# Docker-搭建私有化镜像仓库

在前面的学习中，已经接触了各种各样的Docker镜像：有一些时别人制作好的：比如redis、mysql、nginx镜像，还有一些是我们基于微服务去构建的镜像。这些镜像一定需要有一个地方做统一地保存和管理，这个地方就是Docker镜像仓库，之前的Docker是公共仓库，本次学习如何在本地搭建私有镜像仓库，已经如何在本地私有镜像仓库完成镜像的管理！

## 镜像仓库Docker Repository：

### 公有仓库：

如Docker官方的Docker Hub，国内一些云服务上提供了类似于Docker Hub的公开服务，如网易云镜像服务、阿里云镜像服务等；

### 私有仓库：

本地搭建私有Docker Registry，为什么需要私有的镜像仓库呢？你想企业基于微服务构建的镜像，属于企业的商业机密，把他放在公共的仓库里，就毫无隐私可盐，显然是不安全的，所以企业自己构建的镜像，一般情况下都需要放到私有仓库，而这个私有仓库就需要自己去搭建了！

具体查看MarkDown文档：Centos7安装Docker

## 搭建Docker镜像仓库

搭建仓库可以基于Docker官方提供的DockerRigistry来实现！

我们来实现搭建带图形化界面版本的Docker镜像仓库

图形化界面并不是Docker官方提供的，是由第三方的个人在官方的DockerRigistry基础上开发的，所以是一个额外的服务，所以我们通过DockerCompose来把两者进行组合部署

使用DockerCompose部署带有图像界面的DockerRegistry，命令如下：

**version: '3.0'**

**services:**

**registry:**

**image: registry**

**volumes:**

**- ./registry-data:/var/lib/registry**

**ui:**

**image: joxit/docker-registry-ui:static**

**ports:**

**- 8080:80**

**environment:**

**- REGISTRY\_TITLE=黄远辉的Docker镜像私有仓库**

**- REGISTRY\_URL=http://registry:5000**

**depends\_on:**

**- registry**

解释：第一个是就是官方的DockerRegistry，镜像名就是registry；

第二个就是图像化界面，端口向外开发8080，然后这个服务要依赖第一个registry服务，将来DockerCompose部署的时候就先启动registry，再启动图形化界面服务，而后还配置了两个环境变量，第一个私有仓库名，第二个配置了registry服务的地址，因为ui是依赖于上面的registry服务的，所以要把registry的地址告诉ui，可以看到地址的写法：**<http://registry:5000> ,**5000是registry容器的端口，没什么好说的，端口前面是服务名registry，之前聊过，再DockerCompose中，多个服务之间可以用服务名互相访问！所以这样ui控制台就知道了registry的地址了！但是注意5000端口只是两者之间内部访问端口我们是无法访问的，我们只能访问8080

将来我们只需要把这个DockerCompose文件执行就可以搭建私有仓库了，但是在这还要先配置docker的信任地址！

因为我们在搭私服的时候，没有去给私服配置https协议，所以Docker默认是不信任私服的，需要你去添加一个信任地址！

第一步：

### 配置Docker信任地址

#### # 打开要修改的文件

vi /etc/docker/daemon.json

#### # 添加内容：

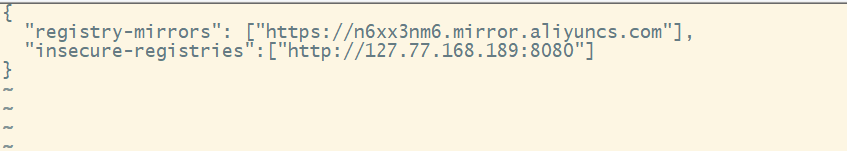
"insecure-registries":["http://120.77.168.189:8080"]

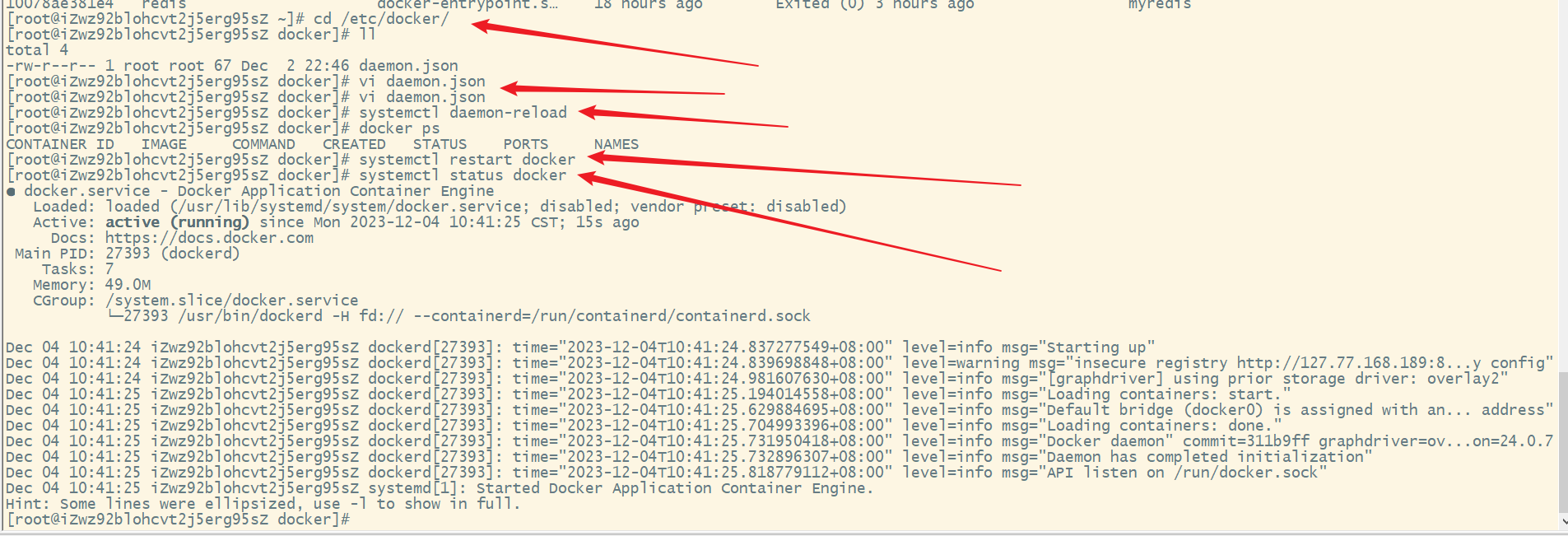
#### # 重加载

systemctl daemon-reload

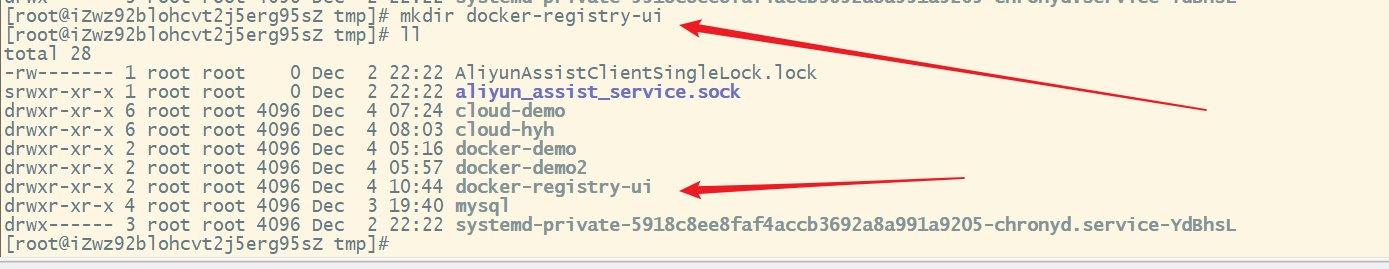
#### # 重启docker

systemctl restart docker

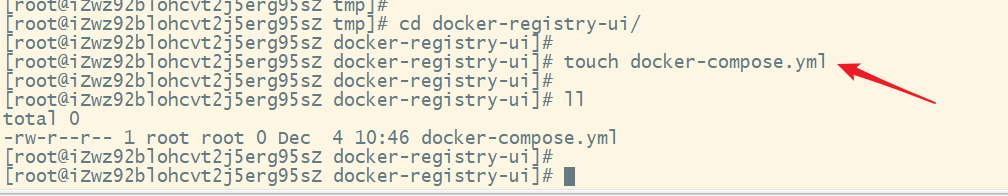




### 在/tmp目录下新建docker-registry-ui目录



### 在docker-registry-ui目录下，新建docker-compose.yml文件



### 修改compose命令：vi docker-composr.yml

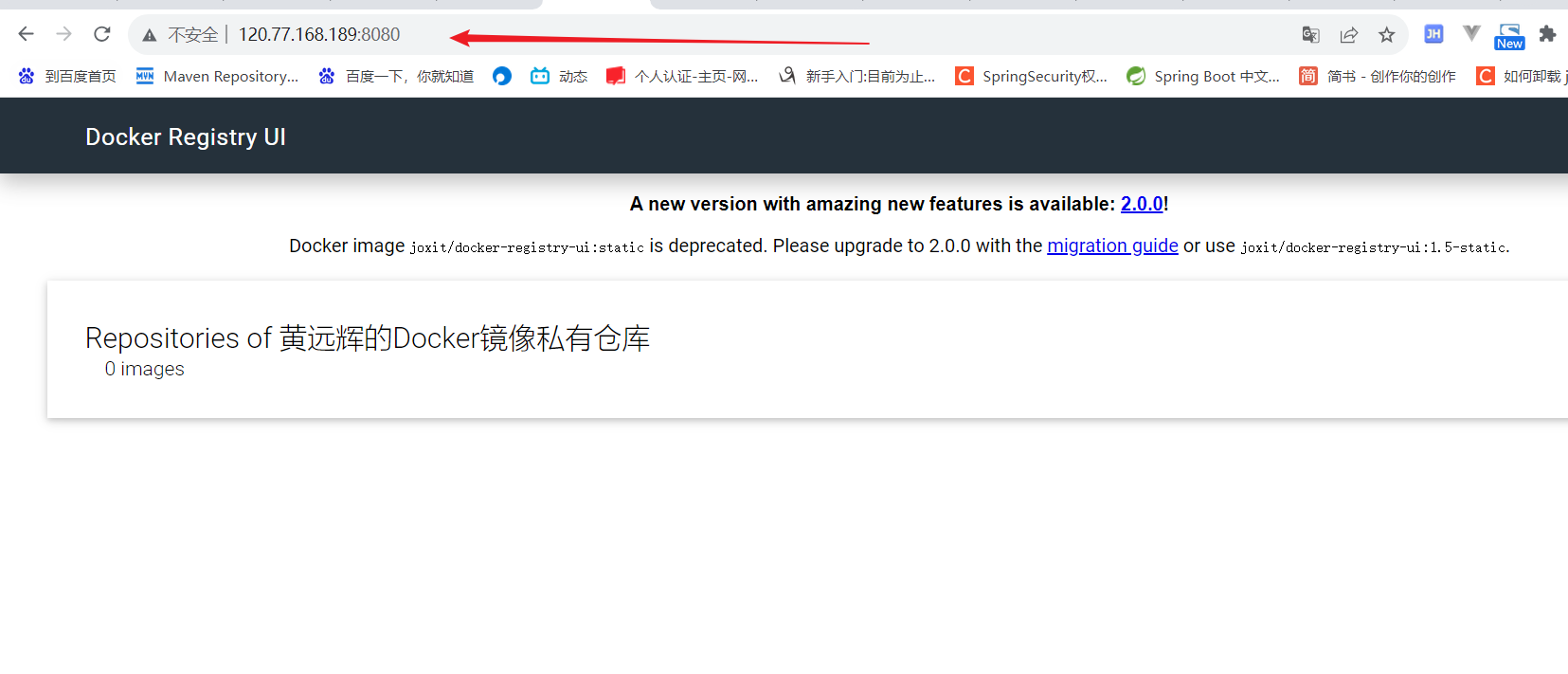


### 执行DockerCompose部署：docker-compose up -d



当然你可以使用docker-compose logs -f查看一下日志！

### 浏览器访问图形化界面

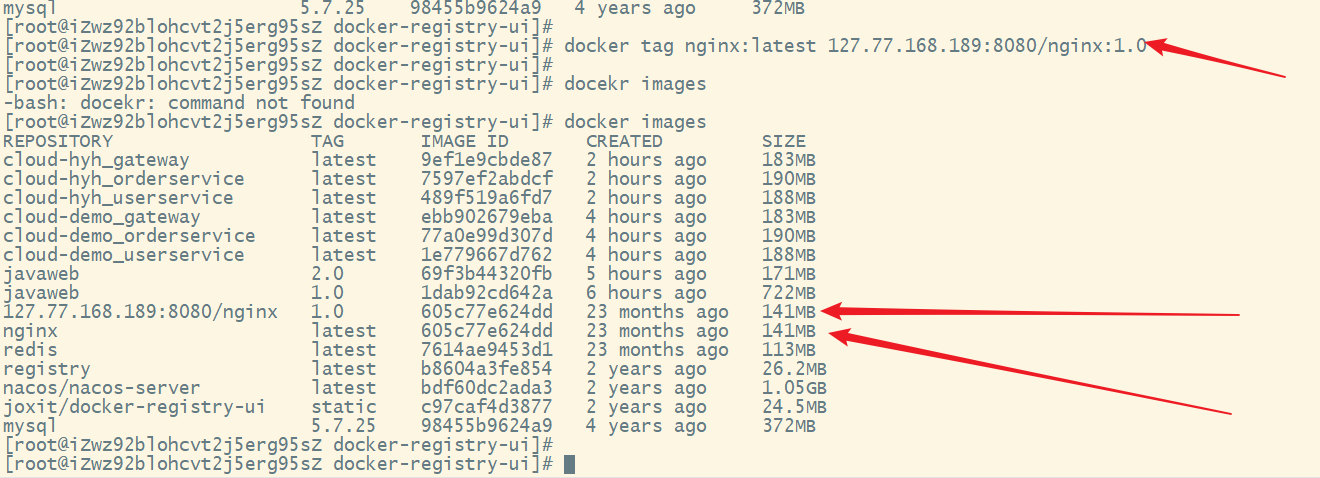


0images，说明目前没有任何镜像！

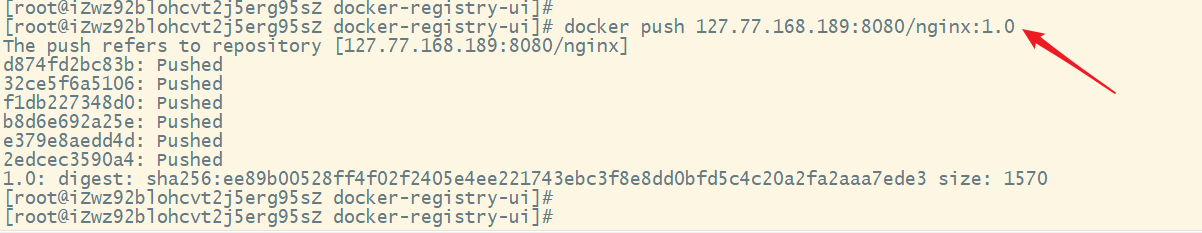
## 上传镜像到私有镜像仓库

要想推送镜像到私有镜像仓库，必须先对本地镜像打tag

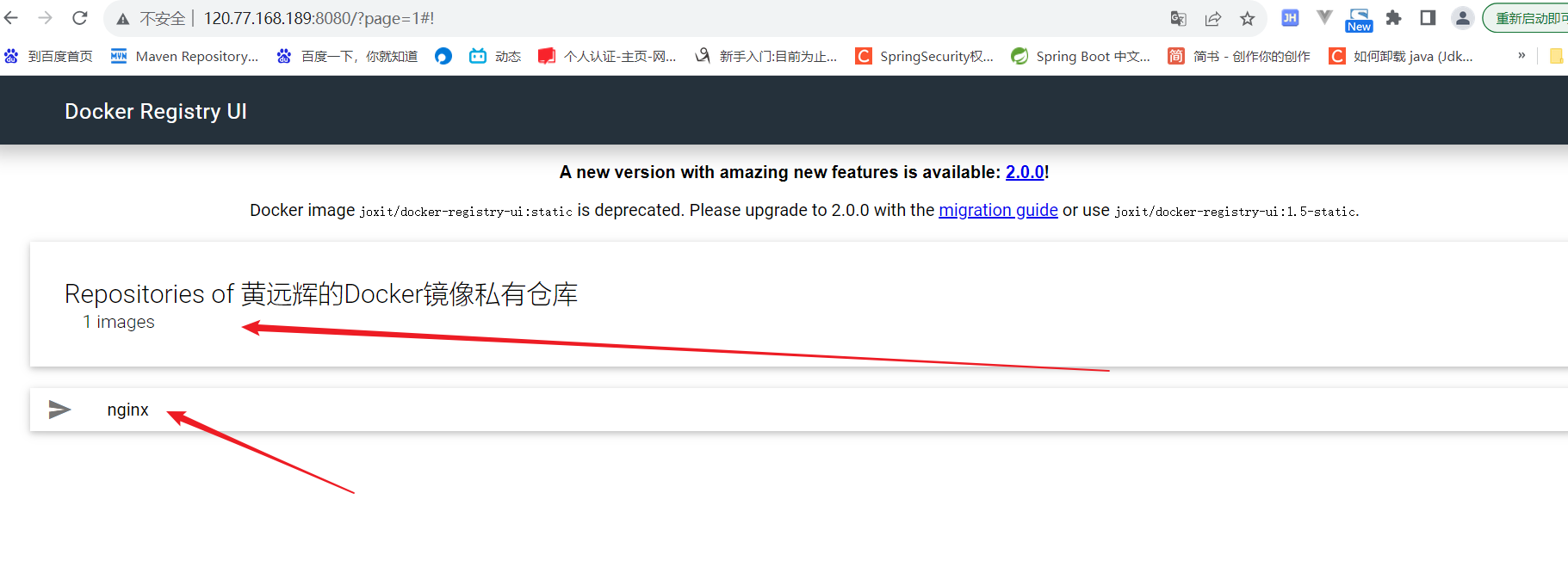
### 1：一定要重新tag本地镜像，就是给镜像重新取名字，名称前缀为私有仓库的地址：127.77.168.189:8080/

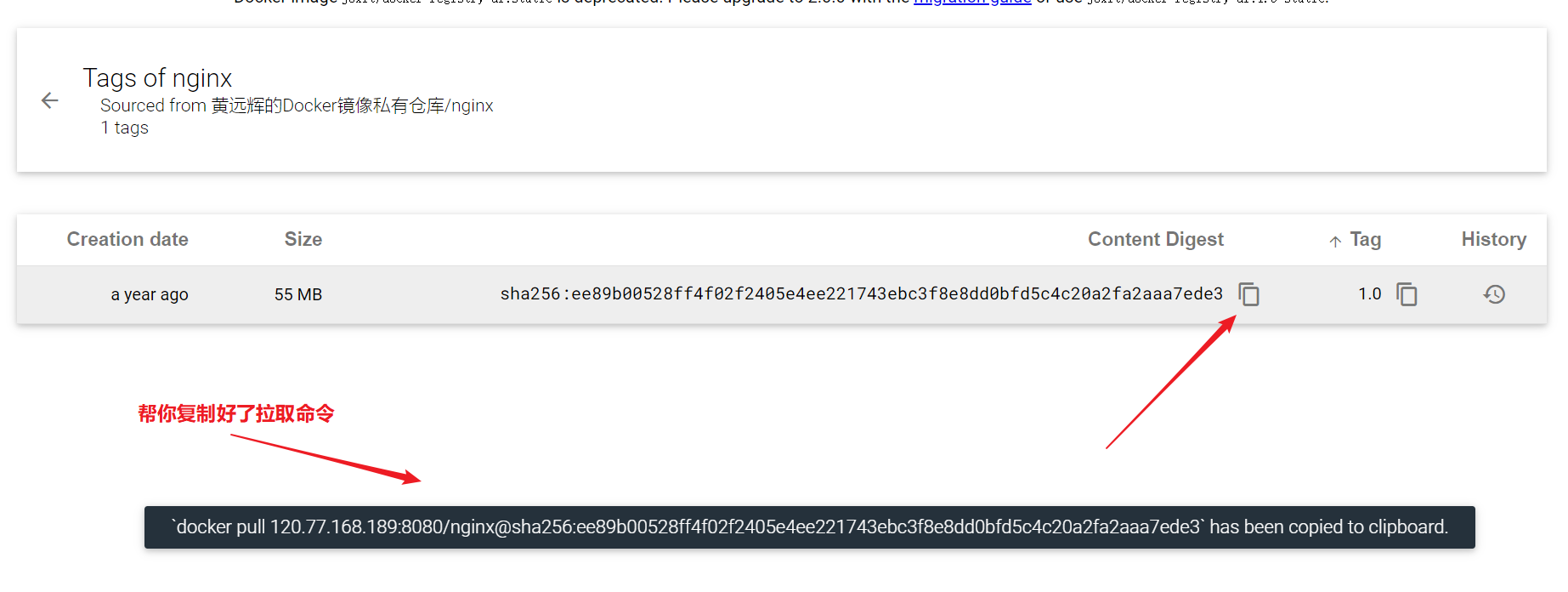


### 2：推送镜像

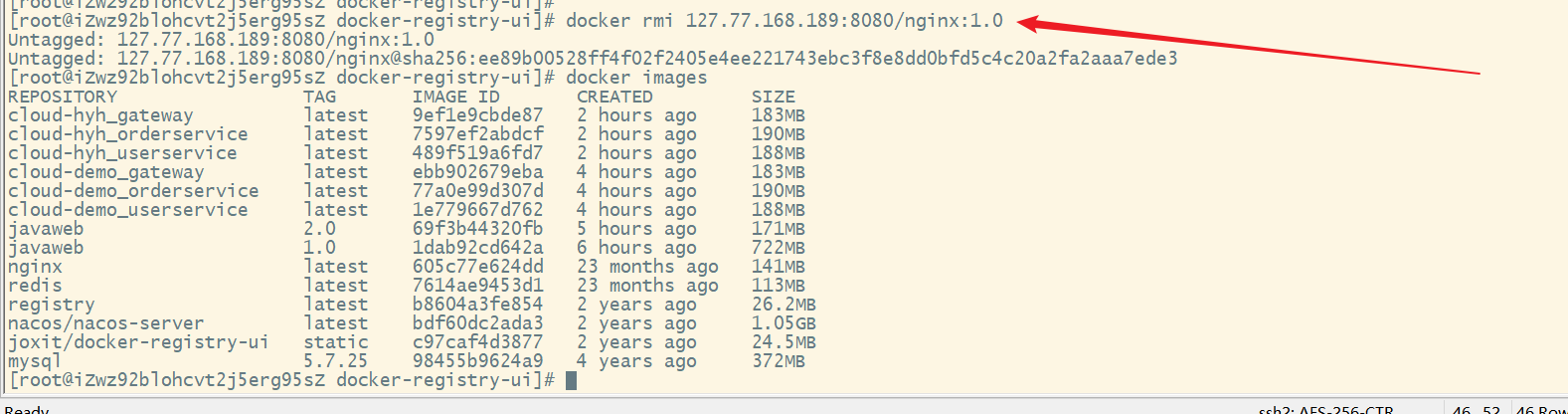


### 3：图形界面查看

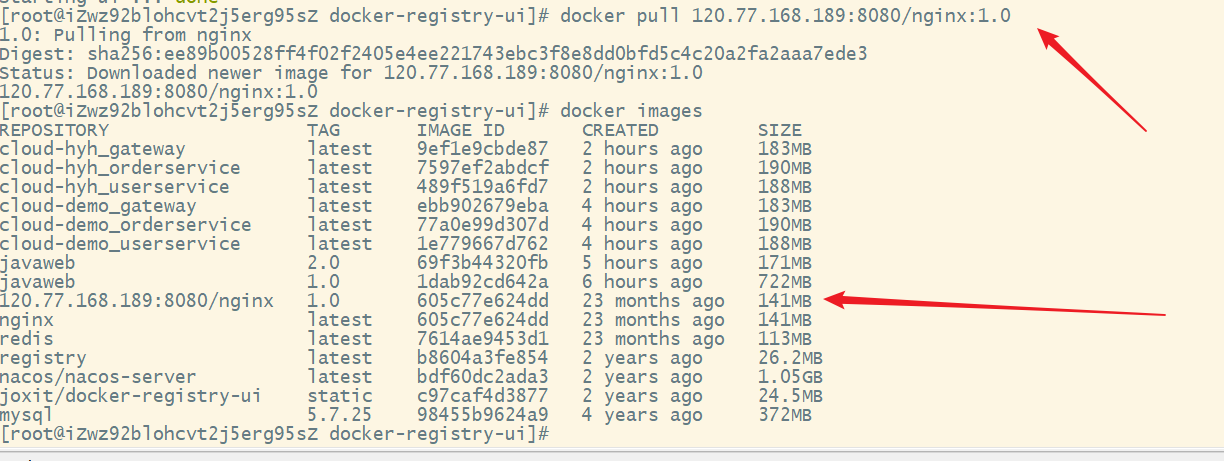




### 4：拉取镜像前最好先将本地的对应镜像删除：



### 5：从私有镜像仓库拉取镜像



## 小总结

1：推送本地镜像到仓库前必须重命名（docker tag）镜像，以镜像仓库地址为前缀；

2：镜像仓库推送前需要把仓库地址配置到docker服务的daemon.json文件中，被docker信任；

3：推送使用docker push命令