一、compose简介：定义运行多容器docker应用程序工具，通过yml文件配置应用程序需要的服务，在文件中创建并启动所有服务

1、使用步骤

a）Dockerfile定义应用程序环境

b）docker-compose.yml定义构成应用程序的服务，在隔离环境中一起运行

c）执行docker-compose up 命令启动并允许整个程序

d）docker-compose.yml配置案例

# yaml 配置实例  
version**:**'3'  
services:  
  web:  
    build**:**.  
    ports**:**   - "5000:5000"  
    volumes**:**   - .:/code  
    - logvolume01:/var/log  
    links**:**   - redis  
  redis:  
    image**:**redis  
volumes:  
  logvolume01**:**{}

二、安装

1、linux：alpine需要如下包（py-pip，python-dev，libffi-dev，openssl-dev，gcc，libc-dev，和 make）

$ sudo curl -L "https://github.com/docker/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

$ sudo chmod +x /usr/local/bin/docker-compose

$ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

$ docker-compose --version

cker-compose version 1.24.1, build 4667896b

三、使用

1、创建测试目录

$ mkdir composetest

$ cd composetest

2、创建app.py文件

**import** time  
**import** redis  
**from** flask **import** Flask  
app = Flask(\_\_name\_\_)  
cache = redis.Redis(host='redis', port=6379)  
**def** get\_hit\_count():  
    retries = 5  
    **while** True:  
        **try**:  
            **return** cache.incr('hits')  
        **except** redis.exceptions.ConnectionError **as** exc:  
            **if** retries == 0:  
                **raise** exc  
            retries -= 1  
            time.sleep(0.5)  
@app.route('/')  
**def** hello():  
    count = get\_hit\_count()  
    **return** 'Hello World! I have been seen {} times.**\n**'.format(count)

3、创建requirements.txt

flask

redis

4、创建Dockerfile

FROM python:3.7-alpine

WORKDIR /code

ENV FLASK\_APP app.py

ENV FLASK\_RUN\_HOST 0.0.0.0

RUN apk add --no-cache gcc musl-dev linux-headers

COPY requirements.txt requirements.txt

RUN pip install -r requirements.txt

COPY . .

CMD ["flask", "run"]

5、创建docker-compose.yml

# yaml 配置  
version**:**'3'  
services:  
  web:  
    build**:**.  
    ports**:**     - "5000:5000"  
  redis:  
    image**:**"redis:alpine"

6、compose命令构建并允许应用

docker-compose up

docker-compose up -d （守护进程）