### 前提

在前一篇文档中已经完成了Jenkins构建后的SSH远程登录传输了，现在我们利用这个来完成整合docker

### Docker安装使用

首先要说明一下什么是docker以及docker的基本用法，首先要说明几个概念，docker中有两个非常重要的概念：镜像和容器。网上对此的解释很多，我的理解就是，容器就是一个虚拟机，镜像就是这个虚拟机的系统镜像。

docker可以用简单的脚本来完成镜像的设计，以达到专门用于某种用途的镜像，而docker的容器可以与外部连接，在容器创建时可以用-v来共享目录，用-p来完成虚拟机和宿主机的端口对应。

Docker适用于Windows、linux、mac系统，安装流程各异，本实例用centOS7.4， 运用最简单的yum完成安装：yum -y install docker

由于各种原因，建议现在宿主机安装好jdk环境

然后启动：service docker start

现在已经完成了docker这个软件的安装启动，然后开始下载tomcat镜像：docker pull tomcat

然后进入关键步骤，启动一个该镜像的容器：

docker run -d --name HelloDocker -p 8080:8080 -v /usr/java/apache-tomcat-7.0.84/webapps/:/usr/local/tomcat/webapps/ tomcat

这一行命令我们要进行一下解释：

run是启动新容器的命令

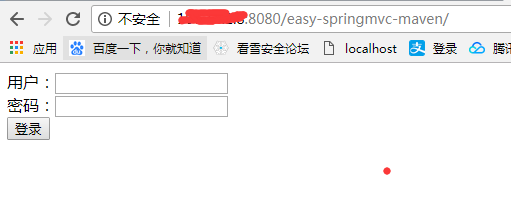
-d是后台运行

--name是定义容器的名字，-p 8080:8080是对应宿主机和容器的端口，由于容器是pull下来的tomcat镜像，因此它的运行端口主要就是8005、8009、8080，注意前面那个是宿主机的端口

-v是目录的共享，前面是宿主机目录，由于要把宿主机的war包运行在容器里，必须共享目录，然后为了方便起见，干脆下载一个tomcat，用它的webapp做共享目录（当然也可以不这样）

最后一个tomcat是空格隔开的要注意，这个是镜像的名字，启动容器要指定好镜像。

然后tomcat就启动了：



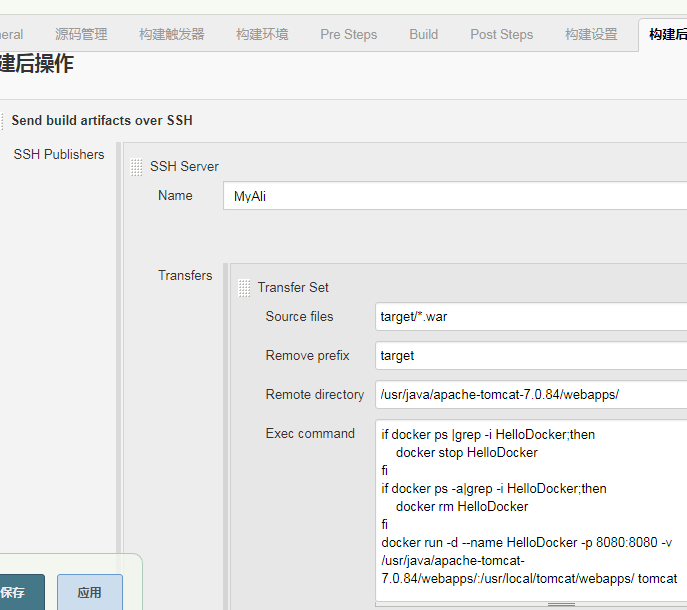
我们可以利用docker ps命令查看正在运行的容器，用docker ps -a查看所有容器

用docker stop停止容器，docker rm移除容器，但是移除的容器不能正在运行，也就是说要移除一个启动容器必须先stop再rm。此外不能同时出现相同名字的容器。

关于docker的详细情况可以查阅资料，这里不细说了。

### 整合进入Jenkins

接下来就简单了，利用上一篇的结果去完成，进入构建后操作：



这里吧war包移到远程主机的webapps下，然后利用刚刚的启动命令去启动容器，但是启动前需要先停止同名容器，利用条件语句完成。

整个脚本如下：

if docker ps |grep -i HelloDocker;then

docker stop HelloDocker

fi

if docker ps -a|grep -i HelloDocker;then

docker rm HelloDocker

fi

docker run -d --name HelloDocker -p 8080:8080 -v /usr/java/apache-tomcat-7.0.84/webapps/:/usr/local/tomcat/webapps/ tomcat

### 小结

1. 目前的整合是每次构建完成以后去新创建一个容器，干净但是速度不一定快，是否有更好的配置，需要具体分析。事实上网上关于Jenkins整合docker的配置五花八门，归根结底只要Jenkins和docker用的熟练，会写shell脚本，怎么配置都是随意的。
2. 运用docker的好处，这个问题我有点困扰，事实上我没觉得有太大好处，充其量每个应用运行在各自的容器里，容器有着各自的镜像，整个更规范，即使有一个崩了也不会影响宿主机，一台宿主机可以运行多个tomcat的容器（只要开放的端口和容器名称不同）等等。