# Docker拉下来的是一个操作系统，里面所有的环境（mysql，tomcat，java）都装好了的，直接拿到这个操作系统，做端口映射到本地虚拟机，即可使用。

# 1.Docker的生命周期管理常用命令：

systemctl stop docker.service

systemctl start docker.service

systemctl restart docker.service

systemctl status docker.service

拉一个项目下来：

docker pull how2j/tmall（拉下来的是一个操作系统，里面所有的环境都装好了的。）

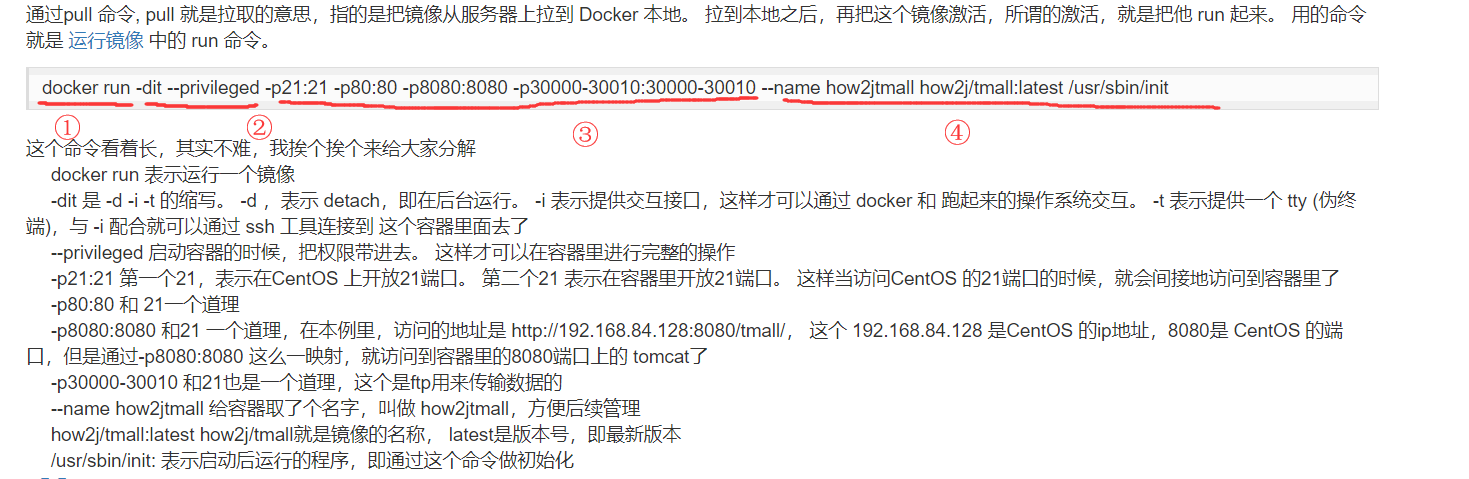
运行：

docker run -dit --privileged -p21:21 -p80:80 -p8080:8080 -p30000-30010:30000-30010 --name how2jtmall how2j/tmall:latest /usr/sbin/init

# 2.概念讲解：

道理主要是 把一个完整的带环境的软件包装成镜像放到docker上，然后要用的话就拉下来，直接使用。

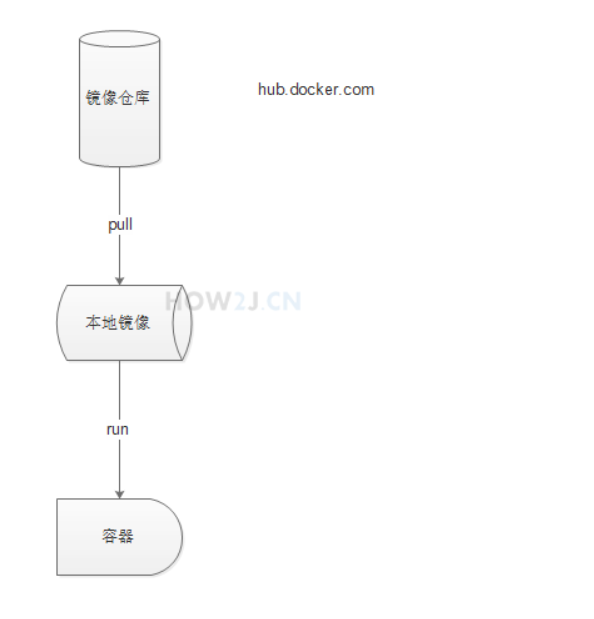
命令讲解：

①.运行镜像

②.系统配制连接交互的必须命令

③.端口映射，将容器（虚拟机）上的端口，映射到本地虚拟机上，就是第一个是虚拟机的端口，第二个是容器的端口，将容器端口映射到虚拟机，在访问虚拟机的8080其实是访问容器的8080。（两个操作系统的端口调一致。）

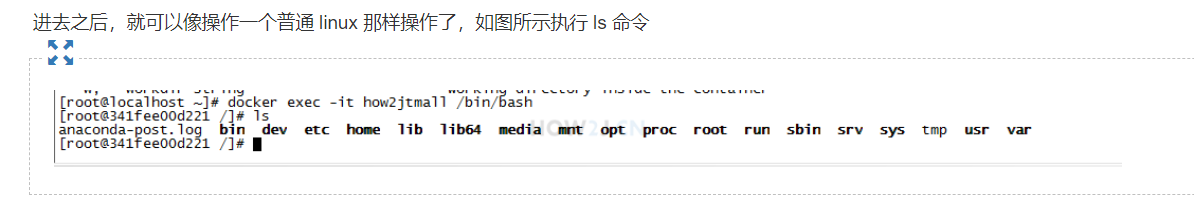
④．--name:命名。 互殴面都是关于命名之后对容器的初始化。



# 3.进入容器

就是进入运行容器的虚拟机：（拉下来的是一个linux操作系统）

docker exec -it how2jtmall /bin/bash

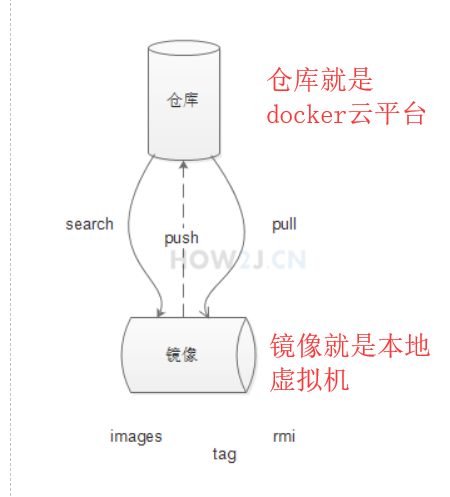


# 4.镜像管理(镜像就是拉到虚拟机的程序)

镜像管理常见的有这么些：  
1. search 查看仓库里有些什么镜像  
2. pull 拉取镜像  
3. images 查看本地有些什么镜像  
4. rmi 删除本地镜像  
5. 修改本地镜像名称  
6. push , 把镜像提交到仓库

7.tag 自定义标签，标记镜像（image时会出现两个一样的镜像，一个自定义，一个原本的）

8.删除全部镜像：docker rmi $(docker images -q)



拉到本地虚拟机的程序

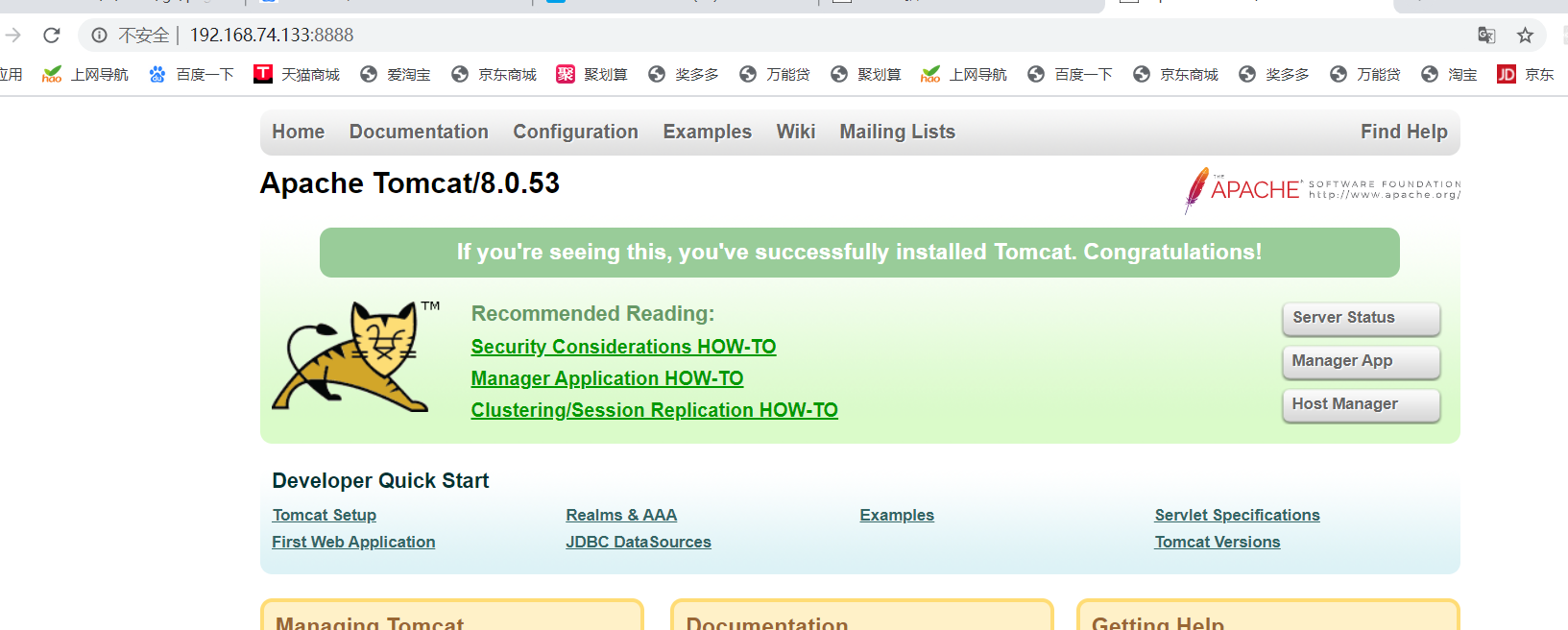
比如拉一个Tomcat镜像下来：

[docker pull tomcat:8.0](https://how2j.cn/k/docker/docker-images/docker%20pull%20tomcat:8.0)

然后本地8888端口运行这个镜像：

docker run -it --rm -p 8888:8080 tomcat:8.0

访问虚拟机地址的8888端口：



# 5.容器管理：

什么是容器？

奇妙比喻：类 （镜像）–>实例（容器）

所以最后运行的都是一个个容器

容器是由镜像实例化而来，这和我们学习的面向对象的概念十分相似，我们可以把镜像看作类，把容器看作类实例化后的对象。

也可以说镜像是文件, 容器是进程。 容器是基于镜像创建的, 即容器中的进程依赖于镜像中的文件, 这里的文件包括进程运行所需要的可执行文件， 依赖软件， 库文件， 配置文件等等...

1. 运行 run

2. 进入 exec attach

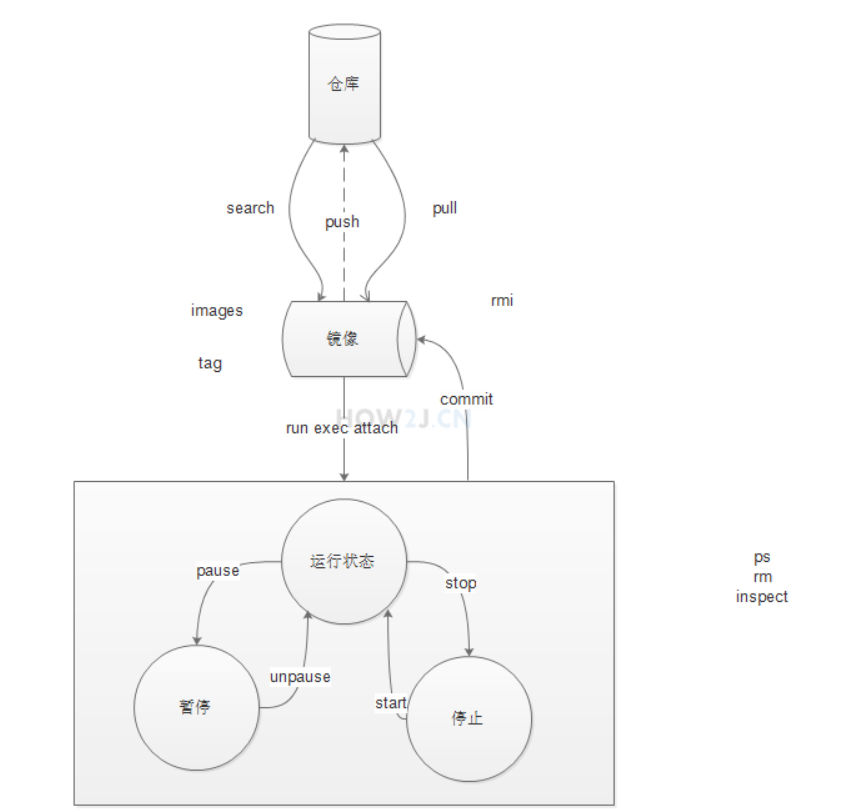
3. 生命周期管理， 暂停，恢复，停止，启动 pause, unpause, stop, start

4. ps 查看所有的容器

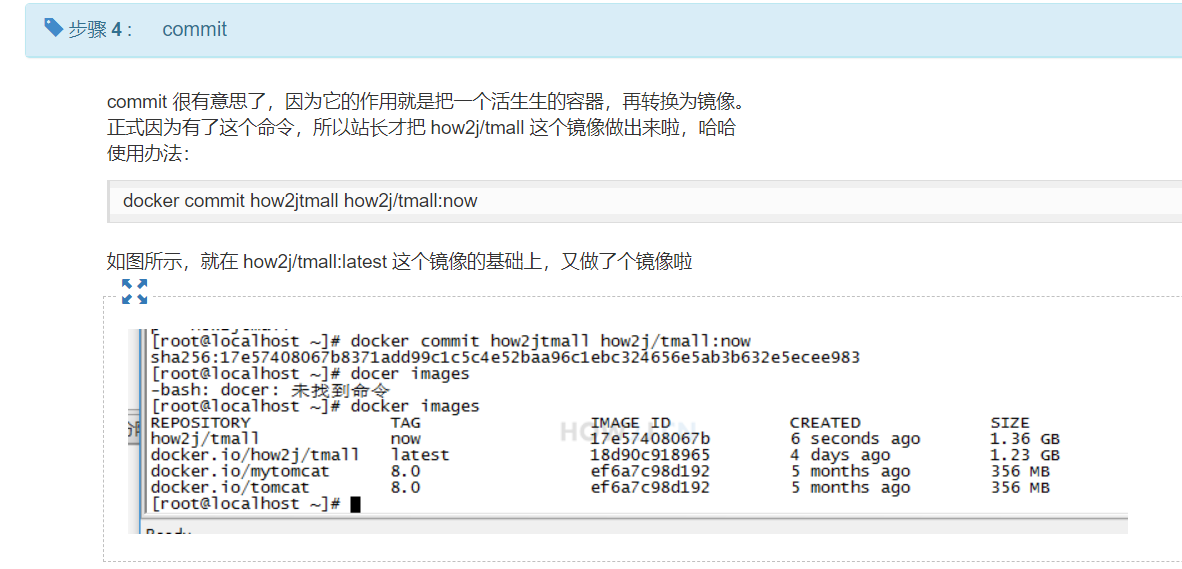
5. 检查某个具体的容器

6. rm 删除容器 （删除容器之后才能删除镜像）

7. commit，对容器做了修改后，把改动后的容器，再次转换为镜像



Commit:容器转镜像



暂停：pause (暂停运行容器，外面访问不了了)  
恢复：unpause (又开始运行容器，外面可以访问)  
停止：stop （停掉所有容器内的服务，tomcat，mysql等）  
开始：start (这个停掉之后再启动所有服务，得等一小会才能看到结果)  
就是常见的生命周期命令，没什么好说的。。。  
需要注意的是， stop 之后再 start, 容器需要启动，tomcat也需要启动，里面的mysql 也需要启动，都很花时间，所以要等待十几秒再访问，才能看到结果，否则会误以启动失败了

PS：

Docker ps –a 查询所有

Docker ps 查询run状态的和pause状态的（在线上的），stop的不查



删除容器：



docker rm `docker ps -a -q` -f

删除所有容器

# 6.网络恢复（重启问题）

重启虚拟机后不能访问Doceker的解决：

虚拟机一旦重启，或者关机过，那么就会出现可以访问 Linux，但是无法访问里面的 Docker的情况。 可以在 CentOS里做如下事情：

vi /etc/sysctl.conf

或者

vi /usr/lib/sysctl.d/00-system.conf

添加如下代码：

net.ipv4.ip\_forward=1

重启network服务

systemctl restart network

查看是否修改成功

sysctl net.ipv4.ip\_forward

如果返回为“net.ipv4.ip\_forward = 1”则表示成功了



# 7.提交镜像：

流程：登录 –>拉下来一个操作系统(linux) -->在这个环境中安装你的东西，mysql，tomcat，java啥的搭建环境🡪 将这个拉下来的操作系统（运行时叫做容器），打成一个镜像 –>提交到自己的docker hub

最主要是要拉下来一个操作系统，再进行环境搭建操作，最后将整个操作系统放进hub。

1.首先登陆：

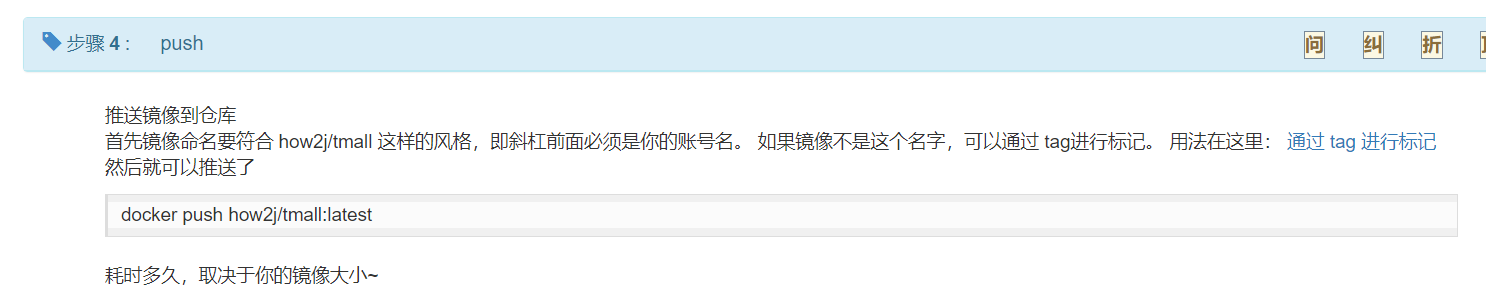
docker login

账号：animal2333

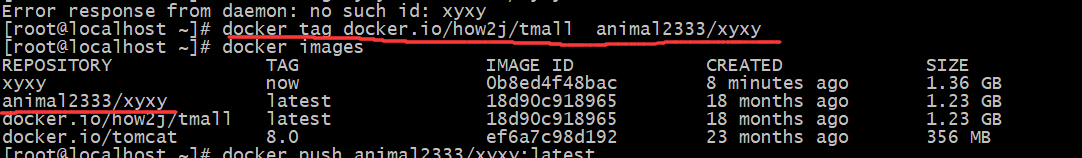
密码：keke3237!

然后推送镜像，注意这个镜像的命名很讲究，必须要是“自己的id”/xxx（自定义）

如:animal/xyxy



Tag出符合格式的image：



然后上传：

docker push animal2333/xyxy:latest

