开发文档 -- Docker 部署

前端，后端，mqtt服务器，mysql分别用一个dockerfile来构建，然后用docker-compose来一键构建，部署运行。并且把端口映射到本机上。

Docker-compose

```
version: '3.8'

services:
  iot-mysql:
    build:
      context: ./docker_mysql
      dockerfile: Dockerfile
    image: iotmysql:8.0
    container_name: iot-mysql
    environment:
      MYSQL_ROOT_PASSWORD: zyz123456

  iot-emqx:
    build:
      context: ./docker_emqx
      dockerfile: Dockerfile
    image: iotemqx:1.0
    container_name: iot-emqx
    ports:
      - "1883:1883"
      - "8083:8083"
      - "8084:8084"
      - "8883:8883"
      - "18083:18083"

  iot-backend:
    build:
      context: ./docker_backend/IotManager
      dockerfile: Dockerfile
    image: iotbackend:1.0
    container_name: iot-backend
    ports:
      - "8000:8000"
    depends_on:
      - iot-mysql
      - iot-emqx

  iot-frontend:
    build:
      context: ./docker_frontend/Iot-manager
```

```
      dockerfile: Dockerfile
    image: iotfrontend:1.0
    container_name: iot-frontend
    ports:
      - "8080:80"
    depends_on:
      - iot-backend
```

mysql

dockerfile

```
#设定初始镜像mysql版本
FROM mysql:8.0.32
#设定工作目录，用于处理初始化mysql的sql文件
WORKDIR /docker-entrypoint-initdb.d
#默认mysql无法输入中文，修改编码使其支持中文
ENV LANG=C.UTF-8
#加入需要初始化的sql文件
ADD initdata.sql .
```

初始化脚本存在initdata.sql中。

django镜像创建

dcokerfile

```
FROM python:3.10.8

RUN apt-get update \
    && rm -rf /var/lib/apt/lists/*

WORKDIR /usr/src/app
COPY requirements.txt ./
RUN pip install -r requirements.txt
COPY . .

EXPOSE 8000
CMD ["python", "manage.py", "runserver", "0.0.0.0:8000", "--noreload"]
```

emqx镜像创建

dockefile

```
FROM emqx/emqx:5.3.2
COPY acl.conf /opt/emqx/etc/acl.conf
```

这里要把配置文件复制进去，来允许订阅系统消息，获取流量信息和设备连接和断开信息。

前端nginx镜像创建

首先使用npm run build打包生产环境版本。

Dockerfile

```
FROM nginx
RUN mkdir /app
COPY dist /app
COPY nginx.conf /etc/nginx/nginx.conf
```