Index：

1安装docker

2根据dockerfile requirement生成一个镜像

进入镜像 check镜像是否符合要求

3.在原有的镜像上添加功能

3.1利用dockerfile

3.2 进入镜像安装 之后commit

4 利用docker-compose 让主机里的代码使用该镜像运行

**1 安装 docker**

网上查找教程windows Ubuntu mac

Ps: Windows: 不建议系统是家庭版的用户 强行增加hyper-v，容易引起docker和虚拟机混乱。

Runtimeerror: 当pip和apt-get的时候遇到连接不上网络，urllib3 runtimeerror相关问题，建议换源 具体可上网查询 pip换源

**2根据dockerfile requirement生成一个镜像**

$ docker images

看现有镜像

$ docker ps

查看现有容器

$Vim requirement.txt

#你需要在python里面装的包都可以放在requirement里面

requirment :

Django

zmq

django-registration

stripe

mysql

apscheduler==2.1.2

opencv-python

$vim dockerfile

Dockerfile:

FROM python:3.6

#3.6不用声明系统是ubuntu还是Windows ，python36可以跨平台

ENV PYTHONUNBUFFERED 1

RUN apt-get update && \ DEBIAN\_FRONTEND=noninteractive apt-get -yq install sqlite3 && \

rm -rf /var/lib/apt/lists/\* /tmp/\* /var/tmp/\*

#run后面是你要安装的东西 可以理解为在镜像里面run了这个指令

#可以把需要安装的包全部声明在requirement.txt里面也可以 run 一个个安装

RUN mkdir /code

#在你的镜像里创造一个文件夹

WORKDIR /code

#把镜像的工作地址设置为那个文件夹

ADD requirements.txt /code/

#把主机上的 requirement.txt添加到刚刚创立的文件夹里面

RUN pip install -r requirements.txt #运行根据requirement.txt安装

ADD . /code/

#把当前目录的内容添加到code文件夹里

Dockerfile和requirement都准备好了就可以

$ docker build -t NAME:TAG .

比如名字叫 python36 Tag叫 Django:

docker build –t python36:Django不设置的话 默认是latest

现在进入你的镜像看看

$docker run -it python36:Django /bin/bash

进入之后

$Python

>>import Django

没有报错的话 镜像制作完成

**3 如果你突然要在老的镜像上添加一些包**

方法一:

更改你的dockerfile

$vim dockerfile

FROM python36:Django

RUN pip install numpy

$ docker build –t python36:Django

你可以用一个新的名字 新的tag做一个新镜像

也可以用一样的名字 覆盖老镜像

同上你可以进入镜像 python输入import numpy查看是否安装成功

**4 下载docker compose**

然后利用 doker-compose.yml 文件调用镜像

version: "3"

services:

web:

image: python36:django

# your image you want to use

working\_dir: /code

command: python manage.py runserver 0.0.0.0:8008

# This port needs to be the same of ports below.

volumes:

- ./django\_eve:/code

# modify django\_eve to your folder name. This folder stores the code.

ports:

- "8000:8000"

# your server port.

restart: always

写好compose之后

$ docker-compose up

跑起来拉~  
如果你想终止你的代码：

$docker-compose down