<https://www.runoob.com/docker/docker-hello-world.html>

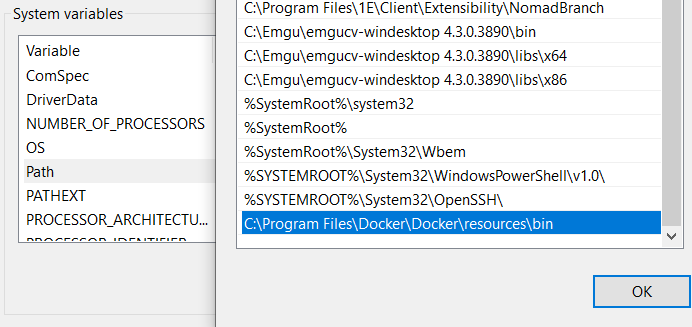
<https://www.cnblogs.com/fundebug/p/write-excellent-dockerfile.html>

<https://www.dazhuanlan.com/2020/03/20/5e73a1779dc2a/>

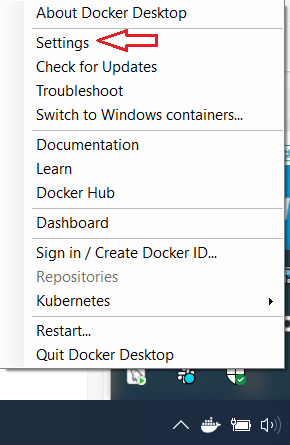
<https://www.jianshu.com/p/0fd24e7bb4ef>

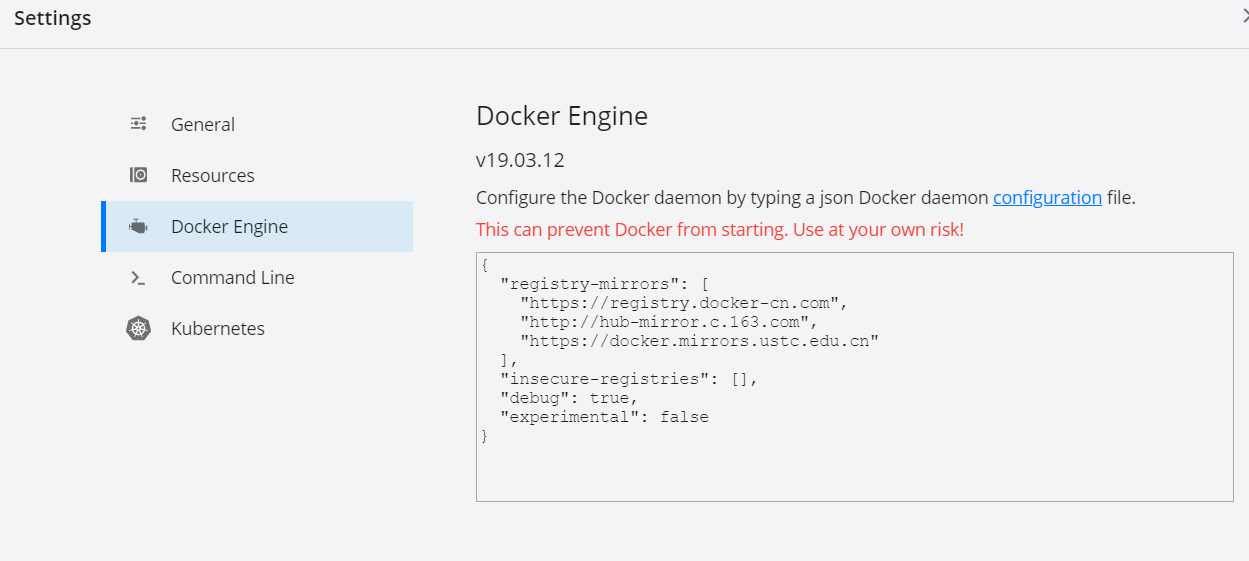
如果出现docker命令无法识别异常，就在环境变量的Path加上：

C:\Program Files\Docker\Docker\resources\bin



国内镜像 <https://www.runoob.com/docker/docker-mirror-acceleration.html>





{

"registry-mirrors": [

"https://registry.docker-cn.com",

"http://hub-mirror.c.163.com",

"https://docker.mirrors.ustc.edu.cn"

],

"insecure-registries": [],

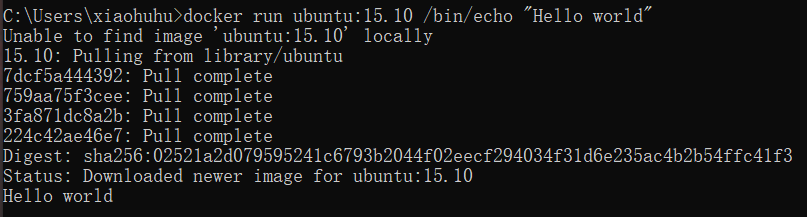
"debug": true,

"experimental": false

}

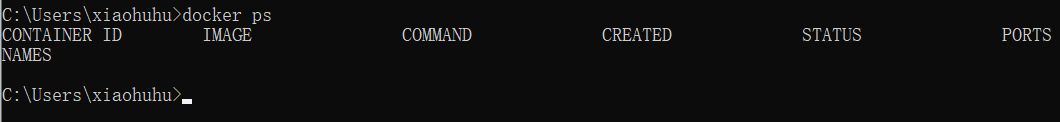
----------------------------------Hello World----------------------------------

1: docker run ubuntu:15.10 /bin/echo "Hello world"

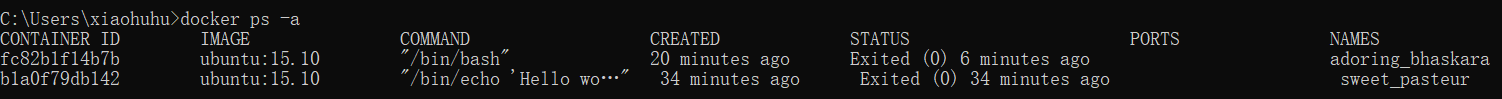


这个运行完之后会自动退出，所以docker ps没有东西

ps命令查看运行状态的容器



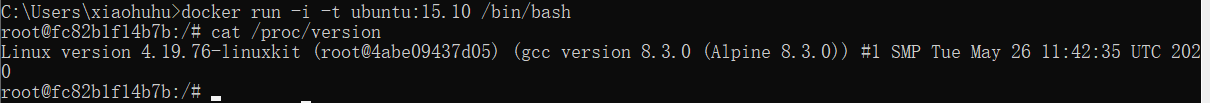
docker ps -a 查看所有状态的容器：



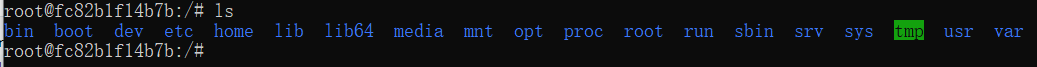
2: docker run -i -t ubuntu:15.10 /bin/bash

此时进入交互状态：/#

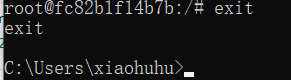
输入：cat /proc/version



输入：ls 当前目录下的文件列表



exit 退出容器



启动容器：docker start {container id}

Container id 可以通过docker ps -a 获得

//已交互模式启动一个已经存在的容器

docker start -i fc82b1f14b7b

-d命令指定容器在后台运行

进入容器：

在使用 **-d** 参数时，容器启动后会进入后台。此时想要进入容器，可以通过以下指令进入：

* **docker attach**
* **docker exec**：推荐大家使用 docker exec 命令，因为此退出容器终端，不会导致容器的停止。

docker start {container id} 启动一个容器后，可以通过docker exec来连接这个容器

docker exec -it {container id } /bin/bash

exit命令退出与容器的交互

rm -f 命令 删除一个容器

rmi 删除一个镜像

docker images 列出本地所有镜像

docker pull 从仓库拉取镜像

----------------------------网络端口映射----------------------------

<https://www.runoob.com/docker/docker-container-connection.html>

我们也可以使用 **-p** 标识来指定容器端口绑定到主机端口。

两种方式的区别是:

* **-P :**是容器内部端口**随机**映射到主机的高端口。
* **-p :**是容器内部端口绑定到**指定**的主机端口。

测试命令：docker run -d -p 5000:5000 training/webapp python app.py

容器互联：

<https://www.runoob.com/docker/docker-container-connection.html>

docker network create -d bridge test-net

----------------------------------React项目发布--------------------------------

<https://www.dazhuanlan.com/2020/03/20/5e73a1779dc2a/>

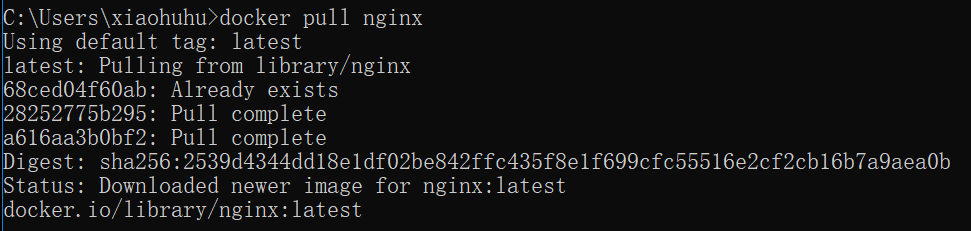
# 方案2：

# React App部署到Docker Container

<https://blog.csdn.net/Alexia23/article/details/100700512>

拉取Ningx镜像

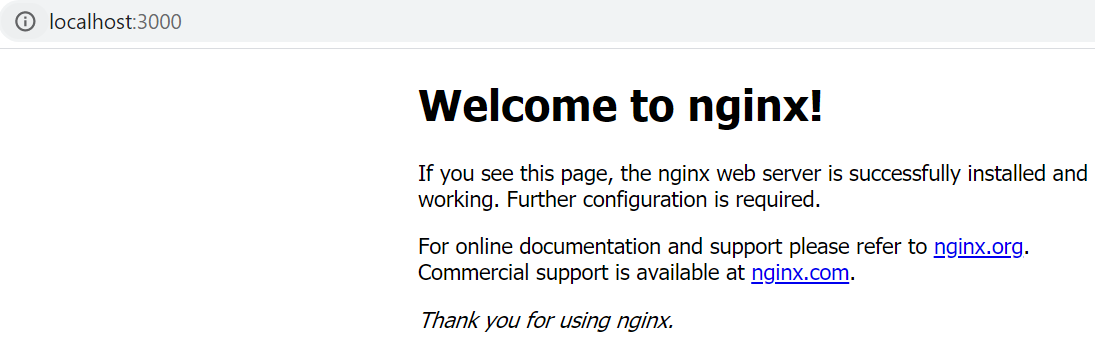
docker pull nginx



将容器端口80映射到主机端口3000

docker run -p 3000:80 nginx

浏览器访问localhost:3000



Npm run build 编译React JS项目

在React JS项目的根目录下执行：

docker cp ./build/. 1c0664e63f38:/usr/share/nginx/html

a96f99630aed是容器ID，该命令会将build文件夹下的所有文件都copy到容器的指定文件夹下

再访问localhost:3000 就是我们的React App了

使用以下命令可以查看Container中的部署情况

docker exec -it a96f99630aed /bin/sh

cd /usr/share/nginx/html

# 方案三：

<https://dev.to/subhransu/nevertheless-subhransu-maharana-coded-5eam>

在容器中运行.net core应用

<https://techcommunity.microsoft.com/t5/windows-dev-appconsult/first-steps-with-docker-quot-dockerizing-quot-your-first/ba-p/317567>

docker stack 与 docker compose的区别

<https://www.jianshu.com/p/1db6f0150fdb>