实现思路

- 后端镜像（Python 3.8-slim）
  - 安装 mysqlclient 所需系统依赖（build-essential、default-libmysqlclient-dev 等）和 pip依赖。
  - CMD： makemigrations → migrate →(only with INIT_DB env at first start) init_db → Gunicorn
  - 生产配置 settings_prod.py 中数据库指向 mysql 容器，密码读取 `MYSQL_ROOT_PASSWORD` 环境变量。
- nginx 镜像
  - 构建：用 Node 构建前端（npm run build），将 build 产物拷贝进 nginx:latest。
  - 配置 app.conf：静态资源根目录 /usr/share/nginx/html，/api/v1/ 反向代理到 `http://app:8000/api/v1/`，SPA 路由回退到 index.html；
- MySQL 服务（8.1）
  - 环境变量：`MYSQL_ROOT_PASSWORD=xxx`（来自 .env）、`MYSQL_DATABASE=thss`、`TZ=Asia/Shanghai`。
  - 启动参数：utf8mb4 字符集、utf8mb4_unicode_ci 排序规则。
  - Healthcheck：mysqladmin ping，间隔 10s，超时 5s，重试 5 次。
  - 挂载：`/home/ubuntu/mysql:/var/lib/mysql`。
- Compose 编排
  - 容器名固定：app、nginx、mysql。
  - 网络：frontend_net（nginx+app）、backend_net（app+mysql）。
  - 仅 nginx 暴露 8000:8000；app 不暴露；mysql 不对宿主机暴露。
  - 后端 `depends_on: mysql: service_healthy`，避免“抢跑”。

踩坑

- mysqlclient 编译依赖缺失会导致 pip 安装失败，需在 Python 镜像中先装好 default-libmysqlclient-dev 和编译工具。
- 容器名即 DNS 名：nginx 里用 `app:8000` 反代，而不是 127.0.0.1。