**docker基础**

1.简介

docker是google基于go语言开发、实现轻量级的操作系统虚拟化的解决方案。和我们之前使用的KVM、VMware、Xen相比，好处很多。



2.基本概念

镜像：docker镜像是一个只读的模板，和我们下载Windows操作系统的镜像一样，可以用它来创建docker容器。

容器：由镜像创建的运行实例。容器可以被创建、启动、开始、停止、删除，简单地说，容器就是一台虚拟机。

仓库：存放镜像文件的场所。和git仓库有很强的相似之处，可联系理解。

仓库有公有仓库和私有仓库的区别，最大的公有仓库就是Docker Hub 。一般来说，我们都会自建私有仓库，后面会详细说明私有仓库的创建和使用。

仓库注册服务器：这个概念常常和仓库混为一谈，不严格区分。实际上，仓库注册服务器上往往存放多个仓库，每个仓库中包含多个镜像，每个镜像用tag来区分。

安装（仅介绍centos7系统下的docker安装）

由于centos7系统的extras库中docker安装包，可直接安装：

》》yum install docker

安装后启动docker服务，并开机自启

》》service docker start

》》chkconfig docker on

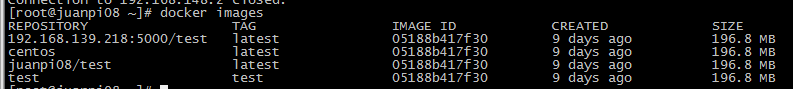
3.镜像

首先要获取镜像，docker pull 命令会从docker hub仓库下载一个centos操作系统的镜像

》》docker pull centos

显示本地已有镜像

》》docker images

REPOSITORY：镜像来自哪个仓库

TAG ：镜像标记，用来标记同意仓库的不同镜像

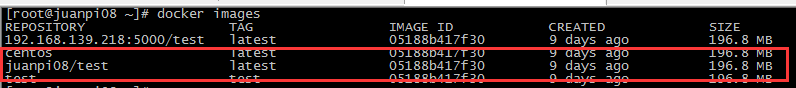
IMAGE ID ：镜像ID（有唯一性），ID相同就是同一镜像

CREATED：镜像创建时间

SIZE：镜像大小

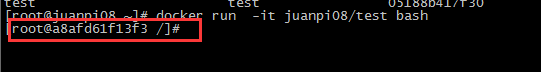
创建镜像

可以使用docker hub上获取已有镜像，然后基于该镜像启动容器，在容器中添加应用，然后退出容器，并提交修改。



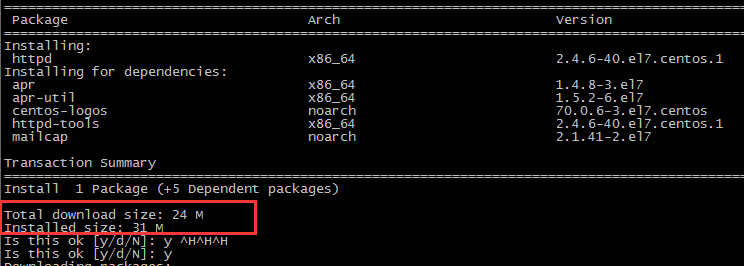
基于juanpi08/test这个镜像启动容器，-i选项

》》docker run -it juanpi08/test bash



a8afd61f13f3 是容器的ID的部分

》》yum install httpd -y

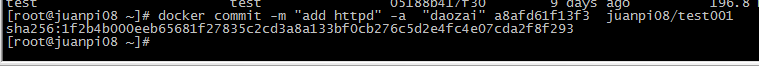


安装完后exit退出

》》exit

此时容器已经被我们修改了，使用docker commit命令来提交更新后的副本

》》docker commit -m “add httpd” -a “daozai” a8afd61f13f3 juanpi08/test001



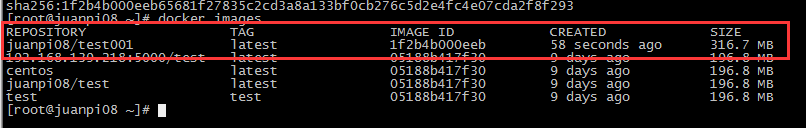
-m 描述，自定义

-a 作者，自定义

a8afd61f13f3 是容器ID

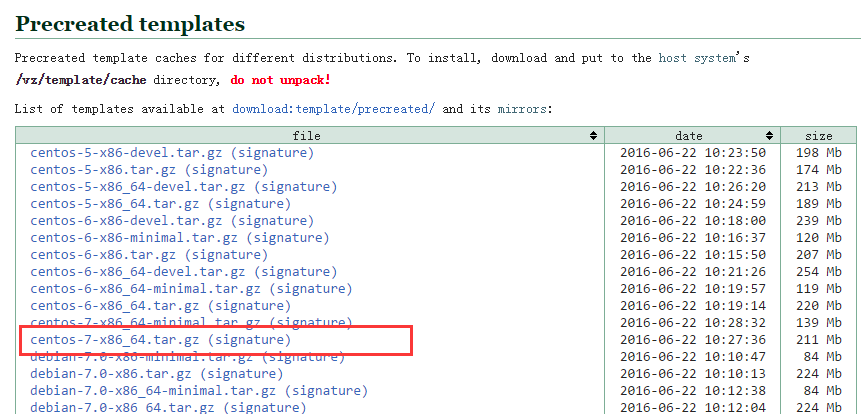
juanpi08/test001 镜像名称，自定义

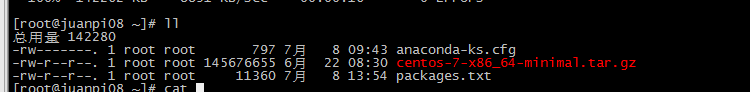
再用docker images 查看本地已有镜像，也可以用该镜像来启动容器，这时候容器内就已经有装好的httpd服务了。



docker push ，可以将本地镜像推到远端的仓库，在后面会结合仓库一起说。

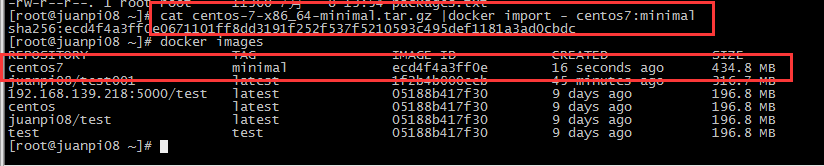
关于docker镜像，除了通过docker pull从docker hub拉取镜像之外，还可以通过下载镜像文件来导入镜像 ，下载地址：https://openvz.org/Download/template/precreated



框中部分是我下载的一个版本，服务器在国外，下载速度很慢。

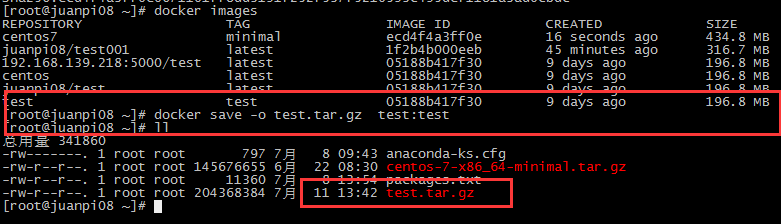
下载好了之后，使用docker import来导入镜像

》》cat centos-7-x86\_64-minimal.tar.gz |docker import - centos7:minimal



还可以将本地镜像保存成镜像文件的形式，使用docker save 命令

》》docker save -o test.tar.gz test:test



还可以用docker load 从导出的本地镜像文件再导入到本地镜像库中

》》docker load --input test.tar.gz

或

》》docker load < test.tar.gz

移除本地镜像

》》docker rmi

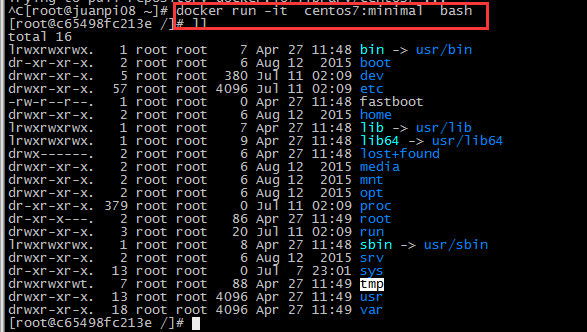
注意区别docker rmi和docker rm ，前者是移除镜像，后者是移除容器。如果某一镜像还有在被使用，该镜像就不能够被删除，也就是说在删除某一镜像前，需要将依赖于该镜像的容器先删掉。

4.容器

前面提到，容器就是一个虚拟机，更完整地说，容器是独立运行的一个或一组应用，以及它们的运行环境。

新建并启动容器

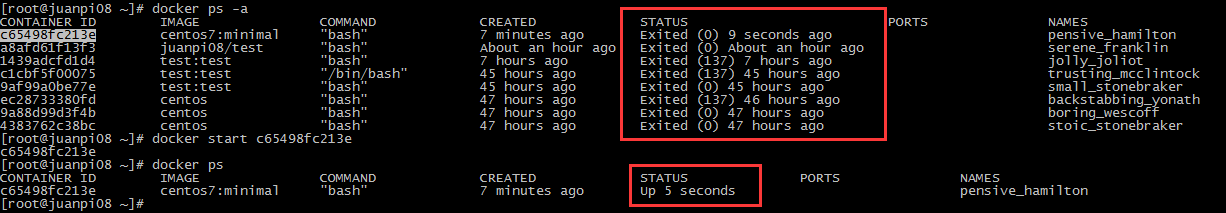
》》docker run -it centos7：minimal bash



-i 选项是让容器的标准输入保持打开，-t选项让docker分配一个伪终端并绑定到容器的标准输入上；

启动已终止容器

》》docker start container\_ID



守护态运行容器

更多时候，需要让容器在后台已守护态姓氏运行，可以通过-d参数实现

》》 docker run -d michael /bin/bash -c "while true; do echo hello world; sleep 1; done"

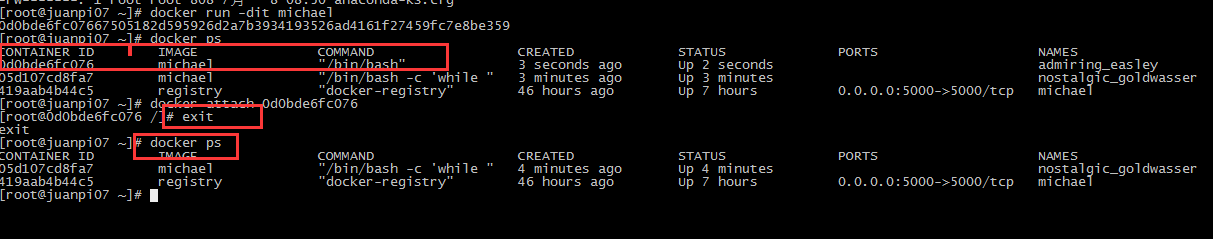
终止容器

》》docker stop container\_ID

进入容器

》》docker attach container\_ID

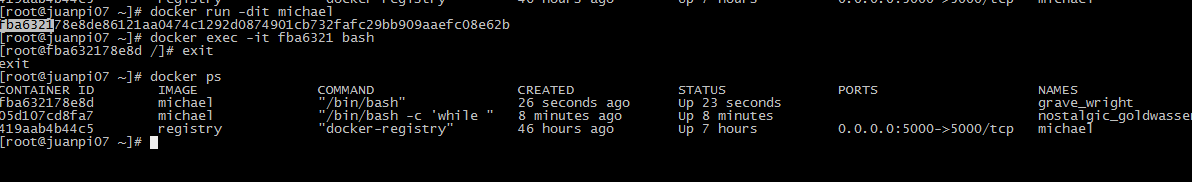
》》exit



使用docker attach 进入容器，退出后该容器退出，不建议使用docker attach进入容器

》》docker exec -it container\_ID bash

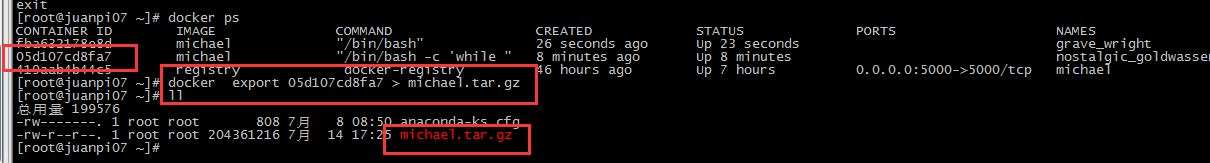
》》exit



退出容器后，容器继续正常运行。

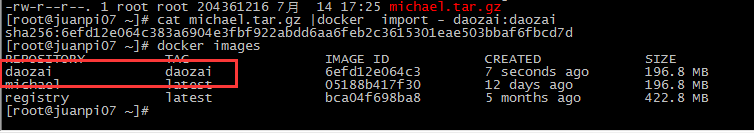
导出容器，应用在容器的迁移上，十分有用。

》》docker export 05d107cd8fa7 > michael.tar.gz



导入容器

》》cat michael.tar.gz | docker import - daozai：daozai

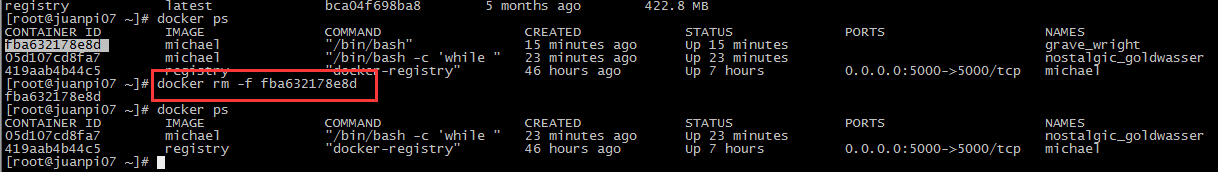


用户既可以使用 docker load来导入镜像文件到本地镜像库，还可以通过docker import来导入容器快照到本地镜像库。

删除容器

》》docker rm container\_ID

删除处于终止状态的容器，加上-f选项可以删除处于运行状态的容器。



5.仓库

仓库就是存放镜像的地方。

公有仓库需要先到docker hub上注册账号，这里就不说了，感兴趣的童鞋可以自己去玩玩。

这里主要说怎么自建私有仓库

192.168.139.218 这台机器上搭建私有仓库

》》docker pull registry

》》docker run -d -p 5000:5000 --name registry -v /data/docker:/tmp/registry registry

在其他机器往私有仓库提交镜像

》》docker tag image\_name 192.168.139.218:5000/image\_name

》》docker push 192.168.139.218:5000/image\_name

这样就把镜像提交到私有仓库了