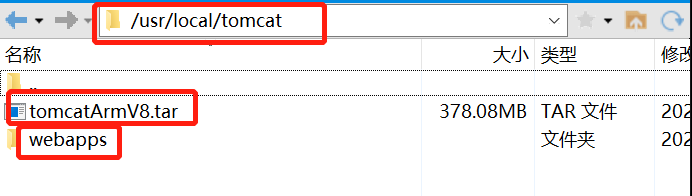
**Arm服务器Docker离线部署Tomcat**

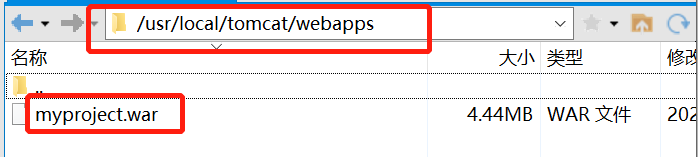
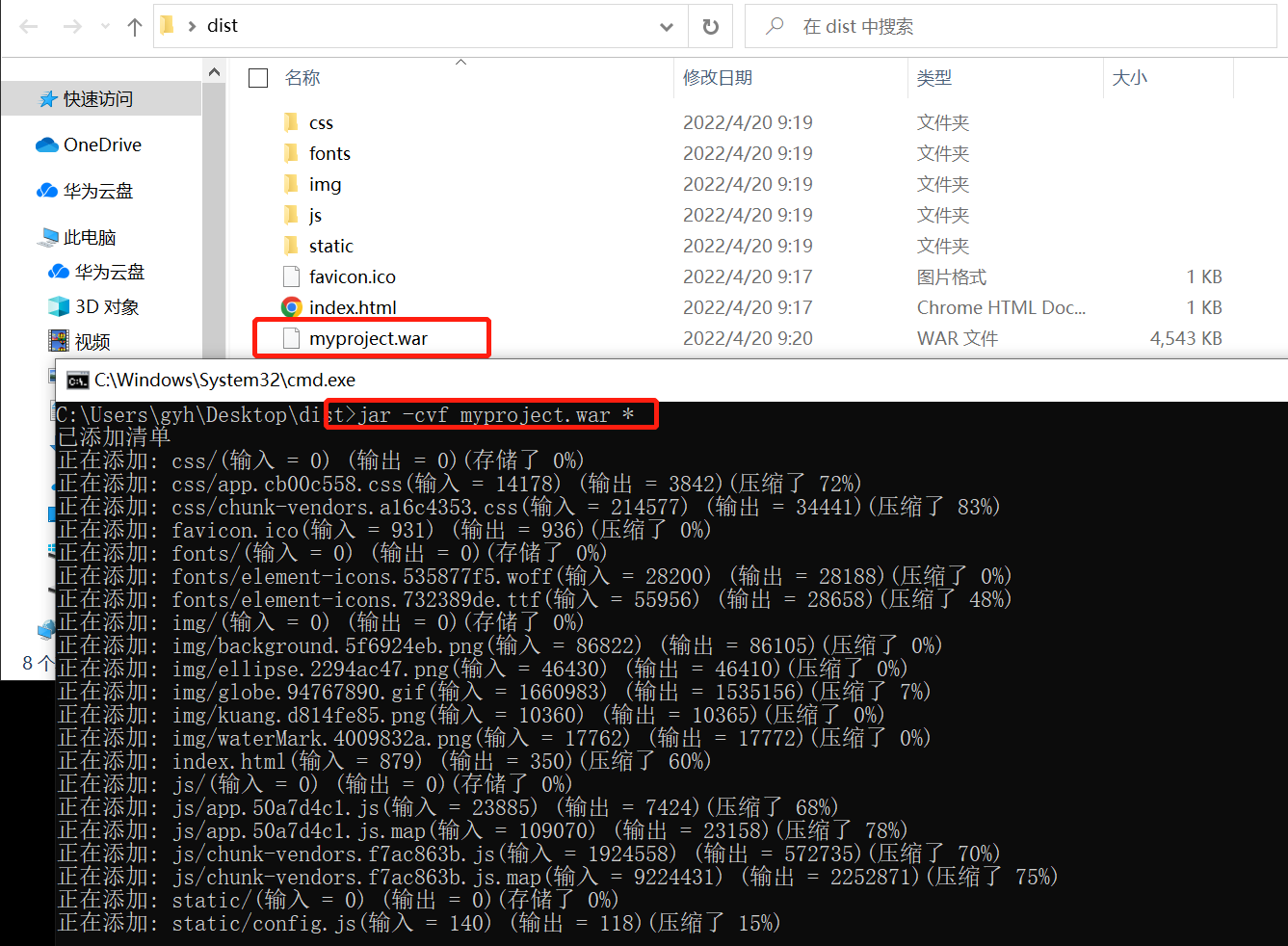
**———工具：xshell和xftp**

# Tomcat部署

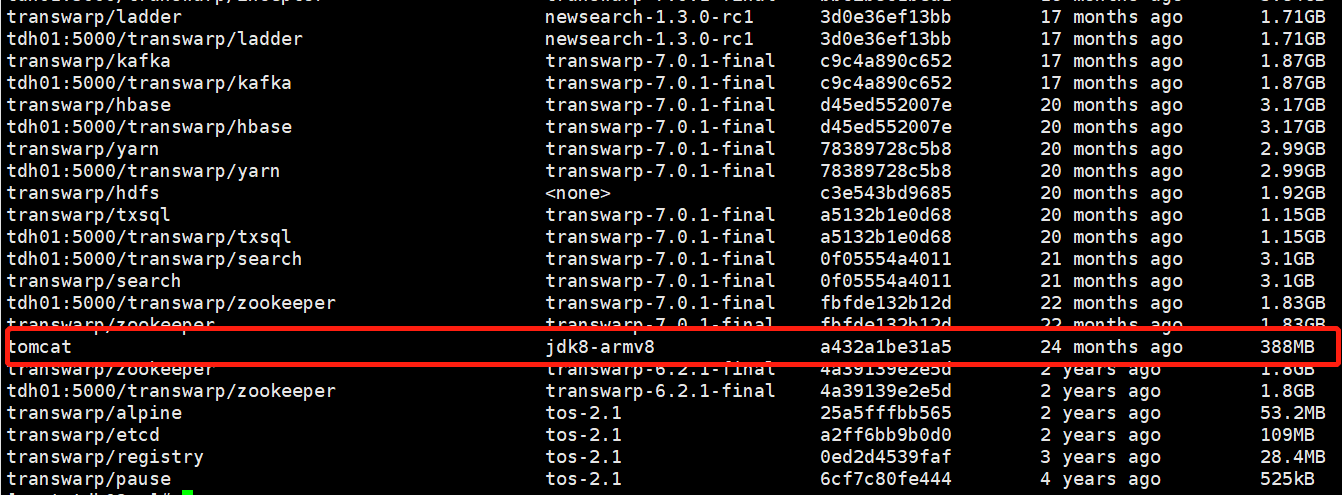
* 利用xftp上传tomcat镜像至服务器/usr/local/tomcat下并创建webapps文件夹（为了映射到容器中的tomcat）



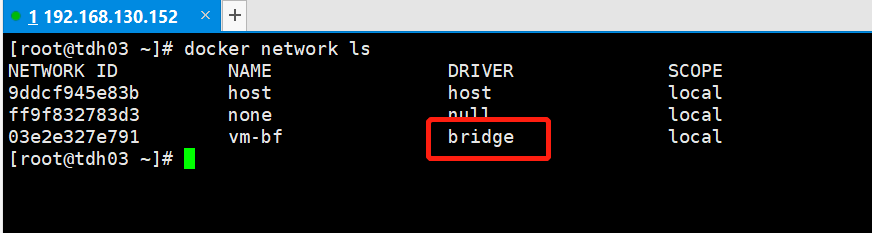
* 利用xftp上传前端war包至/usr/local/tomcat/webapps下（在dist文件cmd输入jar -cvf 文件名.war \*生成war包）



* 利用xshell执行docker load -i 镜像包文件名加载镜像，docker images查看所有镜像



* 利用xshell输入docker network ls查看是否有bridge网桥

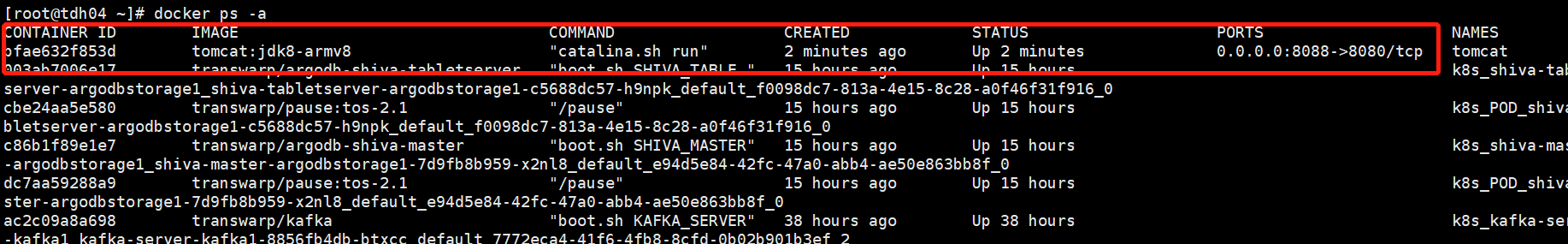


* 利用xshell输入docker network create -d bridge vm-bf创建名为vm-bf的bridge网桥（如果有bridge网桥跳过此步）
* 利用xshell输入docker run -di --name=tomcat -p 8088:8080 --network=vm-bf -v /usr/local/tomcat/webapps:/usr/local/tomcat/webapps tomcat:jdk8-armv8（docker run -di --name=容器名字（自定义） -p 外部端口:内部端口（与tomcat配置端口一致） --network=bridge网络模式名字 -v 宿主机映射文件位置：docker下tomcat要映射的文件位置 镜像名字:TAG）
* 启动成功

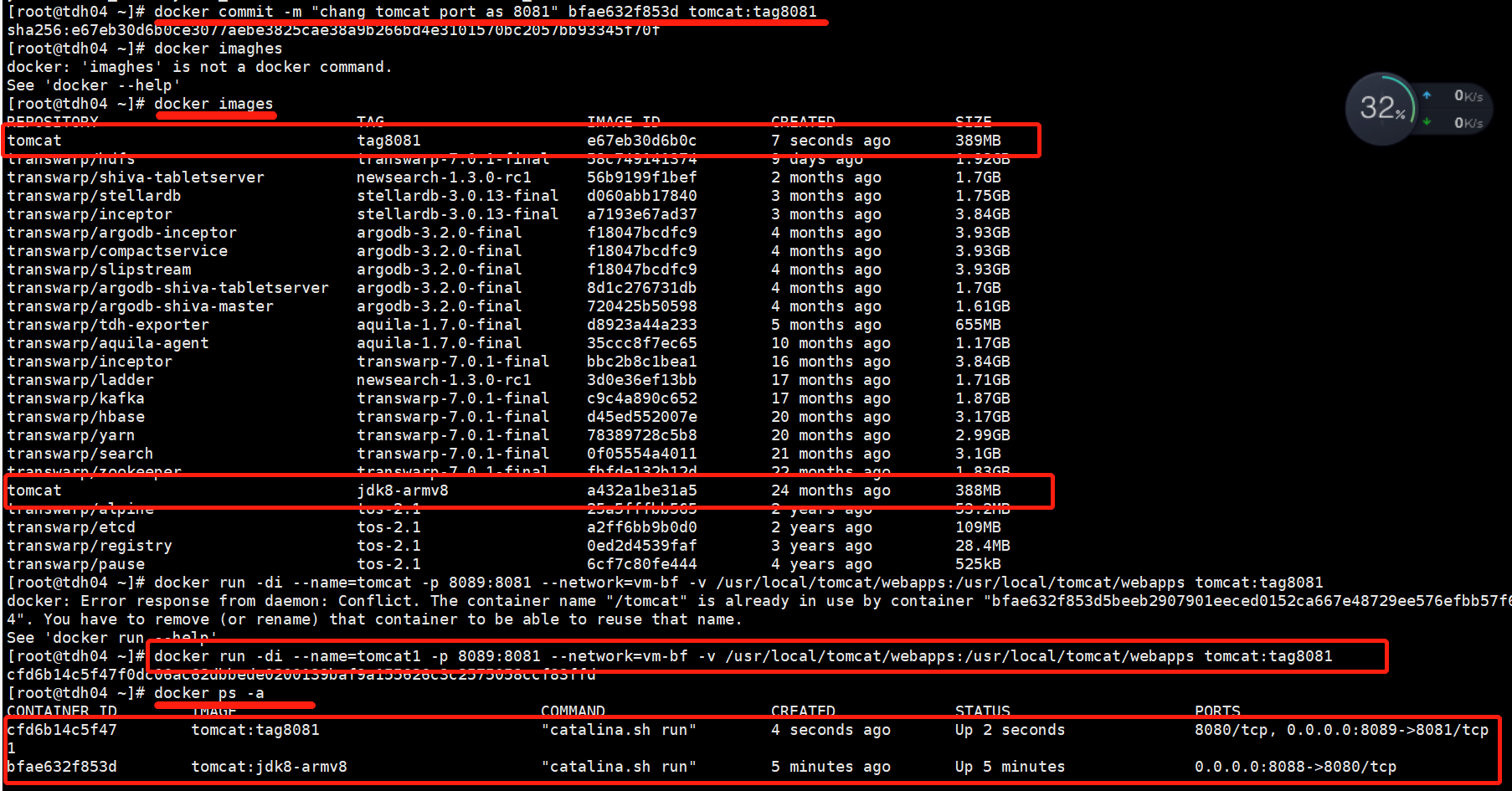
访问地址+端口号+war包名（192.168.130.150:8088/myproject）

# Docker下更改Tomcat端口并打包成新镜像

* 利用xhsell输入docker ps -a查看正在运行的容器ID



* 利用xhsell输入docker exec -it 容器ID /bin/bash进入容器查看tomcat配置文件位置，并执行exit退出容器
* 利用xhsell输入docker cp 容器名字:/usr/local/tomcat/conf/server.xml server.xml（将容器下的server.xml拷贝到宿主机）
* 利用xhsell修改宿主机下的server.xml端口为8081（执行netstat -anp | grep 8081查看端口是否被占用）并执行docker cp server.xml tomcat:/usr/local/tomcat/conf/server.xml（将宿主机下server.xml拷贝覆盖容器下的tomcat配置文件）
* 利用xshell执行docker commit -m "chang tomcat port as 8081" tomcat的容器id tomcat:tag8081打包成新的镜像并执行新的tomcat容器
* 利用xshell执行docker run -di --name=tomcat1 -p 8089:8081 --network=vm-bf -v /usr/local/tomcat/webapps:/usr/local/tomcat/webapps tomcat:tga8081



* 启动成功

访问地址+端口号+war包名（192.168.130.150:8089/myproject）

# 导出镜像

docker save 镜像ID> 镜像名字（自定义）.tar

