به نام خدا

محمد امین روح بخش مجرد مستندات پروژه نهایی کنگرو

فاز اول پروژه

۱- ابتدا باید فایل docker-compose.yml را در مسیر پروژه ایجاد کنیم. سپس درون آن سرویس های nginx و wordpress1 , 2 و دیتابیس Mariadb را بالا میاوریم.
۲- در ابتدای فایل version را اضافه میکنیم ،در این قسمت میگوییم که میخواهیم از کدام ورژن داکر انجین استفاده کنیم.
۳- سپس با استفاده از تگ Services ، سرویس های مورد نیازمان را تعریف میکنیم و برای هر کدام از سرویس ها ، image رسمی آن را که از داکرهاب با تگ دلخواه میتوان pool کرد ، قرار میدهیم.
۴- سپس کانفیگ های مورد نیاز را طبق عکس زیر انجام میدهیم. توجه داشته باشید که باید یک volume و networks هم برای سرویس هایمان ایجاد کنیم.

در این قسمت محتوای فایل docekr-compose.ymlرا کامل میکنیم.

```yaml
version: '3'

services:
  wordpress1:
    image: wordpress
    restart: always
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_NAME: wordpress
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: password
    volumes:
      - wordpress1_data:/var/www/html
    networks:
      - wordpress_network

  wordpress2:
    image: wordpress
    restart: always
    environment:
      WORDPRESS_DB_HOST: db
```

```
      WORDPRESS_DB_NAME: wordpress
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: password
    volumes:
      - wordpress2_data:/var/www/html
    networks:
      - wordpress_network

  db:
    image: mariadb
    restart: always
    environment:
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: password
      MYSQL_ROOT_PASSWORD: root_password
    volumes:
      - db_data:/var/lib/mysql
    networks:
      - wordpress_network

  nginx:
    image: nginx
    restart: always
    ports:
      - "80:80"
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf:ro
    depends_on:
      - wordpress1
      - wordpress2
    networks:
      - wordpress_network

networks:
  wordpress_network:

volumes:
  wordpress1_data:
  wordpress2_data:
  db_data:
```

services: این بخش شامل تعریف سرویس‌های وردپرس (wordpress1 و wordpress2)، دیتابیس MariaDB (db) و لود بالانسر Nginx (nginx) است.

volumes: در اینجا اسم‌هایی که برای ورودی‌های داده‌ای مورد نیاز تعریف شده‌اند (wordpress1_data، wordpress2_data، db_data).

networks: شامل تعریف شبکه‌ای است که سرویس‌ها از آن استفاده می‌کنند (wordpress_network).

۵-ایجاد فایل nginx.conf

حال وقت آن رسیده که فایل nginx.conf را در مسیر nginx ایجاد کنیم و کانفیگ های لازم را درآن قرار دهیم . نمونه ای از کانفیگ های لازم برای این فایل:

```
events {}

http {
    upstream wordpress {
        server wordpress1:80;
        server wordpress2:80;
    }

    server {
        listen 80;

        location / {
            proxy_pass http://wordpress;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
        }
    }
}
```

در اینجا برای لود بالانسینگ میان دو سرویس وردپرس ، باید نام هر سرویس وردپرس را در قسمت upstream بنویسیم و به آن پورت 80 که توسط سرویس nginx اکسپوس شده است ، بدهیم.

سپس در قسمت location با proxy_pass درخواستمان را به اسم upstream که میتواند دلبخواه باشد میدهیم.

upstream: در اینجا تمام سرورهای وردپرس به عنوان یک upstream به نام wordpress تعریف شده‌اند.

server: یک سرور Nginx با گوش دادن به درگاه ۸۰ تعریف شده است که درخواست‌ها را به upstream wordpress ارسال می‌کند.

مراحل اجرا:

۱. ابتدا فایل‌های docker-compose.yml و nginx.conf را ایجاد یا تغییر دهید.

۲. با استفاده از دستور docker-compose up -d، پروژه را اجرا کنید.

۳. سپس از مرورگر خود به http://localhost:80 بروید تا وردپرس اجرا شده را مشاهده کنید.

۴. با هر بار رفرش کردن و استفاده از دستور docker compose logs میتوانید ببینید که درخواست از سمت وردپرس ۱ است یا وردپرس ۲ .

این مراحل به شما کمک می‌کند تا این پروژه راه‌اندازی شود و با استفاده از Docker و Docker Compose به سادگی قابل مدیریت باشد.

فاز دوم پروژه

پیکربندی سرویس Mariadb

در این قسمت ما باید به فایل docker-compose.yml یک سرویس دیتابیس دیگر به نام replica-db اضافه کنیم و همان کانفیگ های قبلی که برای دیتابیس ماریا دی بی یا سرویس db انجام دادیم ، برای این سرویس هم انجام دهیم .

توجه داشته باشید که volume های این دو دیتابیس باید از یکدیگر جدا باشد.

در صورتی که مشکلاتی در ارتباط با دیتابیس داشتید ، به هر کدام از سرویس های دیتابیس یک پورت بدهید . برای مثال برای master-db از پورت3306 و برای replica-db از پورت 3307 .

البته توجه داشته باشید که این دو پورت در هاست خودمان متفاوت هستند ، اما هنگام اکسپوس کردن این دو پورت ، باید هر دو پورت را به پورت 3306 یا 33060 اکسپوس کنیم تا از ارور های احتمالی جلوگیری کنیم.

حالا لازم است به مسیر Amin_roohbakhsh_cangrow/masterdb بروید و در انجا با استفاده از از دستور initial.sh/. فایل بش را اجرا کنید و کمی صبر کنید تا کانیتنر ها توسط docker compose ساخته شوند.

توجه داشته باشید که فایل initial.sh حتما دسترسی اجرا را داشته باشد.

اگر نداشت ، با استفاده از دستور chmod +x initial.sh ، به این فایل دسترسی اجرایی بدهید و آن را اجرا کنید.

حال که دستور اجرا شد، با استفاده از دستور زیر :

```
docker exec –it master-db mariadb –u root –p'somewordpress' –e "CREATE DATABASE [your database name]"
```

با user روت به دیتابیس master متصل شوید و یک دیتابیس بسازید.

برای اینکه ببینید دیتابیس شما ساخته شده است یا نه از دستور زیر استفاده کنید :

```
docker exec –it master-db mariadb –u root –p'somewordpress' –e "SHOW DATABASES"
```

حال مانند قبل ما با user که ساختیم به دیتابیس replica وصل میشویم یعنی به جای master-db از replica-db استفاده میکنیم.

برای اینکه ببینیم آیا دیتابیس ساخته شده در سرویس دیتابیس master به دیتابیس replica متصل است یا نه ، کافی است با دستورSHOW DATABASES دیتابیس های موجود در دیتابیس replica را مشاهده کنیم.

چناچه دیتابیسی که در master ساخته بودیم در اینجا هم نمایش داده شود ، میتوانیم پی ببریم که دیتابیس replica ما با دیتابیس master متصل است و چنانچه دیتابیس master پایین بیاید یا قطع شود ، همچنان دیتابیس replica ما ران است و اطلاعات را به خوبی ذخیره میکند.