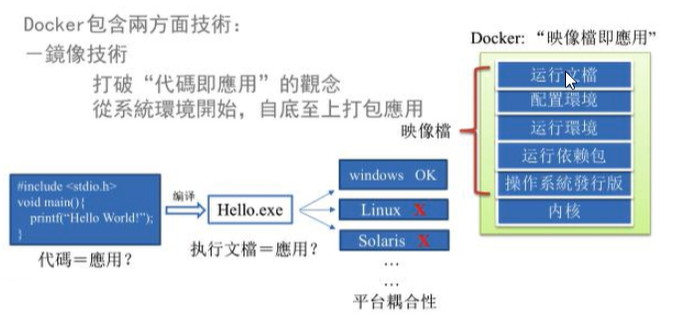
docker技术

## docker作用：

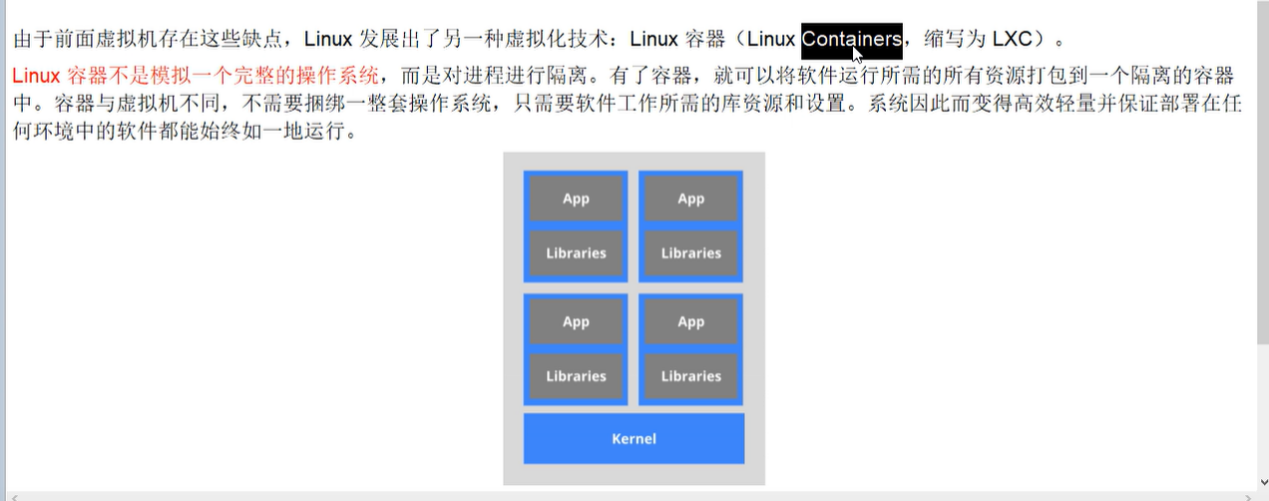
在应用系统基础上，安装一个个集装箱（容器），容器里面有镜像，各个集装箱（容器）无不干扰，这样在安装软件时候，第一省时间，直接下载镜像即可，第二：当大规模集群的时候，可能需要在每天机器上都装上MySQL，tomcat，php，jdk，nginx等，很耗时间，我们安装docker就节省很多时间。第三，当开发人员给代码给我们的时候，需要我们发布部署，可能因为我们的环境和开发环境不一样，所以发布就有错误，我们希望开发人员直接将这个环境，比如nginx环境直接打包镜像给我们，我们只要装docker，即可。



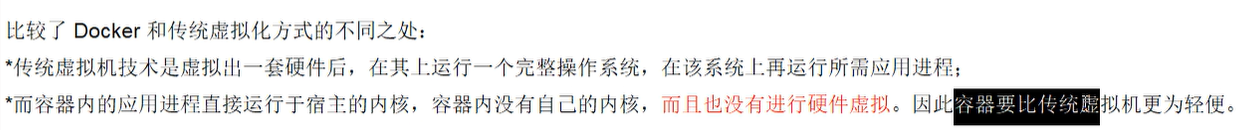
## 虚拟机与docker的区别：



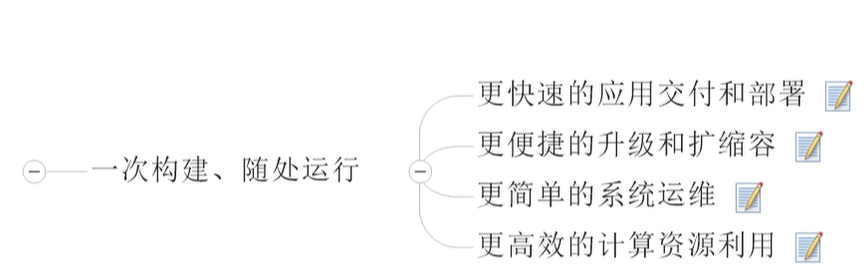
虚拟机就是安装一些完整服务，并且消耗硬件的操作系统

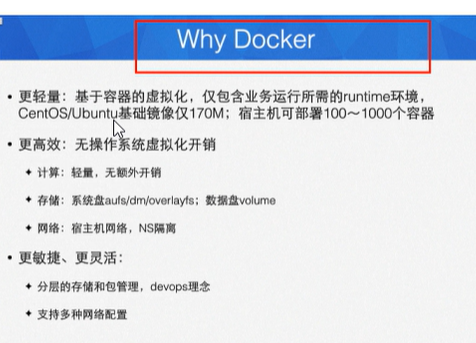


docker是基于操作系统上面部署容器进程，在每一个进程下面安装所需要的软件。只消耗进程所需的资源，这样部署在任何环境下的软件都能如一进行。

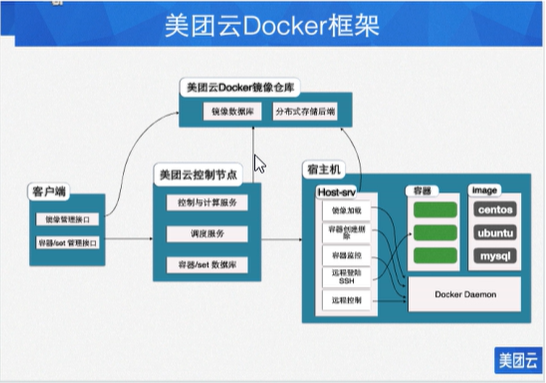


传统虚拟机是基于硬件模拟出来的系统，而docker则是运行宿主机的内核，容器本身没有内核。因此，容器要比传统的更为轻便。

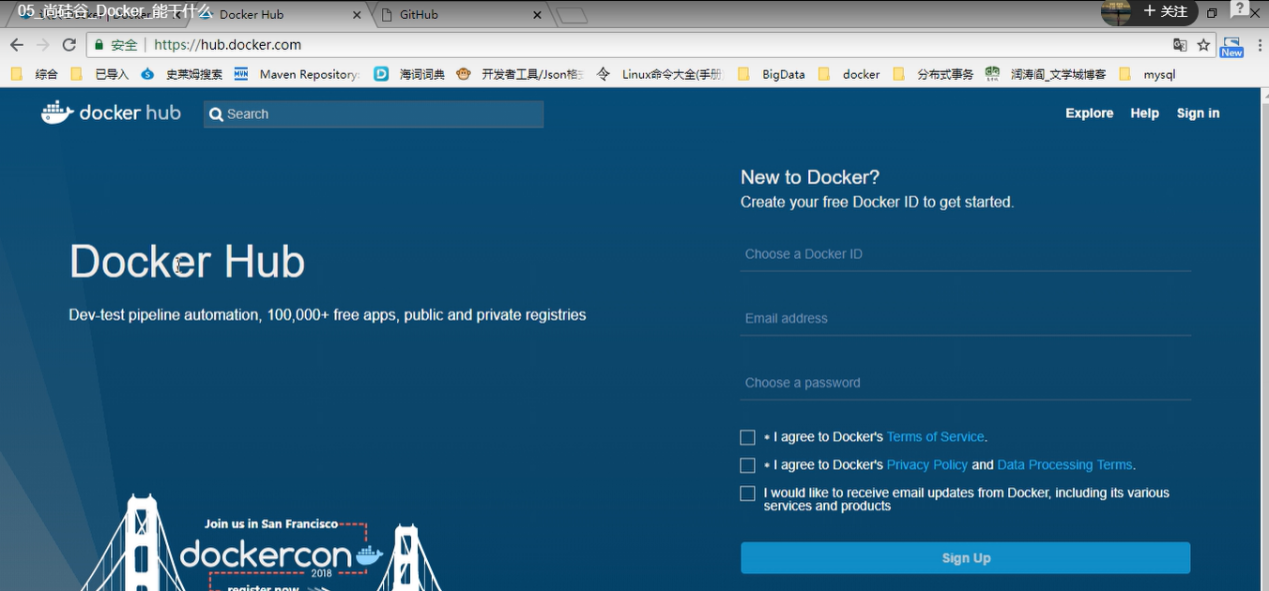




在docker上面安装centos只要170m,因为他只有内核，不消耗内存等，所以一台宿主机可以部署很多容器。docker其实就是一个缩小版的centos



## docker远程仓库：

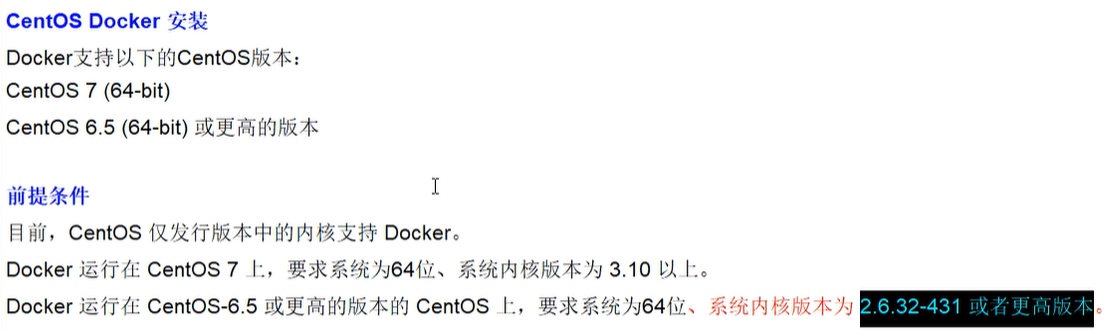


类似于github，docker hub就是docker的仓库，但是一般我们不用，因为国外防火墙太慢

## docker安装条件：

centos 6.5以上 内核要满足

centos 7以上

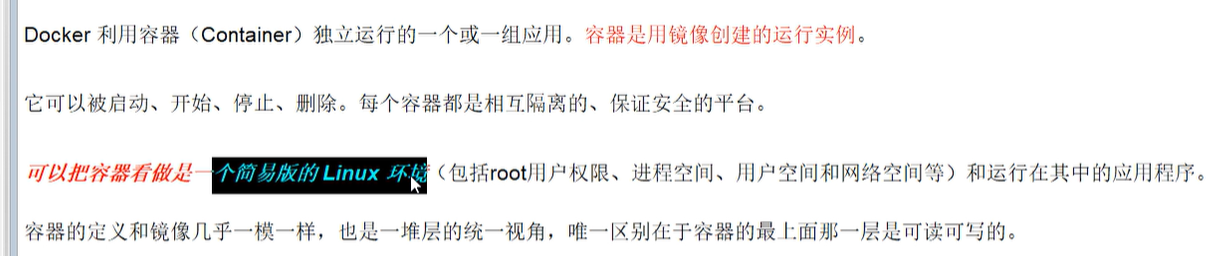


## docker的基本组成：



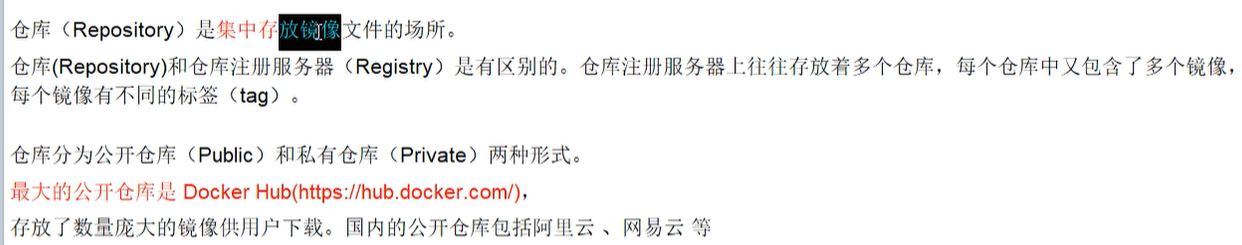


### 镜像就是一个只读的模板

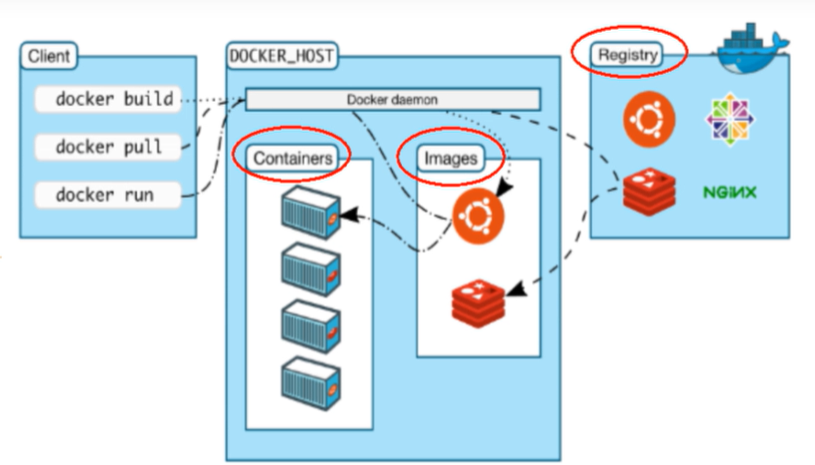


### 容器是独立的一组应用

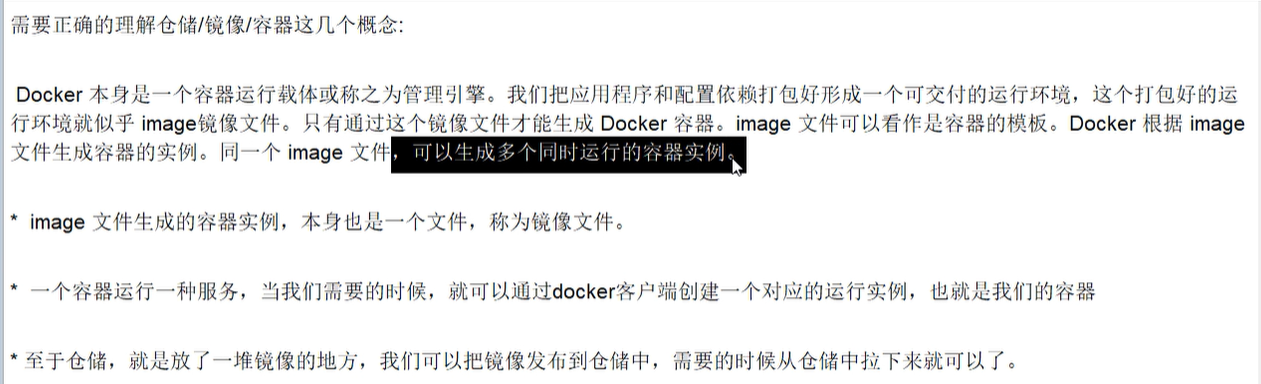
里面安装了镜像，且每个容器都是相互独立的。



### 仓库是集中存放镜像文件的场所

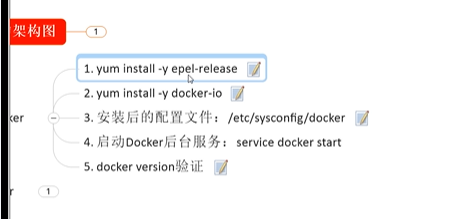


三者的工作原理。docker\_host:docker宿主机就是我们这个centos虚拟机，我们从仓库拉镜像，放到容器上面。而一些客户机器将自己的应用打包成镜像放到仓库。上面的图，非常生动形象。



# 安装docker

## 基于6.8系统：



## 基于7.0系统：

Docker从1.13版本之后采用时间线的方式作为版本号，分为社区版CE和企业版EE。（没钱，装社区版）

1. 查看yum包是否最新（这个贼慢）

sudo yum update

(问题)

我执行这个命令出错了

出错1，超时：

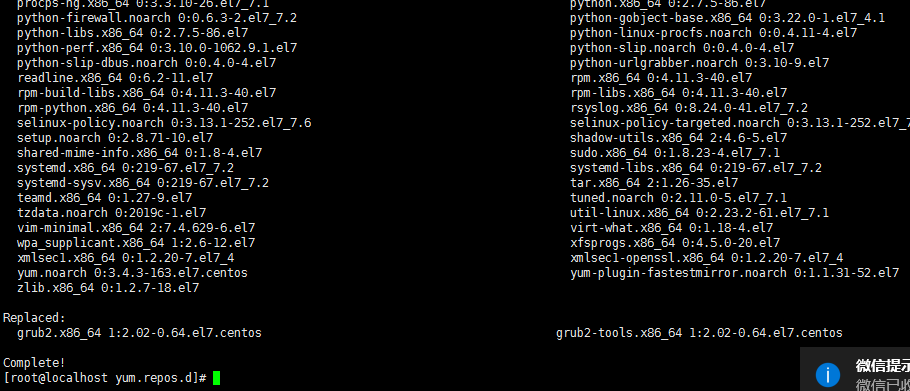
出错2，

Another app is currently holding the yum lock; waiting for it to exit

我执行了：

rm -f /var/run/yum.pid

这是强制关闭Yum进程，可以解决上面更新问题。



更新完美

1. 卸载旧版本docker，没装默认取消这一步

yum remove docker docker-common docker-selinux docker-engine docer-io

1. 安装需要的软件包

sudo yum install -y yum-utils device-mapper-persistent-data lvm2

1. 设置yum 源

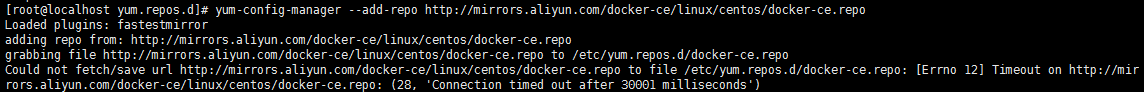
sudo yum-config-manager --add-repo <https://download.docker.com/linux/centos/docker-ce.repo>



其实上面是大坑

我们这里设置阿里源

yum-config-manager --add-repo <http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo>



设置阿里源出错。

没有办法，我只好手动更换。

先下载<http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo>

下载好Docker-ce后

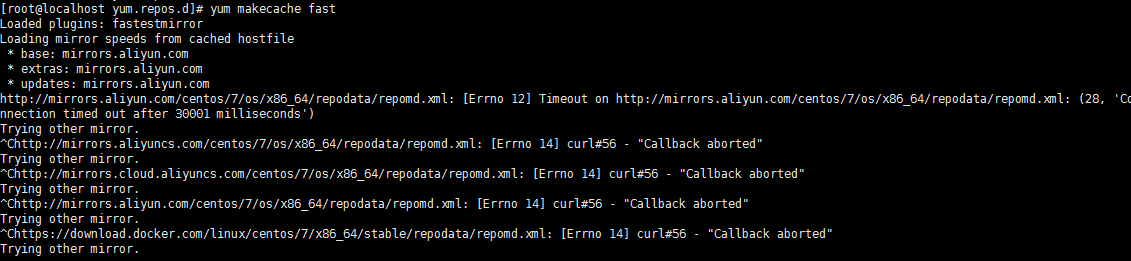
然后放到yum.repos.d下面，并给他

命名为docker-ce.repo



1. 安装yum 软件包索引

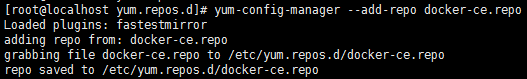
yum makecache fast



这一步又出错了。

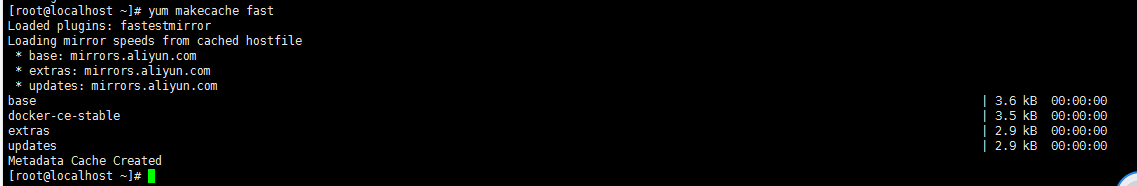
看来光将docker-ce.repo放到yum下面还不行。我这边需要安装一下。

yum-config-manager --add-repo docker-ce.repo



哈哈哈哈哈哈，原来是我网络问题，我试了一下yum install wget都不行，才知道是我设置了代理，哈哈哈，这是家里的网络

再次尝试，完美。



1. 安装DOCKER CE

yum -y install docker-ce

1. 启动docker

systemctl start docker

1. 测试
2. 查看版本

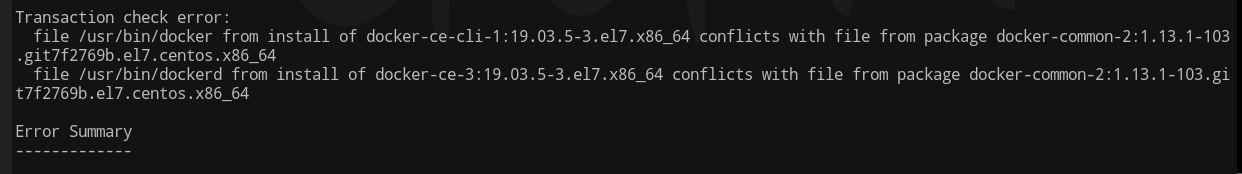
docker version

1. 运行hello world

docker run hello-world

安装失败：

执行yum -y install docker-ce



好像是没卸载干净有冲突了

好吧

yum remove docker\*

然后再执行：

yum -y install docker-ce



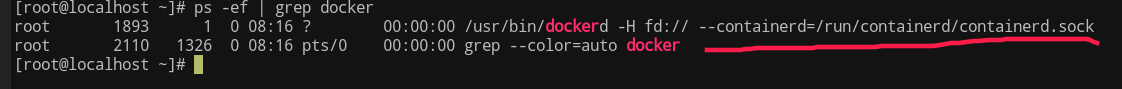
# 阿里云镜像加速：

docker镜像加速：

如果没配置的是阿里云，那么在拉仓库镜像就是国外的，国外网站因为有防火墙，非常慢，所以，我们一般要配置镜像加速

查看：

ps -ef | grep docker





将国外的docker镜像仓库放到了阿里云仓库上面，这样下载镜像速度就很快。

网址：<https://cr.console.aliyun.com/cn-hangzhou/instances/mirrors>

然后我用的是淘宝账号，我还注册了用户名：今年很难回家了



获得了一个加速器地址：

<https://1xiogwue.mirror.aliyuncs.com>

按照操作文档：

/etc/docker/daemon.json（这个我没有，所以新建了这个文档）

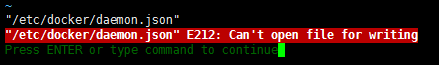


所以我直接在/etc/docker下面创建：

daemon.json文件加入内容：

{ "registry-mirrors": ["https://1xiogwue.mirror.aliyuncs.com"] }

保存

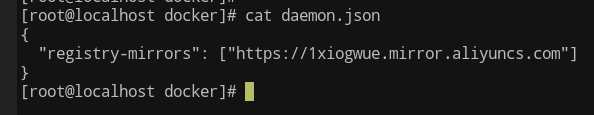


保存的时候提示这个。

没办法，加权限。

sudo systemctl daemon-reload

sudo systemctl restart docker

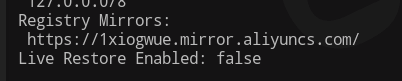


这样就行了

如何查看：

docker info

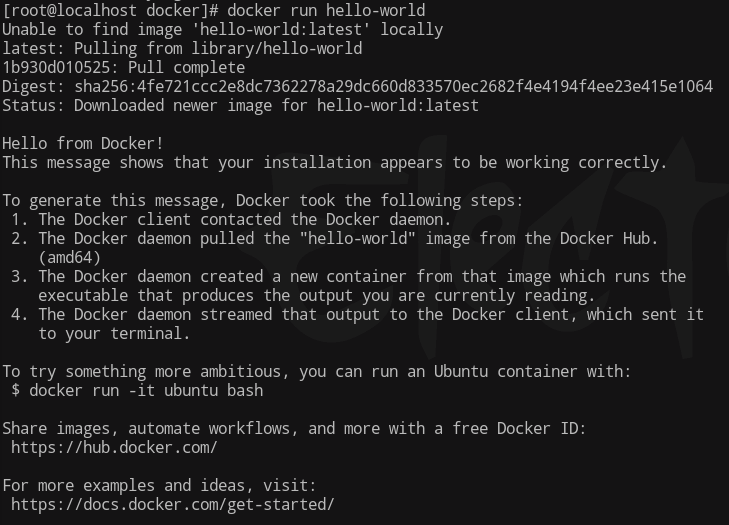
就可以看到：



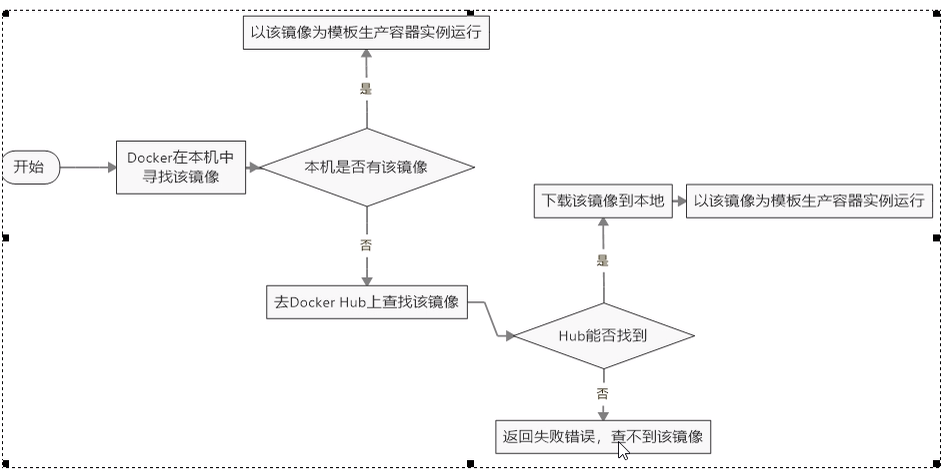
# docker run helloworld

这是docker 运行镜像的命令：

我们试试 docker run hello-world



一开始就提示，没有Hello-world这个镜像：



# docker帮助命令：

1. docker version

查看版本号

1. docker info

更详细的docker

甚至可以看见有多少容器，多少镜像

1. docker --help

一些命令介绍

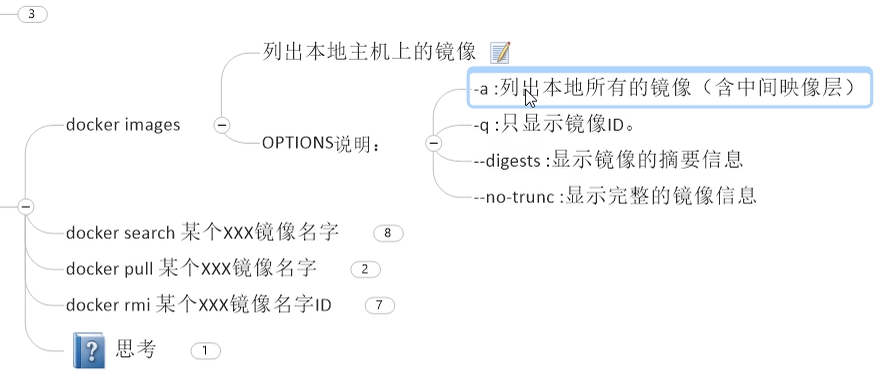
# docker daemon

docker deamon是docker的守护进程，docker 客户端通过命令行与deamon进程通信，进而完成docker操作。

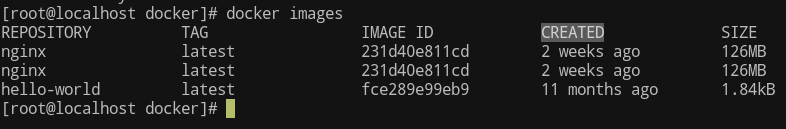
Docker Daemon有不同的修改方式：命令行修改，修改启动项，修改配置文件。其中仅仅只是研究或一次使用Docker Daemon的不同选项时，可以通过命令行的方式，此时Docker Daemon运行在前端，日志直接打印在终端上；当Docker Daemon的配置很稳定，不常修改时，可以将Docker Daemon的选项放在启动项中；当Docker Daemon的配置需要定时修改时，就可以把变化的选项放在配置文件中

重启： systemctl daemon-reload

# docker images的一些参数：



docker images



我之前docker pull nginx了，所以这里有三个

docker images -a 列出所有本地镜像（含中间镜像层）

docker images -q 列出镜像的id

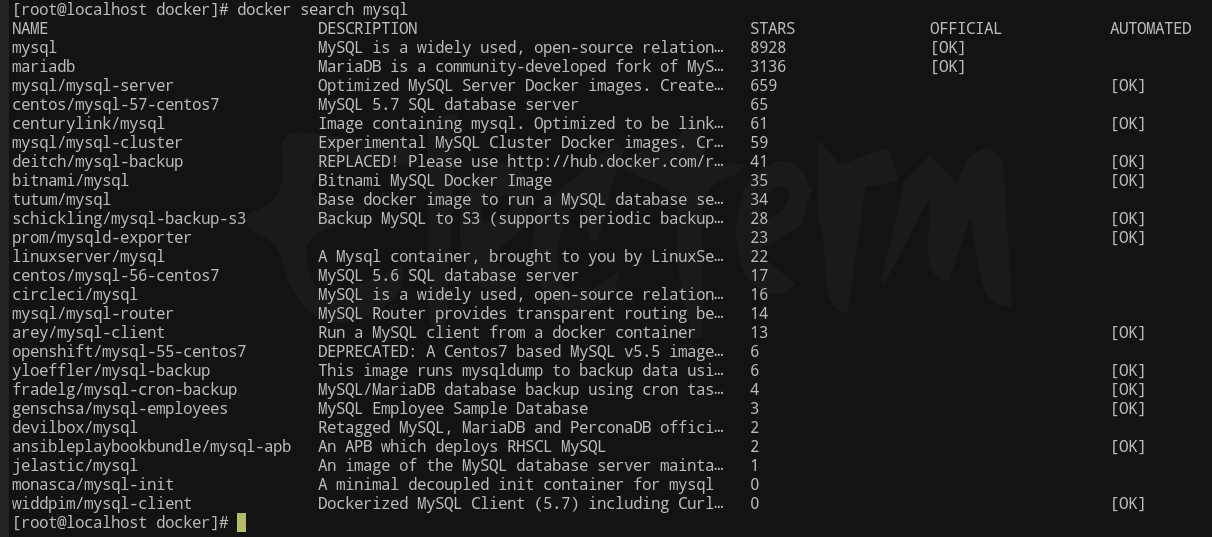
docker images digests

深度截图_选择区域_20191212224932

# docker search



docker search镜像名字



# docker pull

下载镜像，这是在阿里云上下载：

docker pull tomcat

这个命令默认等于

docker pull tomcat:latest

是拉取阿里云最新的一个tomcat

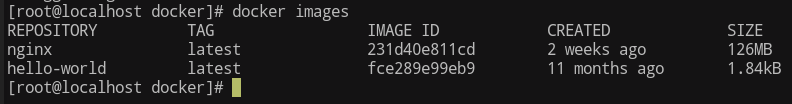
# 删除本地镜像：

docker rmi 镜像名

默认删除最近的一个镜像，如果有二个tomcat的话：

docker rmi -f +images名

强制删除这个镜像



## 问题：为啥这个nginx要126M，不是说docker轻量级吗？

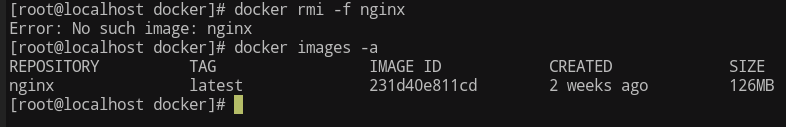
## 删除多个：

docker rmi nginx hello-world mysql

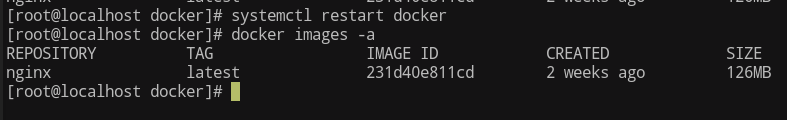
同时删除nginx与hello-world还有mysql

全部删除：

docker rmi -f $(docker images -qa)



很奇怪，我明明删除全部镜像，但是还是有nginx

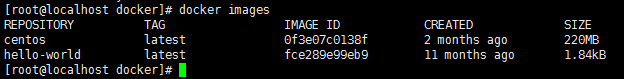
逼我重启docker：  


还是有，有毒啊，这个

# Dcoker容器：

Docker容器，我们可以看做这是一个简易版linux环境。

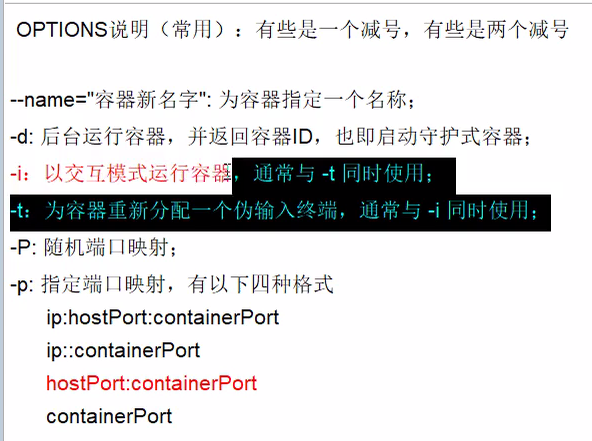
我们再拉取一个centos镜像试试



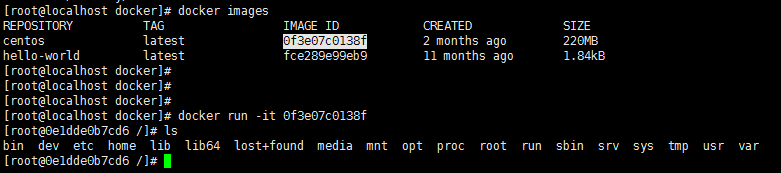
可以看到这个centos只有220M,因为减少了内核等

## 运行镜像，启动容器

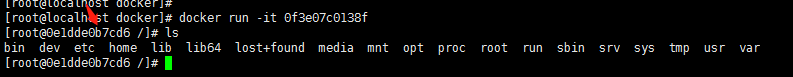
docker run 选项 images



先说-it，这二个一般是一起使用，-i是交互式，表示运行且进入这个容器，-t是不管输入镜像名或者镜像id都可以，我们一般是输入镜像id，因为，可能存在很多相同镜像。



这个看到，已经进入到这个容器里面了。

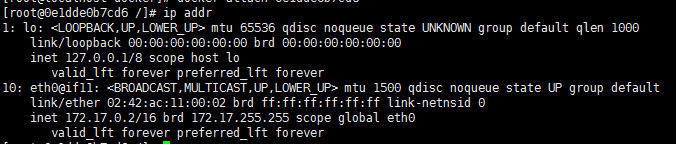
且，

这个是容器的id

## docker run的意义：

使用docker run命令来启动容器，docker在后台运行的标准操作包括  
1.检查本地是否存在指定的镜像，不存在则从公有仓库下载  
2.**使用镜像创建并启动容器**  
3.分配一个文件系统，并在只读的镜像层外面挂载一层可读可写层  
4.从宿主主机配置的网桥接口中桥接一个虚拟接口道容器中去  
5.**从地址池分配一个ip地址给容器**  
6.执行用户指定的应用程序  
7.执行完毕之后容器被终止

但是我进入centos容器后，运行ip addr，发现没有ip地址啊



## 查看宿主下面的所有运行容器

如何查看docker 下面所有运行的容器呢。

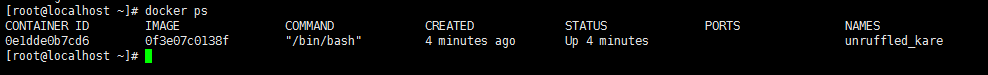
docker ps +选项



docker ps –a

查看所有运行过的容器

新打开linux终端，查看：



这个信息量很大，第一，可以看到容器id，第二，可以看到容器名。

至于name，因为我们没有给其命名，所以随机分配，其实我们在运行的时候可以给其分配id的

## 退出容器



exit 停止且退出

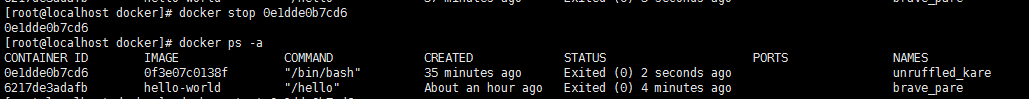
ctrl + P + Q不停止退出



但是，我虽然运行容器，且退出容器在后台运行，但是我想进入这个容器怎么办？

## 停止容器：

docker stop 容器号



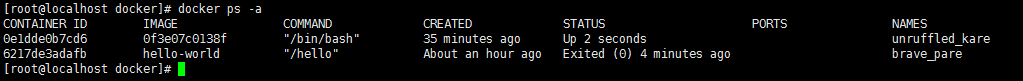
还有强制关闭容器

docker kill 容器号

## 启动容器

docker start 容器id

如何一个容器，我们退出了，千万不要以为这个容器就废了，我们没有删除这个容器，就意味着可以再启动。





启动的时候，如果返回容器编号，就说明成功了。

## 进入容器：

那如何再进入这个启动的容器呢？

docker attach 容器id

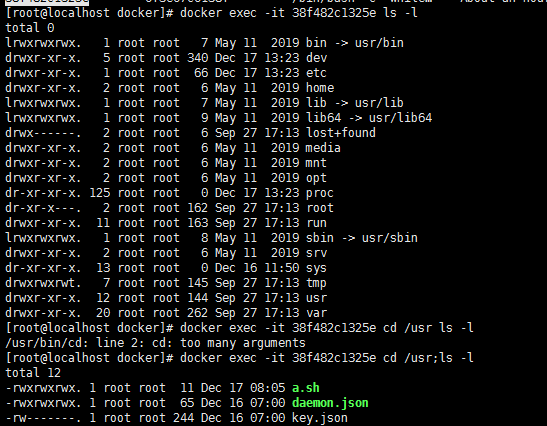
attach 依附

当然可以创建一个容器并进入，之前说过了。

也有在容器里面安装ssh，当然每一个容器其实都是轻量级别的linux,当然可以安装ssh，这样也可以



exec，进入，但是是启动新的进程。

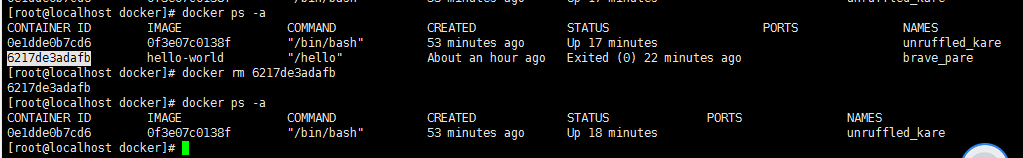


可以知道，可以在新的终端打开了容器

## 删除容器：

docker rm 容器号

-f是强制删除



那么问题来了，如何一次性全部删除

docker ps –a -q | xargs docker rm

-q是只打印容器id

## docker 启动后台守护容器：

之前在运行镜像的时候说过，有一个-d选项，这就是后台启动镜像id

我们再创建一个容器，并将其放入后台

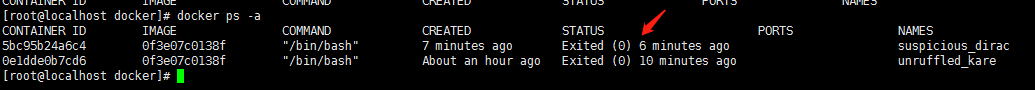




我是没看懂啦。

看了三遍才看懂，没办法，我笨啊。

docker ps 是查看docker前台运行的容器，但是我们是后台运行，如果前台没有一个进程，默认创建完这个容器就关闭，所以，我们可以通过 docker ps –a看到



这个进程其实关闭了。所以，要想这个进程运行，必须要有前台运行的进程才可以

如果我们给后台一个进程让他一直运行呢

docker run –d images /bin/bash –c “while true do echo nihao;sleep 2 done ”

先说下这个命令

/bin/bash –c “while true do echo nihao;sleep 2 done ”

官方解释：

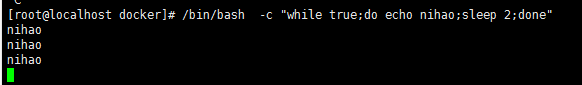
If the -c option is present, then commands are read from the first non-option argument command\_string. If there are arguments after the command\_string, they are

assigned to the positional parameters, starting with $0.

首先/ bin/bash –c，如果有这个命令，那么会从第一个非选项参数中读取命令。

如果是/bin/bash –c ./move.sh nihao

那么就会打印nihao，而不是执行脚本



上面的-c 千万不可以去掉，因为后面是命令，如果去掉默认执行脚本文件。



命令解释

while true

do

echo nihao;sleep 2

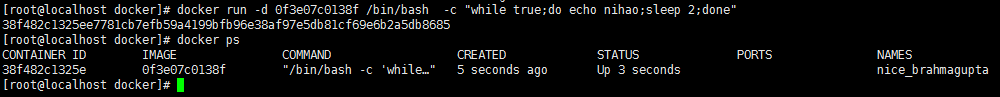
done

等同于while true;do echo nihao;sleep 2;done

我们在docker 运行容器时候docker run –d images /bin/bash –c “while true do echo nihao;sleep 2 done ”

其实就是在容器中执行命令

这是一个不间断命令，所以，这个后台docker进程不会关闭。



## docker查看容器日志：

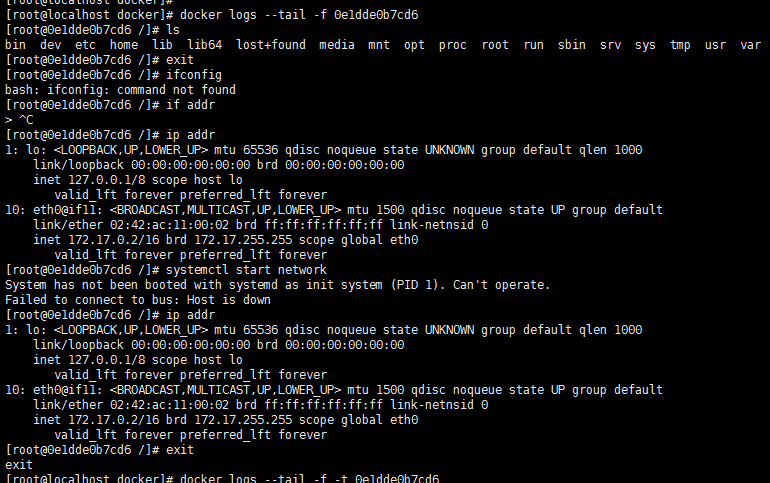
我们查看linux日志一般怎么查

tail –f 日志文件

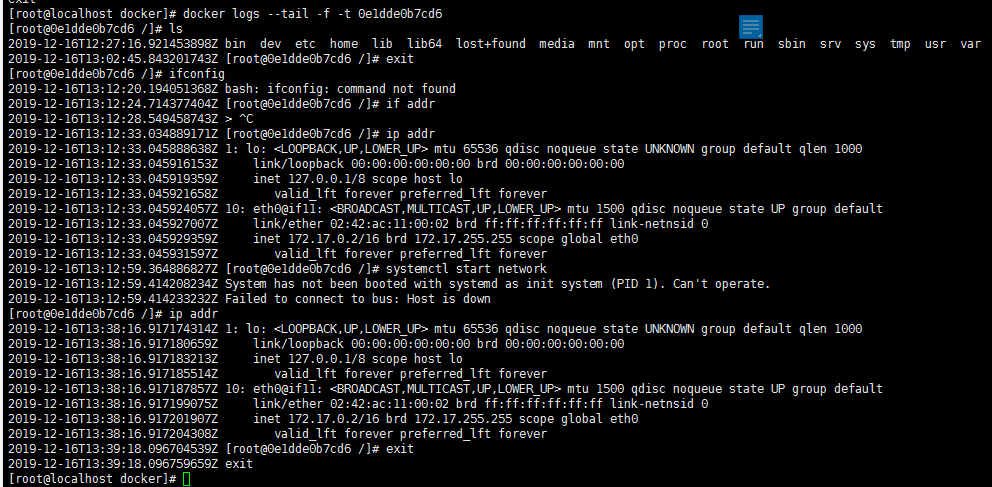
这里是docker



可以看到，我在这个容器所有操作内容都可以看到



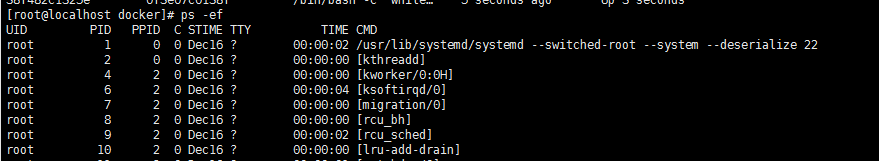
那么加-t，时间戳，这个其实就是时间标记



## 容器进程：

linux进程：

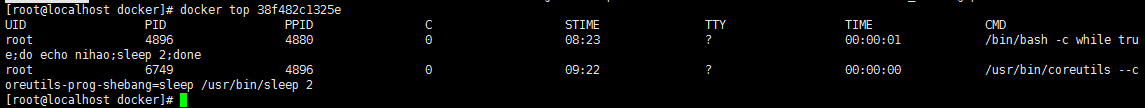
ps –ef



这是linux进程，pid是进程id，ppid是父进程id。

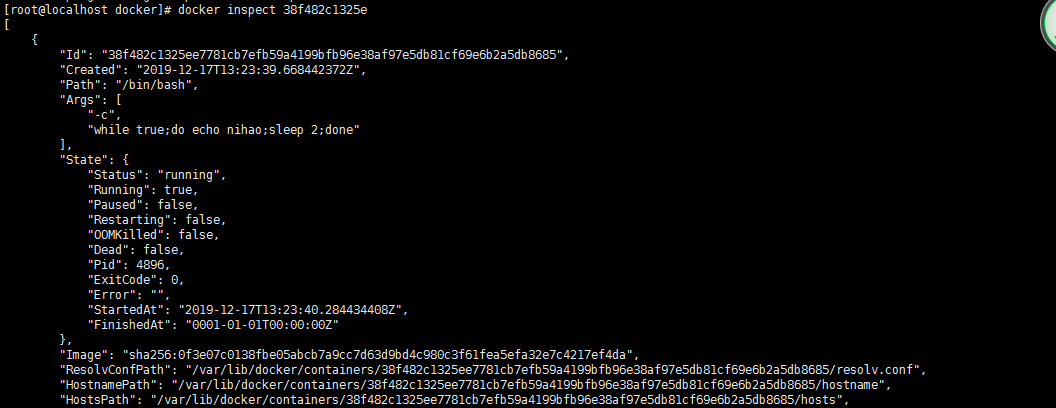
容器也有进程

docker top 容器id



## 查看容器内部细节：

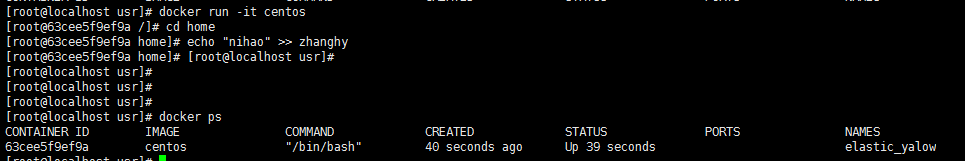
docker inspect 容器id



有很多关于容器的信息

## 将容器文件拷贝下来到宿主机：

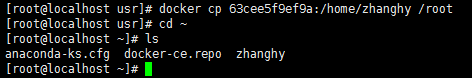
我们再容器下面创建一个文件，添加内容



接下来我们只推出容器而不关闭。

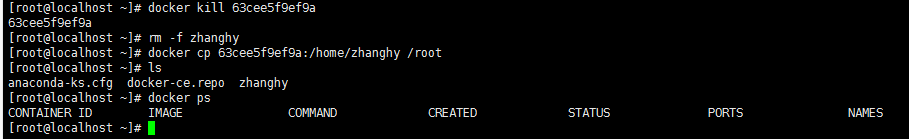
然后将文件拷贝出来

docker cp 容器id:文件路径 本地路径



可以看到，我成功将容器文件拷贝出来。

如果是关闭的容器可以吗。



也可以，哈哈哈

docker提交镜像

将容器中的镜像提交

docker docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]

option选项：

-a “作者“

-m “说明”

-p 提交时将容器暂停

-c 用dockerfile创建镜像

repository:

镜像名

tag

版本

一般提交第一次就是V2

更新镜像：

进入镜像

apt-get update

(因为这不是centos，这是linux)

dockerfile创建镜像。

[**关于Sending build context to Docker daemon 数据很大的问题**](https://www.cnblogs.com/cxbhakim/p/9316565.html)

以往进行docker build的时候都是在新建的文件夹下面进行，这次为了图方便，就直接放在开发根目录下进行build，这样子问题就来了。于是就有了下面的文件大小发送量：

|  |  |
| --- | --- |
| 1 | Sending build context to Docker daemon 4.309 GB |

第一反应就是是不是当前目录都给发送过去了。查了下，果然是的，发现在Docker client会发送Dockerfile同级目录下的所有文件到Dockerdaemon。

解决的方式就是之前第一句话说的，在一个新的文件夹中进行build这样子问题就不存在了，当然这样子会比较简洁，但是有时候脑袋就是这么直，我就是想要在根目录进行呢。这个时候该怎么办呢？—— .dockerignore 。

没错了，在Dockerfile同级目录下添加一个 .dockerignore 文件。

在 .dockerignore 中添加需要忽略的文件或者文件夹即可。于是在添加之后：

|  |  |
| --- | --- |
| 1 | Sending build context to Docker daemon 679.9 kB |

我这里新建了一个文件夹

在里面创建文件：



添加内容：

FROM ubuntu:latest

MAINTAINER zhanghy 26698@qq.com

RUN cd /usr \

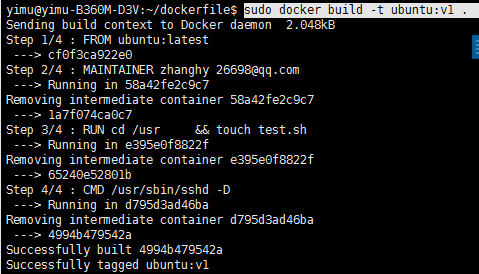
&& touch test.sh

CMD /usr/sbin/sshd -D

并且：

sudo docker build -t ubuntu:v1 .

表示我要创建一个ubuntu



可以看到新生成一个镜像 tag为v1

