Practical – 5

docker login docker.io
docker compose up -d --build

docker-compose.yml

```
services:
  db:
    image: mysql:8.0
    container_name: ecommerce-db1
    restart: always
    environment:
      MYSQL_DATABASE: ecommerce
      MYSQL_ROOT_PASSWORD: root
    ports:
      - "3307:3306"
    volumes:
      - db_data:/var/lib/mysql
    networks:
      - ecommerce-net
    healthcheck:
      test: ["CMD", "mysqladmin" ,"ping", "-h", "localhost", "-uroot", "-proot"]
      interval: 10s
      timeout: 5s
      retries: 5

  backend:
    build:
      context: .
      dockerfile: Dockerfile.backend
    container_name: ecommerce-backend1
    restart: always
    environment:
      SPRING_DATASOURCE_URL: jdbc:mysql://db:3306/ecommerce
      SPRING_DATASOURCE_USERNAME: root
      SPRING_DATASOURCE_PASSWORD: root
      SPRING_JPA_HIBERNATE_DDL_AUTO: update
    depends_on:
      db:
        condition: service_healthy # wait for MySQL to pass healthcheck
    ports:
      - "8083:8080"
    networks:
      - ecommerce-net
```

```
  frontend:
    build:
      context: .
      dockerfile: Dockerfile.frontend
    container_name: ecommerce-frontend1
    restart: always
    ports:
      - "8082:8080"
    depends_on:
      - backend
    networks:
      - ecommerce-net

volumes:
  db_data:

networks:
  ecommerce-net:
    driver: bridge
```