注意 stop，restart后的容器，或者commit后的镜像启动环境变量需要重新加载，比如 source /etc/source

我们运行docker容器的时候，使用了-d参数，把容器在后台运行后。

1.启动docker容器

$ docker run -d -i -t <imageID> /bin/bash

上述命令执行某些命令回报错Failed to get D-Bus connection: Operation not permitted

可使用如下命令启动

$ docker run -d -i -t <imageID> /usr/sbin/init

这个时候，我们使用docker ps命令，我们就可以知道哪些程序在后台运行。

2.我们要怎么进入到docker容器呢？

（1）使用docker attach命令

我们使用

$ docker attach db3

或者

$ docker attach d48b21a7e439

db3是后台容器的NAMES,d48b21a7e439是容器的进程ID CONTAINER ID

然后就进去了这个容器的ssh界面。

但是它有一个缺点，只要这个连接终止，或者使用了exit命令，容器就会退出后台运行

（2）使用docker exec命令

这个命令使用exit命令后，不会退出后台，一般使用这个命令，使用方法如下

$ docker exec -it db3 /bin/bash 或者 docker exec -it d48b21a7e439 /bin/bash

db3是后台容器的NAMES,d48b21a7e439是容器的进程ID CONTAINER ID

/bin/bash 是固定写法

它也能进入这个容器

3.关闭启动重起

$ docker stop 容器id

$ docker start 容器id

$ docker restart 容器id

查看所有容器，包括关闭的

$ docker ps -a

4. docker提交新的镜像

$ docker commit afcaf46e8305(容器id) centos-vim（自定义名）

1. 停止所有的container，这样才能够删除其中的images：

$ docker stop $(docker ps -a -q)

如果想要删除所有container的话再加一个指令：

$ docker rm $(docker ps -a -q)

1. 查看当前有些什么images

$ docker images

1. 删除images，通过image的id来指定删除谁

$ docker rmi <image id>

想要删除untagged images，也就是那些id为<None>的image的话可以用

$ docker rmi $(docker images | grep "^<none>" | awk "{print $3}")

要删除全部image的话

$ docker rmi $(docker images -q)