首先，你要用到的词汇介绍：

**镜像：**

特点：存档点，只读，无法放入文件。

作用：生成新的容器。

原因：镜像的作用是用来生成容器，然后在容器内进行各种修改。如果镜像允许修改的话，那么可能会出现多人为了满足自己程序运行环境不停更换软件版本的问题，所以docker将镜像作为存档点或者系统还原点，当用户想进行修改，需要用镜像生成新的容器。

**容器：**

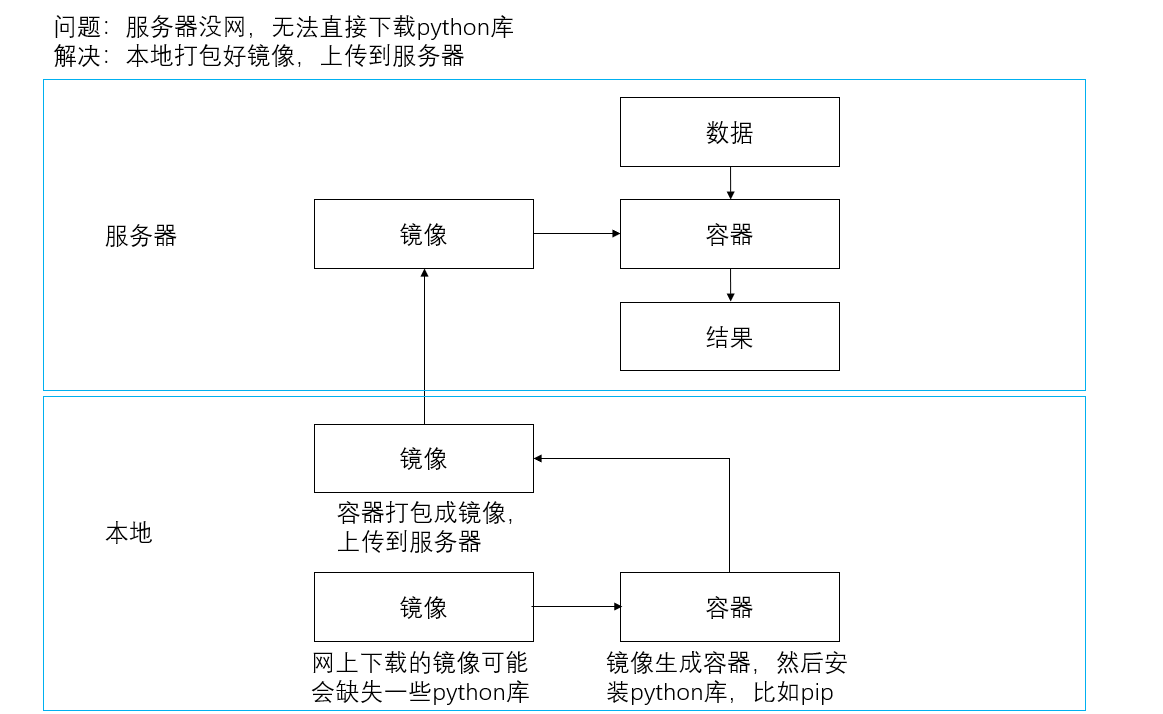
特点：初始内容复制自镜像，可以放入文件，可以修改内部程序。1个镜像可以生成无数个容器，只要空间够。

作用：用来根据自己需要进行各种修改，以满足某个程序运行。

备注：容器从镜像生成后，就已脱离镜像，后续操作只用操作容器就可以。

原因：容器以镜像为基础生成，可以根据自己需求修改。也可以经过打包，生成新的镜像，作为新的存档点。

然后是整体过程：



下面是详细步骤

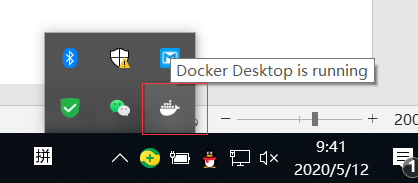
1. 安装

1.下载地址：<https://www.docker.com/products/docker-desktop>

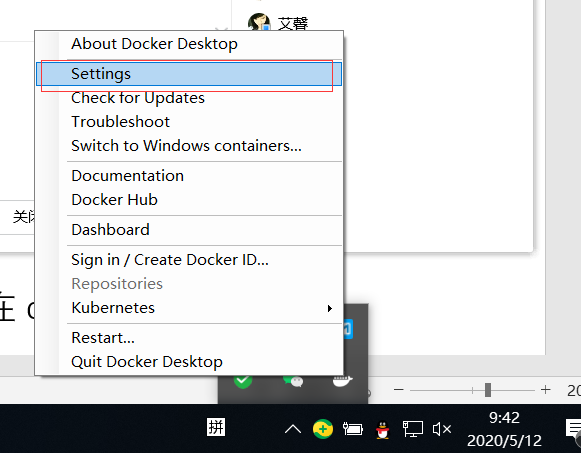
2.正常安装

3.安装后注意事项：

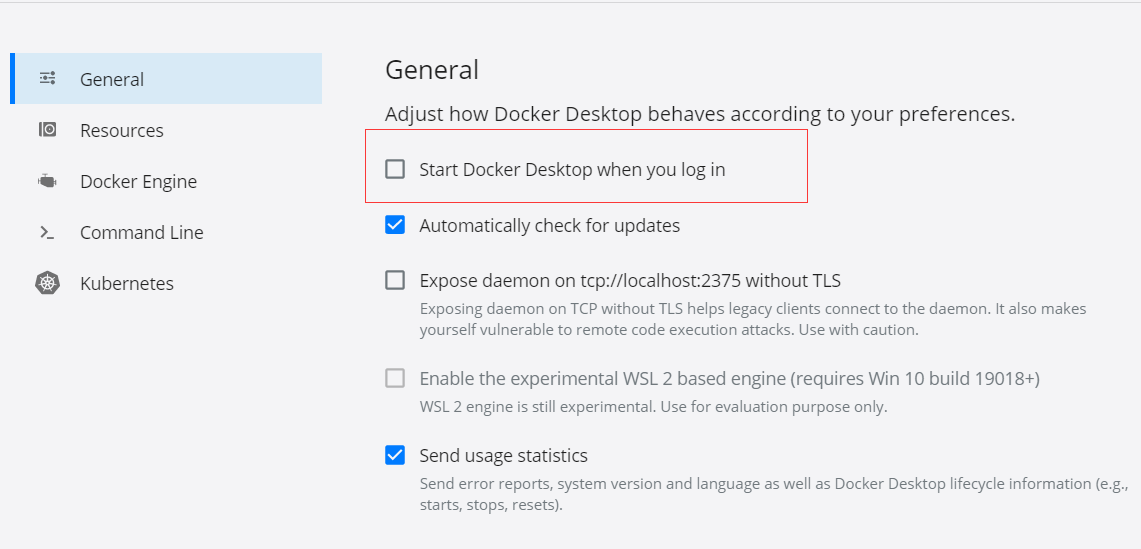
a.建议关掉开机启动，影响开机速度



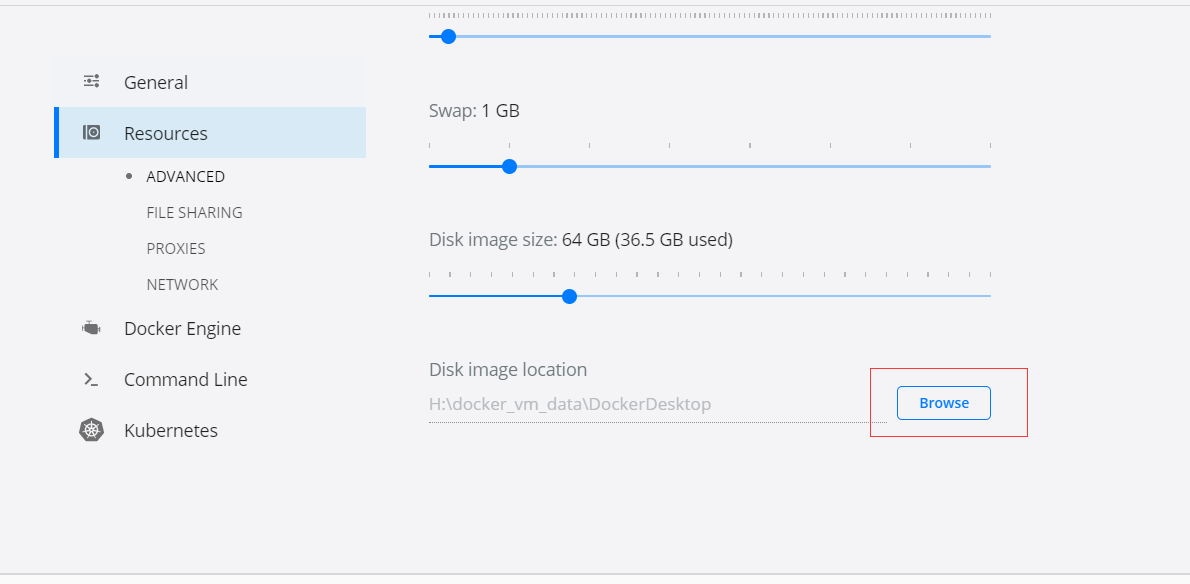
右键点击图标



取消勾选

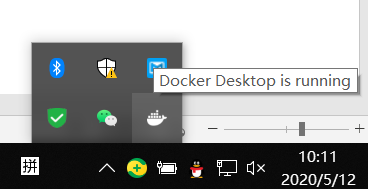


b.建议更改docker系统存放地点。该文件大小一般30GB以上，默认存放在c盘。



1. 使用

1.测试是否安装成功并启动：该位置鼠标放上去，提示Docker Desktop is running



2.在CMD命令行输入docker images，出现下面表头则正常



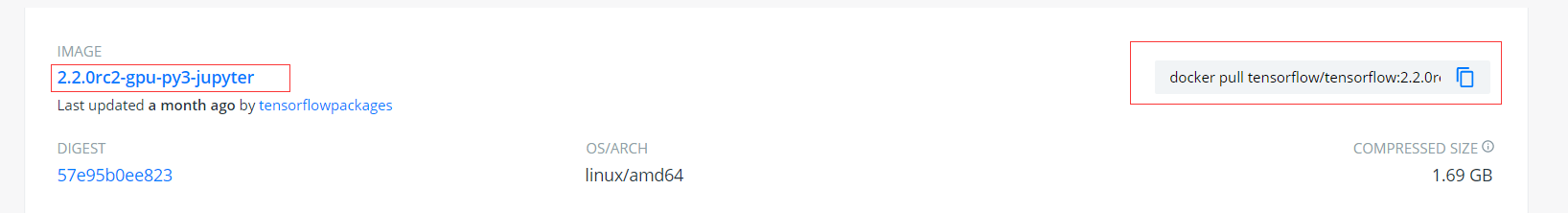
3.下载官方整合好的docker

页面：<https://hub.docker.com/>

这里使用整合好python3，tensorflow2.1，CUDA10.1，jupyter notebook的docker举例

页面：<https://hub.docker.com/r/tensorflow/tensorflow>

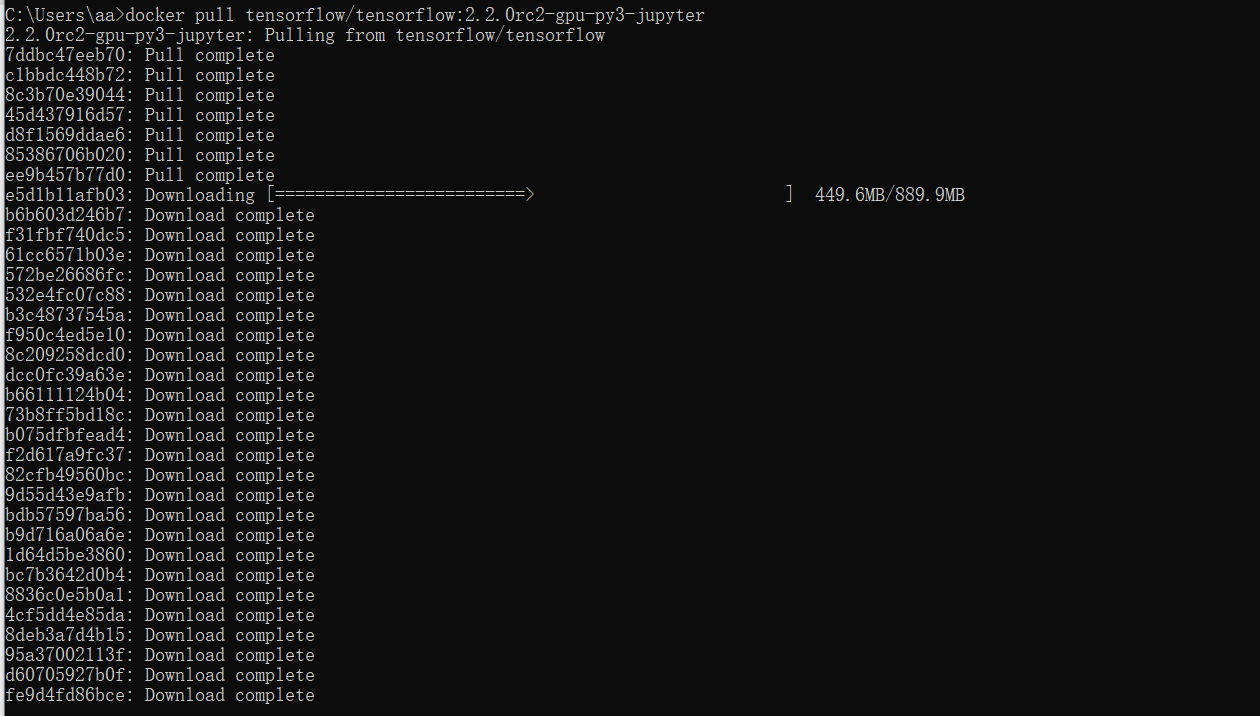
右边红框是下载地址：docker pull tensorflow/tensorflow:2.2.0rc2-gpu-py3-jupyter



在CMD命令行，运行刚才获得的地址，开始下载docker



下载中

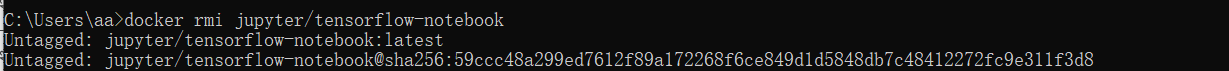


下载全部完成后，在CMD命令行输入docker images，会发现新出现的image镜像



===================================================

如果你想删除镜像的话，可以使用指令 docker rmi 镜像名 ，比如这里是docker rmi tensorflow/tensorflow，但是前提是要删除的镜像的容器需要先删除，方法见后续文档

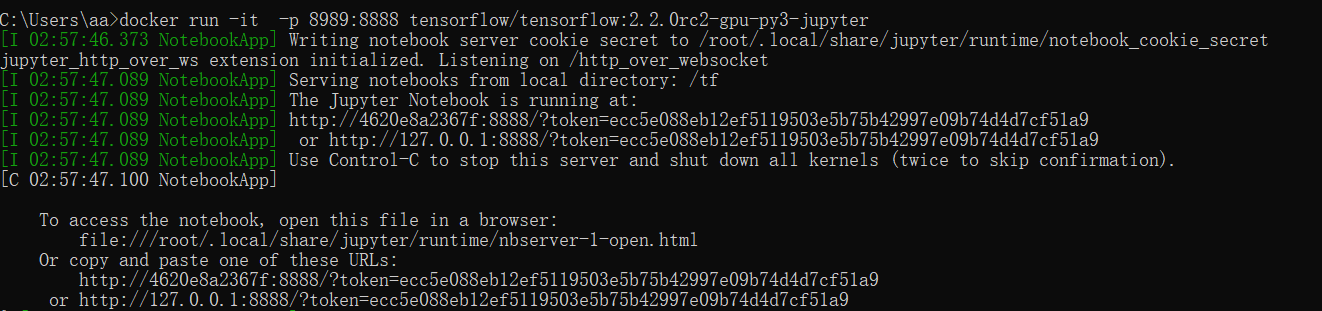


===================================================

4.启动镜像

在CMD命令行输入docker run -it -p 替代端口:8888 镜像名，比如docker run -it -p 8989:8888 tensorflow/tensorflow:2.2.0rc2-gpu-py3-jupyter。因为这里用到GPU(首先确认你的电脑有支持CUDA的显卡)，所以加入额外参数，输入替换为docker run -it --runtime=nvidia -p 8989:8888 tensorflow/tensorflow:2.2.0rc2-gpu-py3-jupyter。

如果你需要指定GPU，那么还要加上-e NVIDIA\_VISIBLE\_DEVICES=1(0代表第一块显卡,1代表第二块显卡,不加参数是全部启动)。比如docker run -it --runtime=nvidia -e NVIDIA\_VISIBLE\_DEVICES=0 -p 8989:8888 tensorflow/tensorflow:2.2.0rc2-gpu-py3-jupyter



这里虽然启动了镜像，但是我们想要启动容器。因为镜像无法写入，容器可以写入文件放入你想要处理的输入或加入你想加载的库。

我们通过Ctrl+C来停止进程（偶尔会有提示，需要你再确认一遍，按提示需要输入Yes之类的）

===================================================

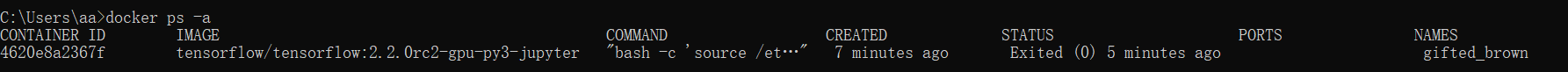
替代端口为随意输入的数字，在服务器上执行时不要与其他人的端口重复，查询方法CMD指令docker ps -a



===================================================

5.启动容器

启动镜像后，我们通过可以在CMD命令行输入docker ps -a来查看所有容器



CONTAINER\_ID：容器ID，不过一般用容器名字就可以了，名字见最后一个参数

IMAGE：来源镜像名

COMMAND：我也不知道，目前没用到

CREATED：容器创建时间

STATUS：容器状态，运行还是退出？运行会显示：Up 时间；退出会显示Exited 时间

PORTS：替代端口，就是docker run 镜像 时设置的

NAMES：容器的名字，是随机生成的

启动容器指令，在CMD命令行输入docker start 容器名字，这里是docker start gifted\_brown



===================================================

关闭容器指令，在CMD命令行输入docker stop 容器名字，这里是docker stop gifted\_brown

删除容器指令，在命令行输入docker rm 容器名字。需要先停止容器才能执行。如果遇到需要删除但是无法删除的情况，可以使用docker rm -f 容器名字 进行强制删除。

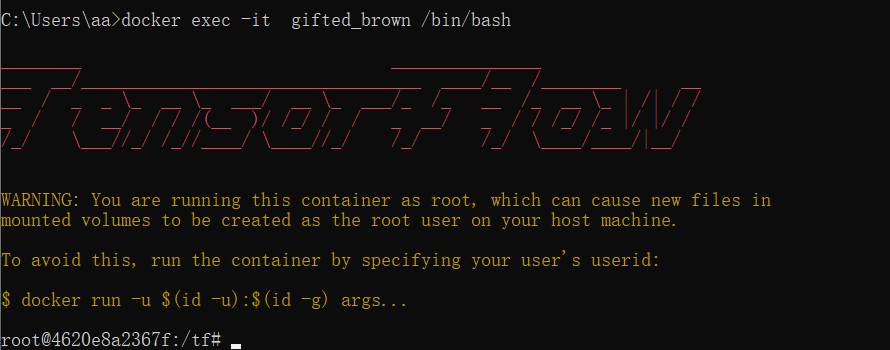
===================================================

6.进入容器

进入容器有两种界面（如果容器集成了juyter notebook那么是两种，否则只有一种linux命令行）：

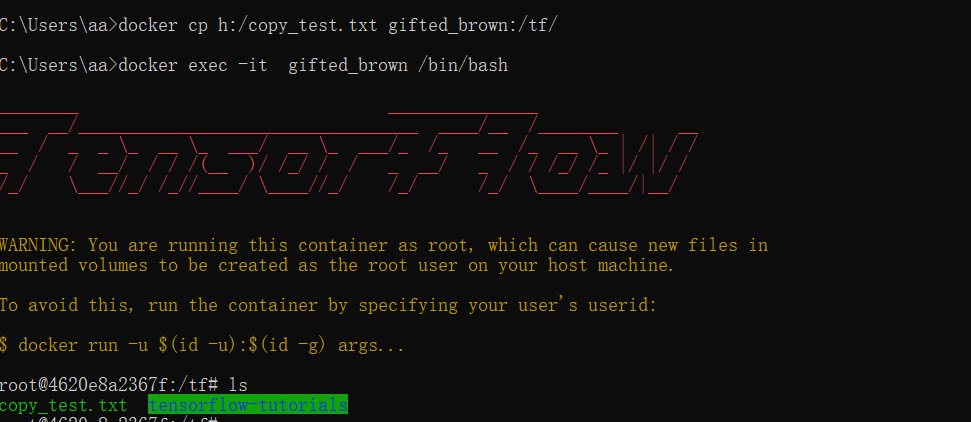
a.进行容器的linux界面，作用是给容器安装程序或者放入数据。使用CMD命令行输入docker exec -it 容器名字 /bin/bash，这里是docker exec -it gifted\_brown /bin/bash。

按Ctrl+D挂起容器，回到CMD界面。



这里可以使用向容器内外拷贝数据，使用CMD命令行输入

向容器内拷贝：docker cp 要拷贝到宿主机的相应路径 容器名字:要拷贝的文件在容器里面的路径



向容器外拷贝：docker cp容器名字:要拷贝的文件在容器里面的路径 要拷贝到宿主机的相应路径

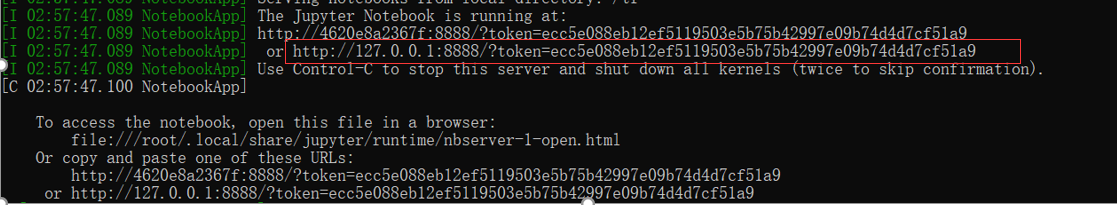


容器内的路径通过pwd指令获取



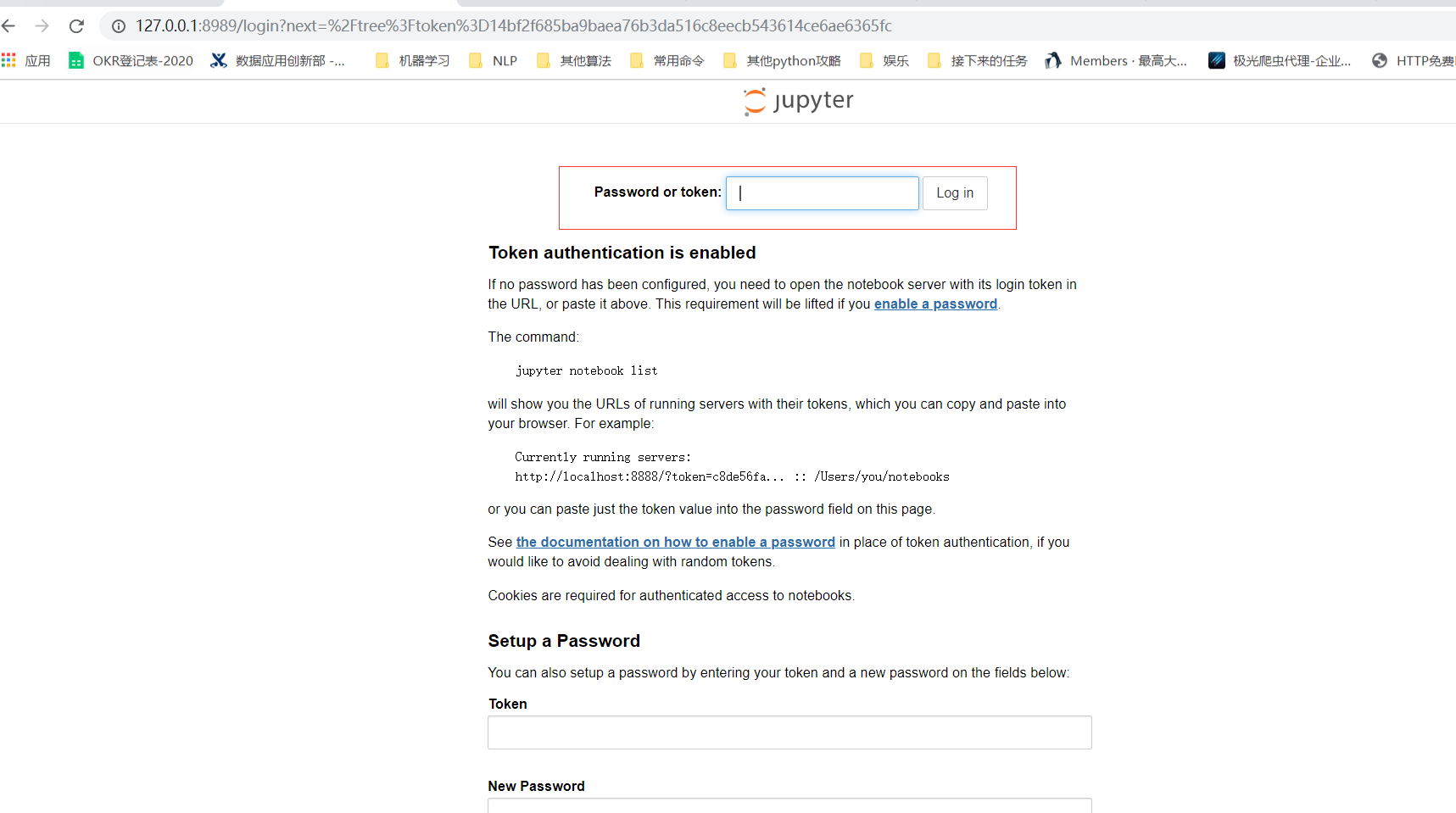
此外，也可以通过pip来安装python库

b.进行容器的jupyter界面（如果有的话），作用使用容器处理数据。使用CMD命令行输入docker exec -it 容器名字 jupyter notebook，这里是docker exec -it gifted\_brown jupyter notebook。

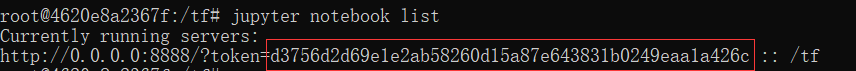


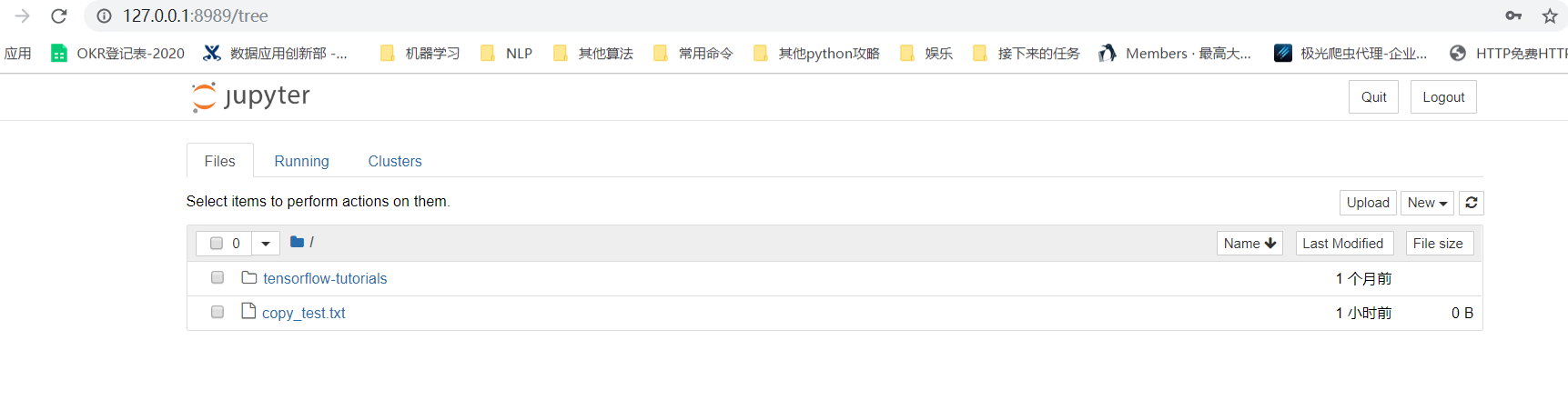
获取地址，然后修改ip部分为 10.120.130.13:自己之前设置的端口号，比如10.120.130.13:8989

打开浏览器，输入修改后地址，即可进入jupyter notebook（下方浏览器地址忽略，这只是截图）



如果输入后提示输入password or token，并且自己没有设置密码的情况下，通过docker exec -it gifted\_brown/bin/bash进入linux界面，然后输入jupyter notebook list，复制下面内容输入password or token窗口，可进入。





7.保存容器成为新的镜像

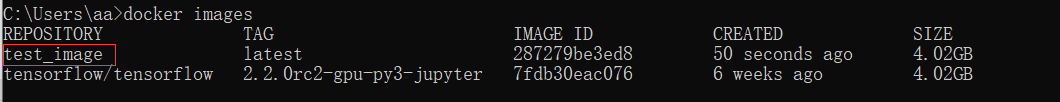
首先关闭要保存的容器，使用CMD命令行输入docker stop 容器名字，比如docker stop gifted\_brown。



将容器保存成为镜像，使用CMD命令行输入docker commit 容器名字 镜像名字，比如docker commit gifted\_brown test\_image



在docker images界面可以看到新增镜像



将新增镜像导出，使用CMD命令行输入docker save -o 保存文件.tar 镜像名：TAG，比如docker save -o test.tar test\_image:latest



8.导入新的镜像，将保存的文件放到新的docker环境导入

在新环境输入docker load -i 路径/镜像.tar，比如：docker load -i e:/test.tar然后重新开始步骤4。