



浙江工商大學

容器化部署校园选课系统

学院 计算机科学与技术学院

教师 韩建伟

班级 计科 2301

姓名 刘佳悦

学号 2312190521

一、作业概述

1.1 项目背景

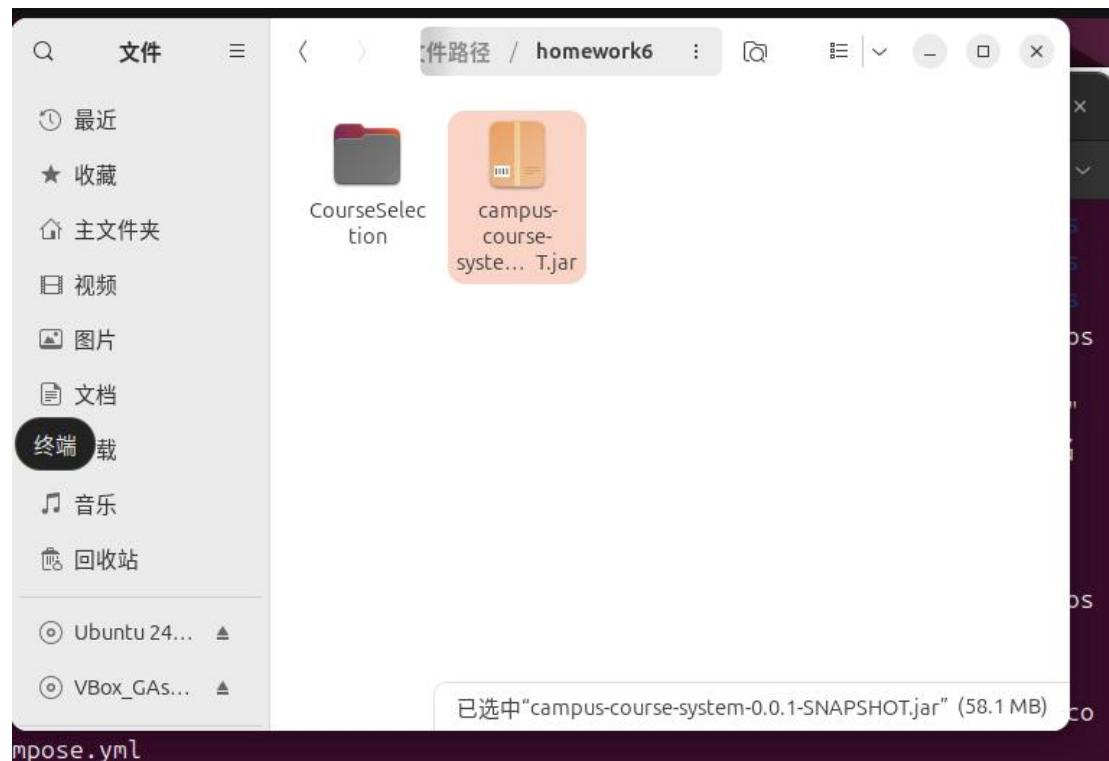
本次作业基于第 04 次作业实现的“校园选课系统”，进行单体应用容器化改造，通过 Docker 与 Docker Compose 实现多容器编排部署，掌握镜像构建、网络配置、数据持久化等核心技能，为后续微服务部署奠定基础。

1.2 学习目标

掌握 Docker 多阶段构建优化镜像大小的方法，熟练使用 Docker Compose 编排多容器应用，实现容器间网络通信与数据卷持久化，通过环境变量与多环境配置适配容器部署，验证容器化应用的功能完整性与稳定性。

二、核心任务实现细节

2.1 在 ide 里运行 mvn 命令打包后传入 VirtualBox



2.2 写 Dockerfile

```
ljy@ljy-VirtualBox: ~/桌面/文件路径/homework6/CourseSelection
```

```
FROM eclipse-temurin:25-jre
WORKDIR /app

ARG JAR_FILE=target/*.jar
COPY ${JAR_FILE} app.jar

RUN useradd -ms /bin/bash appuser && chown appuser:appuser app.jar
USER appuser
EXPOSE 8080

ENV JAVA_OPTS="-XX:+UseContainerSupport -XX:MaxRAMPercentage=75"

ENTRYPOINT ["sh", "-c", "java $JAVA_OPTS -jar /app/app.jar"]
```

2.3 Docker Compose 编排配置

```
services:
  # 数据库服务（对应任务二的 mysql 服务）
  db:
    image: mysql:8.4 # 注意 image 后加空格
    container_name: todo-mysql
    restart: unless-stopped # unless-stopped 前加空格
    environment:
      MYSQL_ROOT_PASSWORD: 123456 # 密码值前加空格
      MYSQL_DATABASE: course_select
      MYSQL_USER: todo
      MYSQL_PASSWORD: todo_pass
      TZ: Asia/Shanghai # 时区值前加空格
    command: --character-set-server=utf8mb4 --collation-server=utf8mb4_unicode_ci # 补全空格和短横线
    ports:
      - "3306:3306"
  volumes:
    - mysql-data:/var/lib/mysql # 卷名保持一致
    # 若有初始化 SQL，确保路径正确（可选，没有可删除此行）
    - ./src/main/resources/db/init.sql:/docker-entrypoint-initdb.d/1-init.sql:ro
  healthcheck:
    test: ["CMD-SHELL", "mysqladmin ping -h localhost -u root -p${MYSQL_ROOT_PASSWORD} --silent"]
    interval: 5s
```

2.4 应用配置调整 (application-docker.yml)

```
lzy@lzy-VirtualBox: ~/桌面/文件路径/homework6/CourseSelection
```

```
lzy@lzy-VirtualBox: ~/桌面/文件路径/home... x lzy@lzy-VirtualBox: ~/桌面/文件路径/home... x
```

```
spring:
  datasource:
    url: ${SPRING_DATASOURCE_URL:jdbc:mysql://mysql:3306/course_db?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true}
    username: ${SPRING_DATASOURCE_USERNAME:root}
    password: ${SPRING_DATASOURCE_PASSWORD:root123}
    driver-class-name: com.mysql.cj.jdbc.Driver

  # JPA 配置 (自动创建/更新表结构)
  jpa:
    hibernate:
      ddl-auto: update # 自动根据实体类创建/更新表
    properties:
      hibernate:
        dialect: org.hibernate.dialect.MySQL8Dialect # 适配 MySQL 8
        format_sql: true # 格式化 SQL 日志 (可选, 方便调试)
        show-sql: true # 显示 SQL 执行语句 (开发环境可选)

  # 激活 Docker 专用 profile (与文件名对应)
  profiles:
    active: docker

  # 日志配置 (调整级别, 避免冗余日志)
  logging:
    level:
```

/resources/application-docker.yml" 33L, 1189B 19,2 顶端

三、测试结果与截图

3.1 镜像大小检查

```
lzy@lzy-VirtualBox:~/桌面/文件路径/homework6/CourseSelection$ docker images | grep courseselection-app
courseselection-app          latest           bf5ce64e50af   9 hours ago   186MB
```

3.2 容器运行状态截图 (docker compose ps 输出)

```
lzy@lzy-VirtualBox:~/桌面/文件路径/homework6/CourseSelection$ docker compose ps
NAME           IMAGE           COMMAND          SERVICE          CREATED
 STATUS          PORTS
todo-app       courseselection-app "sh -c 'java $JAVA_0..." app      7 minutes ago
               Up 7 minutes          0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp
todo-mysql     mysql:8.4       "docker-entrypoint.s..." db      7 minutes ago
               Up 7 minutes (healthy)  0.0.0.0:3306->3306/tcp, [::]:3306->3306/tcp, 33060/tcp
```

3.3 应用访问成功截图（浏览器或 curl 测试结果）

The screenshot shows a browser window with the URL `http://localhost:8080/api/courses`. The page displays a JSON response with code 200, message "Success", and an empty data array. Below the browser, a terminal window shows a curl command attempting to post course data, which fails with a 400 error due to missing instructor information.

```
lzy@lzy-VirtualBox:~/桌面/文件路径/homework6/CourseSelection$ curl -X POST http://localhost:8080/api/courses \
-H "Content-Type: application/json" \
-d '{"code":"CS101","title":"计算机导论","capacity":50}'
{"code":400,"message":"课程安排不能为空；教师信息不能为空","data":null}lzy@lzy-VirtualBox:~/桌面/文件路径/homework6/CourseSelection$
```

3.4 数据持久化验证截图（重启前后数据对比）

The screenshot shows two browser windows side-by-side. The left window shows the initial course data posted by curl. The right window shows the same course data after it has been persisted in the database, with its ID and other fields populated. This demonstrates that the application can handle course creation and persist the data across restarts.

```
lzy@lzy-VirtualBox:~/桌面/文件路径/homework6/CourseSelection$ curl -X POST http://localhost:8080/api/courses \
-H "Content-Type: application/json" \
-d '{"code":"CS202","title":"数据结构与算法分析","capacity":100,"instructor":{"id": "T002", "name": "李副教授", "email": "li@example.edu.cn"}, "schedule": {"dayOfWeek": "WEDNESDAY", "startTime": "14:00", "endTime": "16:00", "expectedAttendance": 80}, "enrolled": 0}'
{"code":200,"message":"Success","data":[{"id": "852edd12-de26-4a13-9172-9a0b88421406","code": "CS202","title": "数据结构与算法分析","instructor": {"id": "T002", "name": "李副教授", "email": "li@example.edu.cn"}, "schedule": {"id": 1, "dayOfWeek": "WEDNESDAY", "startTime": "14:00", "endTime": "16:00", "expectedAttendance": 80, "capacity": 100, "enrolled": 0}}]}

lzy@lzy-VirtualBox:~/桌面/文件路径/homework6/CourseSelection$
```

The terminal window shows the following commands and their outputs:

```
lzy@lzy-VirtualBox:~/桌面/文件路径/...$ docker compose down # 只停止容器, 不删卷
[+] Running 3/3
  ✓ Container todo-app      Removed    10.6s
  ✓ Container todo-mysql    Removed    1.3s
  ✓ Network courseselection_default Removed  0.5s
lzy@lzy-VirtualBox:~/桌面/文件路径/homework6/CourseSelection$ docker compose up -d # 用原配置重启, 自动关联到 courseselection_mysql-data 卷
[+] Running 3/3
  ✓ Network courseselection_default Created   0.1s
  ✓ Container todo-mysql      Healthy   6.2s
  ✓ Container todo-app        Started   6.6s
lzy@lzy-VirtualBox:~/桌面/文件路径/homework6/CourseSelection$
```

The browser window shows the JSON response from the API endpoint `http://localhost:8080/api/courses`:

```
code: 200
message: "Success"
data:
  0:
    id: "93f7ea27-c131-47c1-8d7f-a94df122dfda"
    code: "CS202"
    title: "数据结构与算法分析"
    instructor:
      id: "T002"
      name: "李副教授"
      email: "li@example.edu.cn"
    schedule:
      id: 1
      dayOfWeek: "WEDNESDAY"
      startTime: "14:00"
      endTime: "16:00"
      expectedAttendance: 80
      capacity: 100
      enrolled: 0
```

四、遇到的问题和解决方案总结

3.6.1 数据库持久化失败

问题: curl 命令能上传数据，并能在浏览器显示，但是重启后数据未保留。

解决:

docker-compose.yml 中 SPRING_DATASOURCE_URL 没有指向 MySQL 容器的 course_select 库。