL_DATABASE: mydatabase
```

```
    ports:
```

```
      - "3306:3306"
```

```
  postgres:
```

```
    image: postgres:13
```

```
    container_name: postgres-1
```

```
    environment:
```

```
      POSTGRES_PASSWORD: admin123!
```

```
      POSTGRES_USER: postgres
```

```
      POSTGRES_DB: mydatabase
```

```
    ports:
```

```
      - "5432:5432"
```

Komande i naredbe su malo drugačije nego u komandama, ili u dockerfile. Proći ćemo kroz neke od njih

image – “povlači” sliku sa docker hub-a;

container_name – daje ime container

environment – određuje prethodno spomenute environment varijable

SA_PASSWORD: MyStrongPassword123 – Šifra za korisnika “sa”

ACCEPT_EULA: Y – Prihvatanje EULA zahtjeva

ports:

- "1433:1433" – Otvaranje određenog porta

Jedina razlika jest u sintaksi same baze, odnosno baš u ovim environment variables. Za postgres se određuje i username, sa prefiksom POSTGRES ispred svake varijable. Unutar MySQL-a, default username je definisan kao root. Prefiks koji ide ispred svake varijable je MYSQL_<sintaksa>.

Server smo “hostovali” na DigitalOcean VM-u. Spajanje na baze podataka se vrši preko njihovih DBMS-ova, odnosno Database Management System.

Za MSSQL se koristi SQL Server Management Studio, za MySQL se koristi MySQL Workbench, a za PSQL se koristi pgAdmin.

IV. ZAKLJUČAK

Ukratko, ovaj projekat se mogao realizovati na više načina, ali naglasak je bio na docker-compose radi lakšeg upravljanja kontejnerima. Projekat se može pokrenuti i lokalno (da host bude localhost), ili preko nekog vanjskog servera (u kojem se slučaju kao hostname koristi javna IP adresa tog servera). Multi-database server se koristi za više baza, kako samo ime kaže. Više baza se koristi iz razloga što ne podržavaju sve aplikacije, programi ili web stranice istu bazu podataka. Na taj način, smanjuju se troškovi zakupljenja ili kupovine novih servera. Način na koji se izbjegava spajanje na istu bazu jesu upravo portovi koji su definisani u docker-compose.