

图片管理网站 — 项目搭建指南

1. 环境要求

- 操作系统: Linux / macOS / Windows
- Docker: 20.10+
- Docker Compose: 2.0+

```
# 检查 Docker 版本
docker --version

# 检查 Docker Compose 版本
docker compose version
```

2. 项目获取

```
# 克隆项目
git clone https://github.com/your-repo/photo-manager.git
cd photo-manager
```

3. Docker 部署

3.1 配置环境变量

```
# 复制环境变量示例文件
cp .env.example .env

# 编辑配置（可选，使用默认配置即可运行）
vim .env
```

.env 主要配置项:

变量	说明	默认值
SECRET_KEY	Django 密钥	自动生成
DEBUG	调试模式	False
CELERY_BROKER_URL	Celery 消息代理	redis://redis:6379/0

3.2 构建并启动服务

```
# 构建并启动所有服务
docker-compose up -d --build

# 查看服务状态
docker-compose ps

# 查看日志
docker-compose logs -f
```

3.3 服务说明

服务	端口	说明
redis	6379	消息队列
backend	8000	Django 后端 API
celery-worker	-	异步任务处理（缩略图、EXIF）
celery-beat	-	定时任务调度
frontend	3000	Vue 前端页面

3.4 访问系统

- 前端页面: <http://localhost:3000>
 - 后端 API: <http://localhost:8000>
 - 管理后台: <http://localhost:8000/admin/>
-

4. 创建管理员账号

```
# 进入后端容器创建超级用户
docker exec -it photo-manager-backend python manage.py createsuperuser
```

按提示输入邮箱、用户名和密码即可。

5. 常用命令

5.1 服务管理

```
# 停止所有服务
docker-compose down

# 重启服务
docker-compose restart

# 停止并删除数据卷（清空数据）
docker-compose down -v
```

5.2 查看日志

```
# 查看所有服务日志
docker-compose logs -f

# 查看特定服务日志
docker-compose logs -f backend
docker-compose logs -f celery-worker
```

5.3 数据库操作

```
# 执行数据库迁移
docker exec photo-manager-backend python manage.py migrate

# 收集静态文件
docker exec photo-manager-backend python manage.py collectstatic --noinput
```

6. 验证部署

6.1 检查服务状态

```
docker-compose ps
```

所有服务状态应为 `Up` 或 `running`。

6.2 功能测试

1. 访问 <http://localhost:3000>
 2. 注册新用户账号
 3. 登录系统
 4. 上传一张图片
 5. 等待几秒后刷新，检查缩略图是否生成
 6. 查看图片详情页的 EXIF 信息
-

7. 常见问题

Q1: 端口被占用

```
# 检查端口占用
lsof -i :3000
lsof -i :8000

# 修改 docker-compose.yml 中的端口映射
```

Q2: 容器启动失败

```
# 查看详细日志
docker-compose logs backend

# 重新构建
docker-compose up -d --build --force-recreate
```

Q3: 图片上传后缩略图没有生成

```
# 检查 Celery Worker 状态
```

```
docker-compose logs celery-worker
```

```
# 确认 Redis 正常运行
```

```
docker exec photo-manager-redis redis-cli ping
```