**《云计算技术》**

**本科生课程实验报告**

|  |  |
| --- | --- |
| **题目：** | **Devops环境搭建** |
| **专业：** | **数学与应用数学** |
| **姓名：** | **吕祥宜** |
| **学号：** | **2019302020127** |

**武汉大学计算机学院**

**2022年 11月 22日**

**郑 重 声 明**

本人呈交的实验报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本实验报告不包含他人享有著作权的内容。对本实验报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本实验报告的知识产权归属于培养单位。

本人签名： 吕祥宜 日期： 2022.11.22

**目 录**

## **1** **概述**

1.1 选题背景与实验目的………………………………1

1.2 实验内容 …………………………………………1

## **2** 方案设计与实现

2.1 系统架构设计 …………………………………… 2

2.2 系统技术选型与相关工具…………………………2

## 3 项目搭建过程

3.1 部署gitlab…………………………………………3

3.2 部署jenkins…………………………………………4

3.3 配置jenkins…………………………………………6

3.4 实现基础的CI操作………………………………8

3.5 部署harbor…………………………………………14

## 4 总结

4.1 小结…………………………………………21

4.2 项目存在的问题与不足……………………………21

4.3 项目展望………………………………………21

## 5 学习体会

5.1 项目相关技术学习体会…………………………………21

5.2 云计算技术的行业应用体会……………………………21

## 参考文献……………………………22

# 1 概述

## 1.1 选题背景/需求分析

DevOps是一套结合软件开发（Dev）和IT运营（Ops）的系统。它的目的是缩短系统开发周期，并提供具有高软件质量的持续交付。本实验基于Devops的理念，以jenkins工具为中心，实现了应用提交之后的部署自动化以及镜像版本的管理。

## 1.2实验内容

本实验实现了一个spring web工程的简单自动化部署.

# 2 系统设计与实现

## 2.1 系统架构设计

## 

首先通过git将工程代码push到准备好的gitlab服务器上，通过jenkins拉取代码并用maven构建成jar包。用jenkins中配置好的docker将jar包构建成镜像并上传到harbor服务器上。然后jenkins通知目标服务器从harbor上拉取镜像并运行成容器，完成整个自动化部署流程。

## 2.2 系统技术选型与相关工具

Vmware虚拟机安装docker，docker-compose，并用docker安装jenkins,gitlab,harbor容器。项目使用 java语言和IDEA环境。

# 3 项目搭建过程

## 3.1:部署gitlab

拉取gitlab镜像：docker pull gitlab/gitlab-ce

准备docker-compose.yml文件：

|  |
| --- |
| version: '3.1'  services:  gitlab:  image: 'gitlab/gitlab-ce:latest'  container\_name: gitlab  restart: always  environment:  GITLAB\_OMNIBUS\_CONFIG: |  external\_url 'http://192.168.11.11:8929'  gitlab\_rails['gitlab\_shell\_ssh\_port'] = 2224  ports:  - '8929:8929'  - '2224:2224'  volumes:  - './config:/etc/gitlab'  - './logs:/var/log/gitlab'  - './data:/var/opt/gitlab' |

## 3.2:部署 jenkins

Docker.yml文件如下

|  |
| --- |
| version: "3.1"  services:    jenkins:      image: jenkins/jenkins      container\_name: jenkins      ports:        - 8080:8080        - 50000:50000      volumes:        - ./data/:/var/jenkins\_home/ |

配置docker镜像源时出现报错

sudo systemctl restart docker

Failed to restart docker.service: Unit not found.

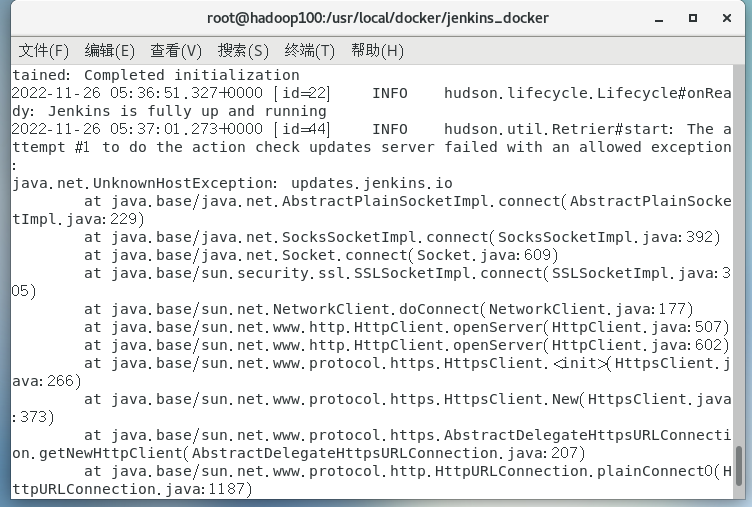
参考

<https://blog.csdn.net/zytmaster/article/details/106170908>

的安装步骤重新安装了一遍docker

重启后启动jenkins发生报错

Unknown host Exception



猜测是因为docker容器内部的dns解析出现问题而找不到主机

配置docker容器中的Dns域名解析服务器

参考

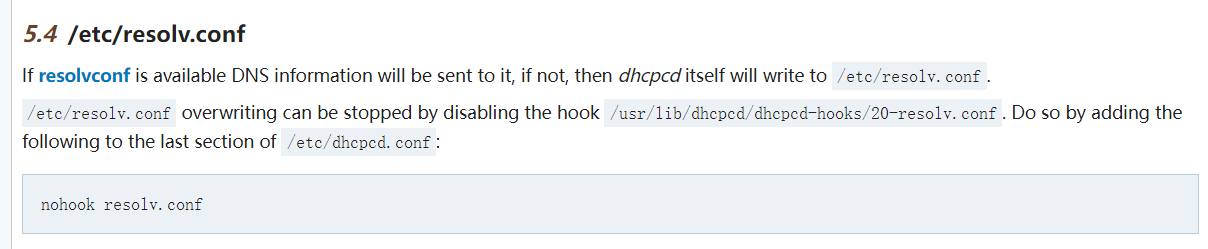
<https://blog.csdn.net/qq_37950196/article/details/125297203>

进入docker容器内部配置文件



参考

配置

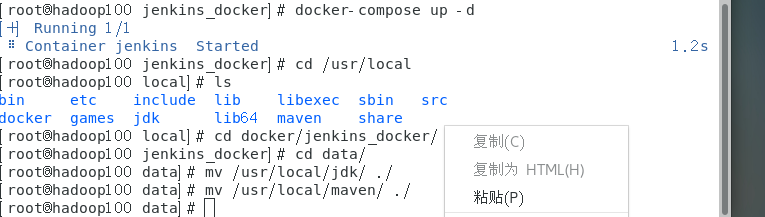


后解决问题

## 3.3配置jenkins

配置jdk 和maven，通过volume映射到容器

放到data目录下

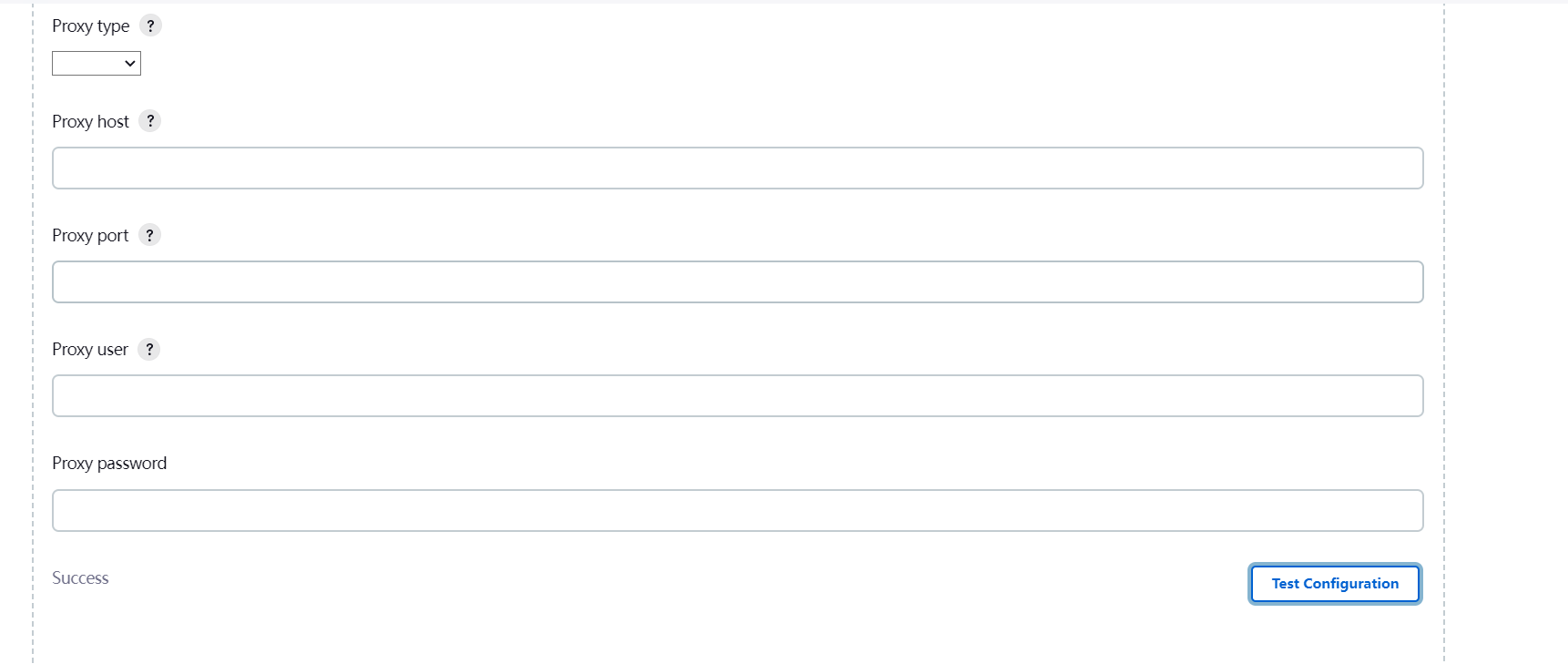


安装插件git 和publish over ssh

Publish over ssh 相关配置

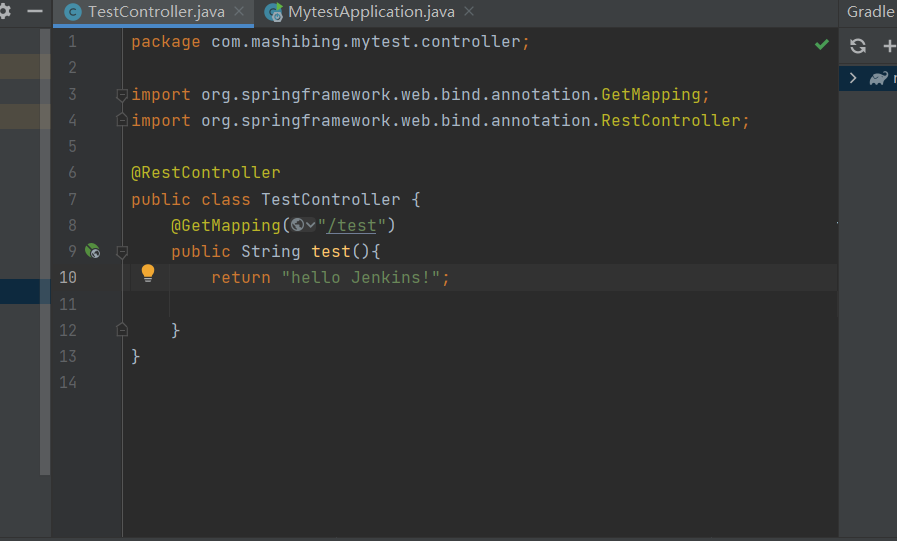


在对应主机上建立test目录，成功连接。



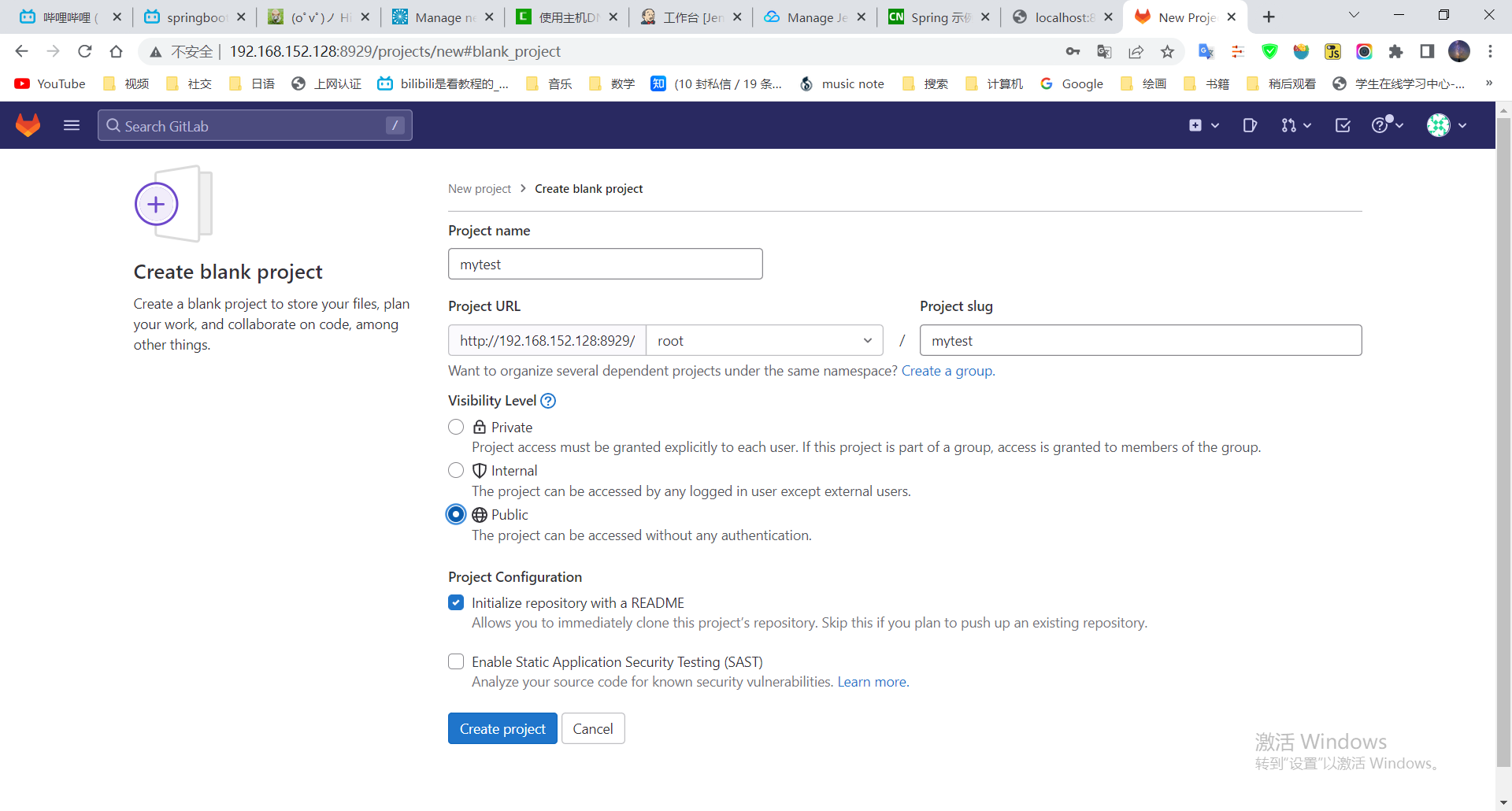
## 3.4实现基础的CI操作

准备一个spring web工程

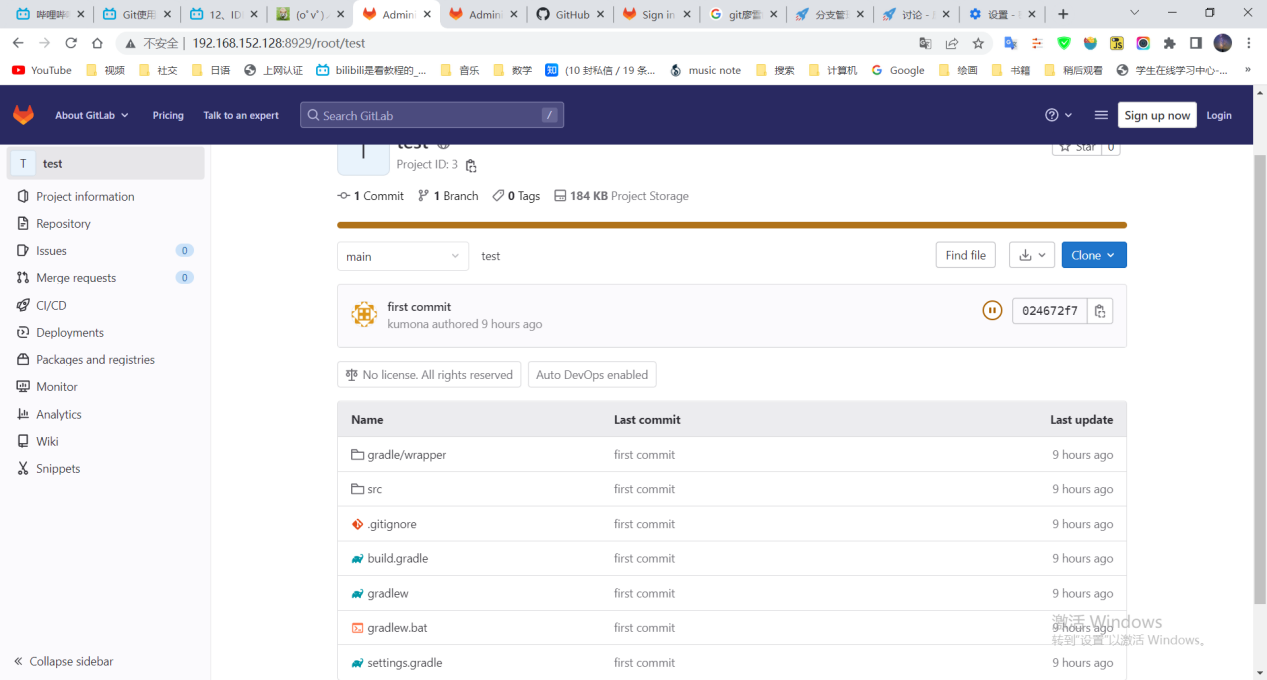




在准备的gitlab上创建一个新工程test



在IDEA上将工程代码push到项目中



中途发现无法启动jenkins，定位到报错

# **com.jcraft.jsch.JSchException: Auth fail**

错误来源是安装的publish over ssh插件

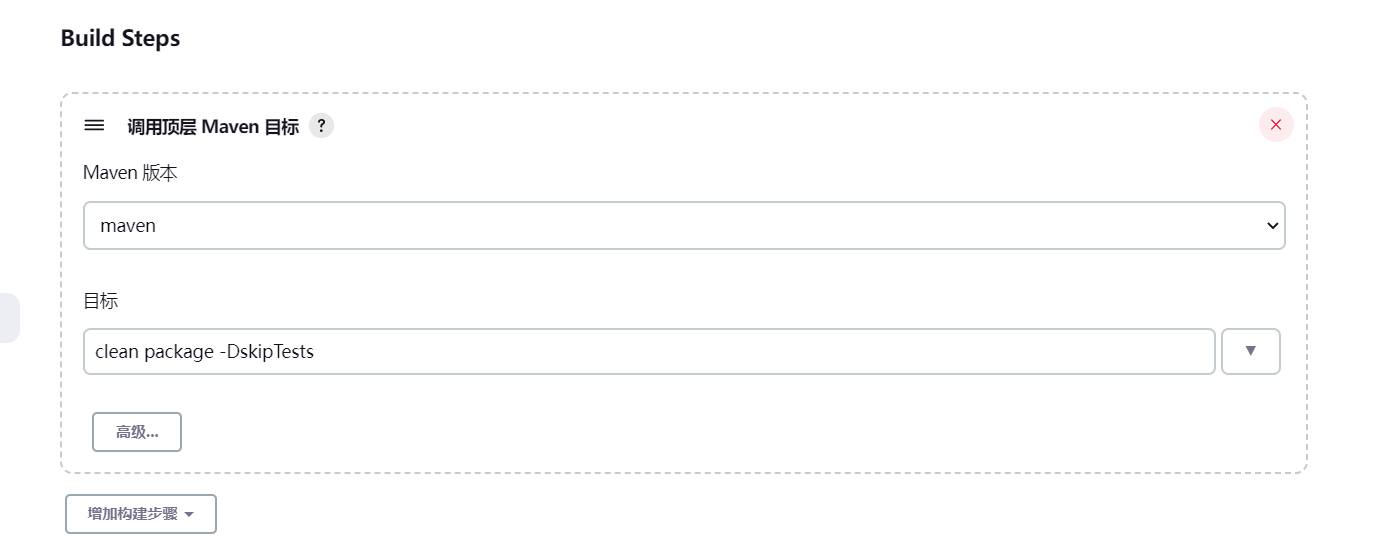
尝试禁用该插件和修改test目录权限 均无效。

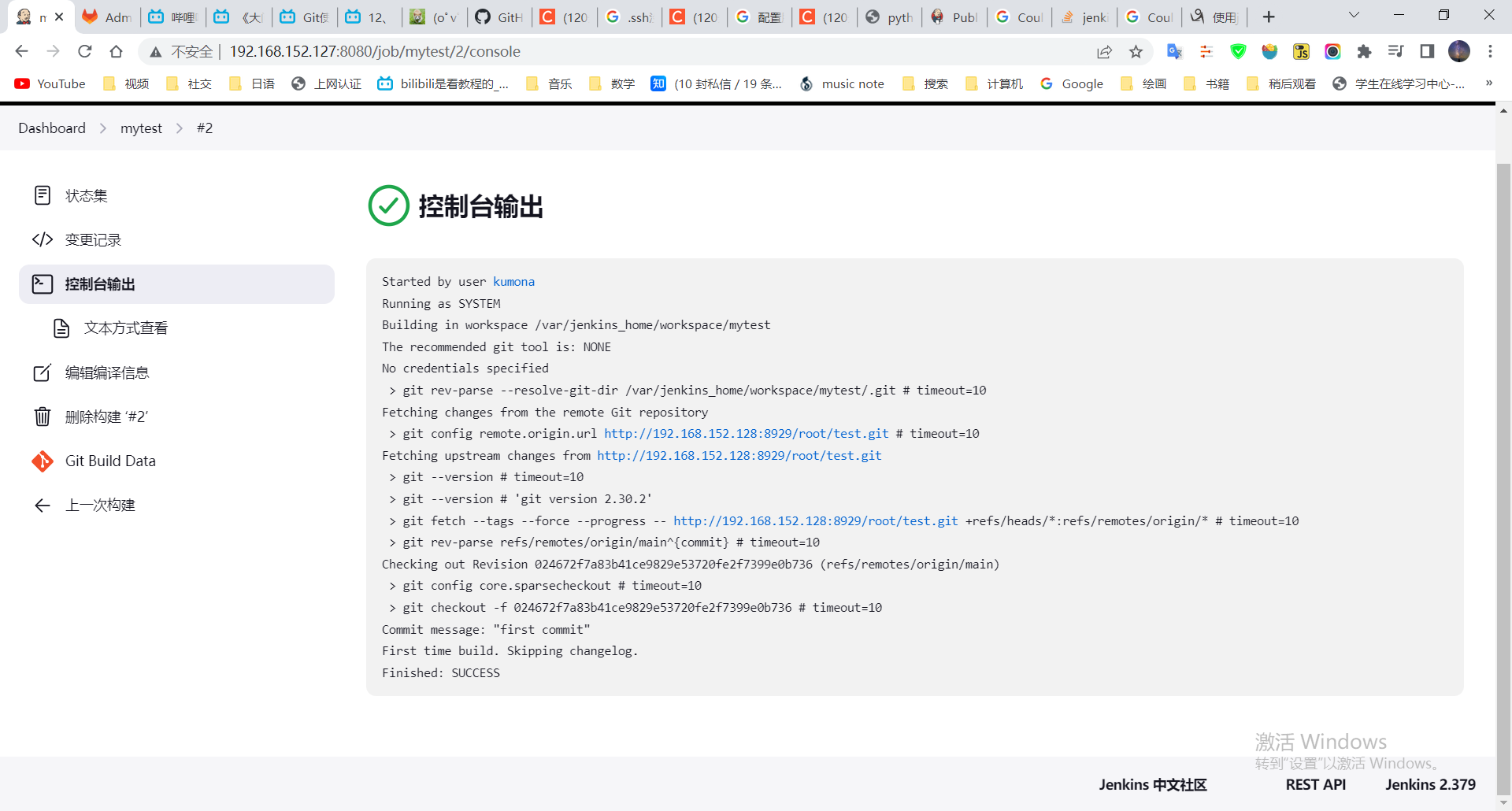
最后只能应用快照并重复了上述的步骤。

在jenkins中设置gitlab的仓库地址



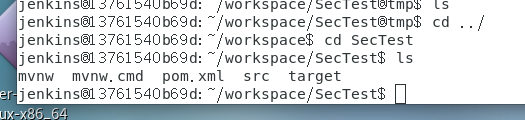
调用maven构建





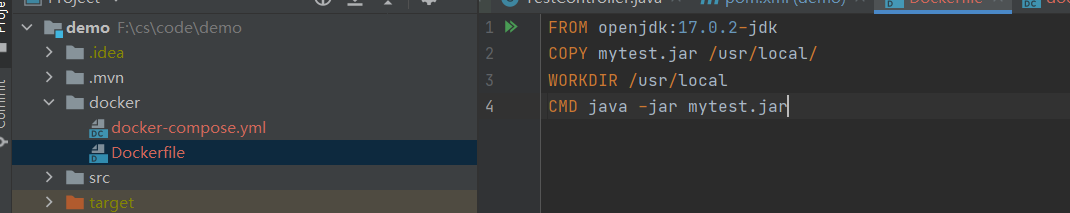
构建成功

在docker内部容器workspace目录下可以找到target目录

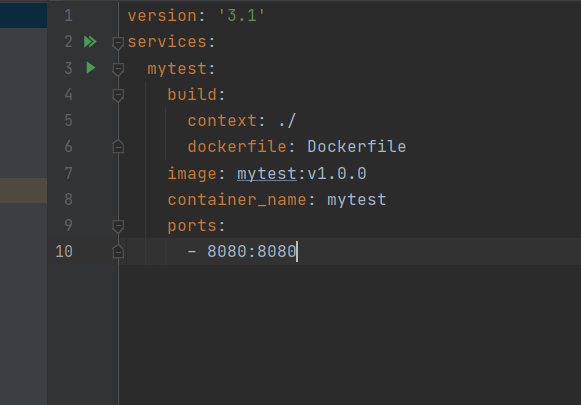


追加构建后操作publish over ssh将jar包推送到目标服务器上,此时的想法是在目标服务器上构建镜像并用docker-compose运行成容器。

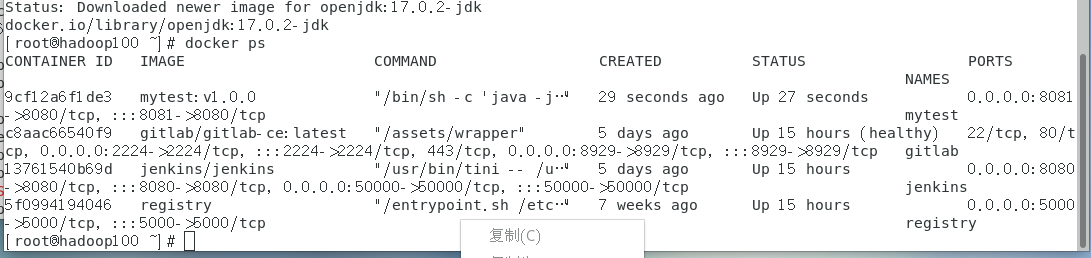
配置dockerfile用来构建镜像



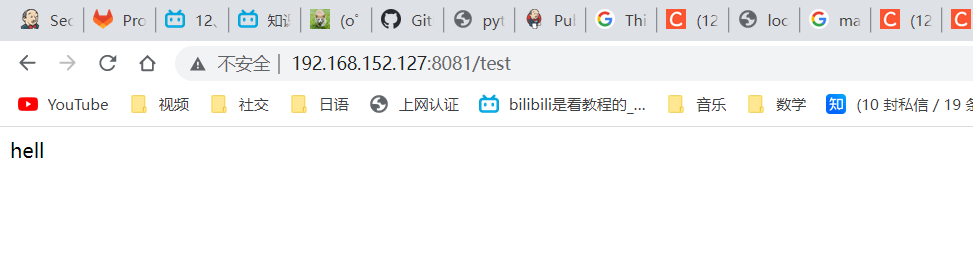
配置docker-compose.yml以运行容器



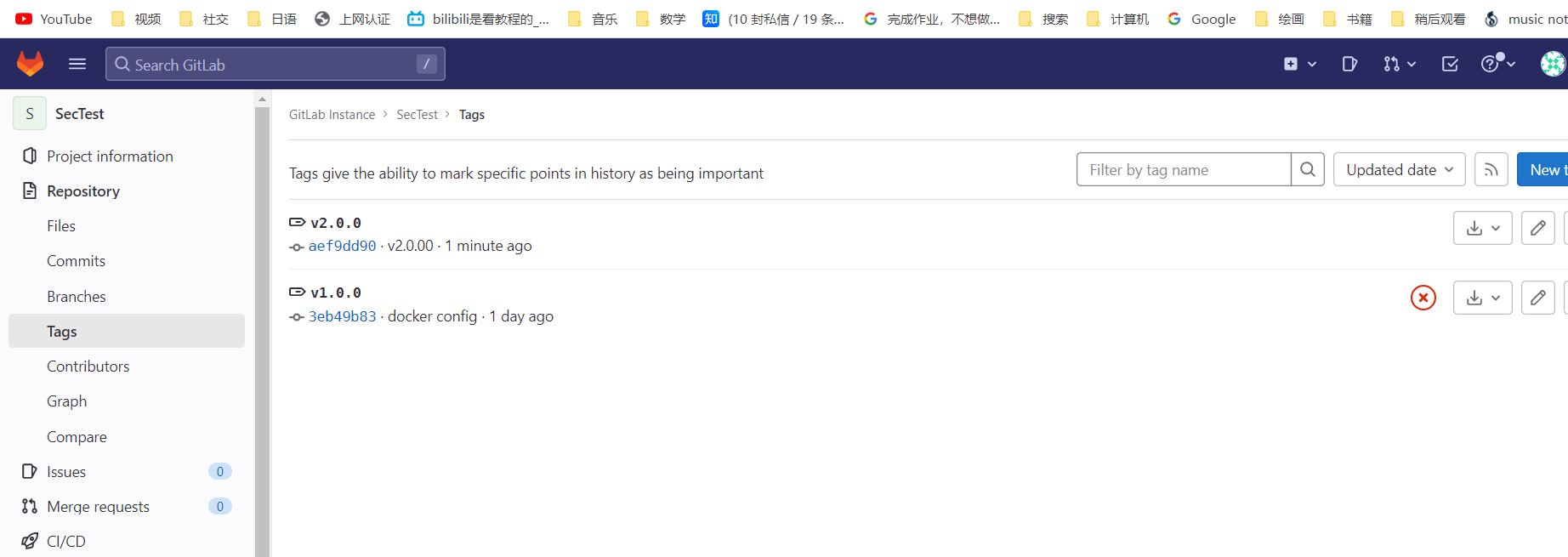
遇到两个问题，两个端口都是8080造成冲突，dockerfile跳转目录的时候必须使用绝对路径，修改后运行成功，可以看到第一个容器即启动成功的服务

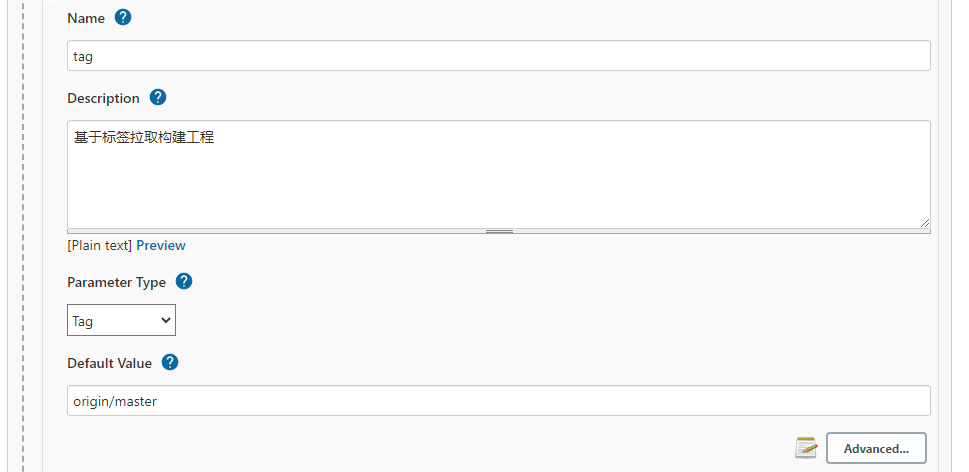


访问成功



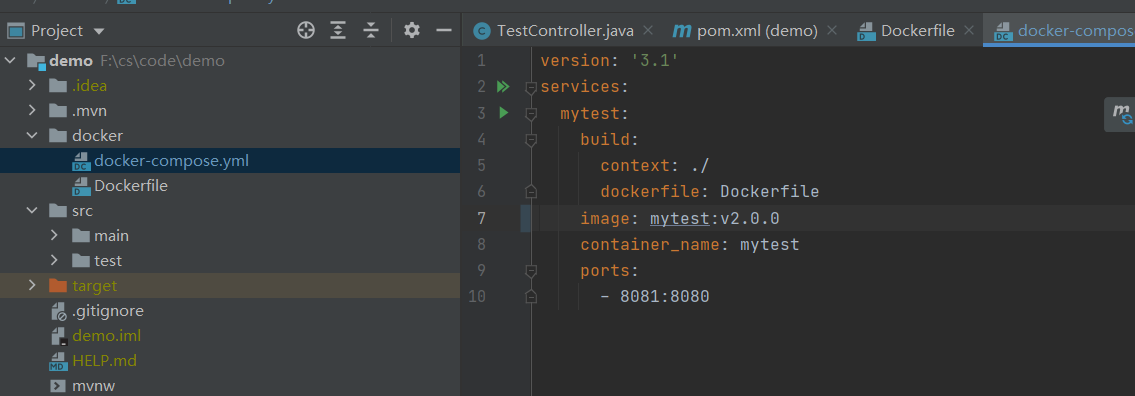
利用git parameter基于不同版本拉取镜像





分别打两个标签

更改docker-compose里的配置文件



即可通过标签拉取不同版本的镜像

## 3.5部署harbor

遇到报错无法安装harbor，发现问题出于mv命令复制文件

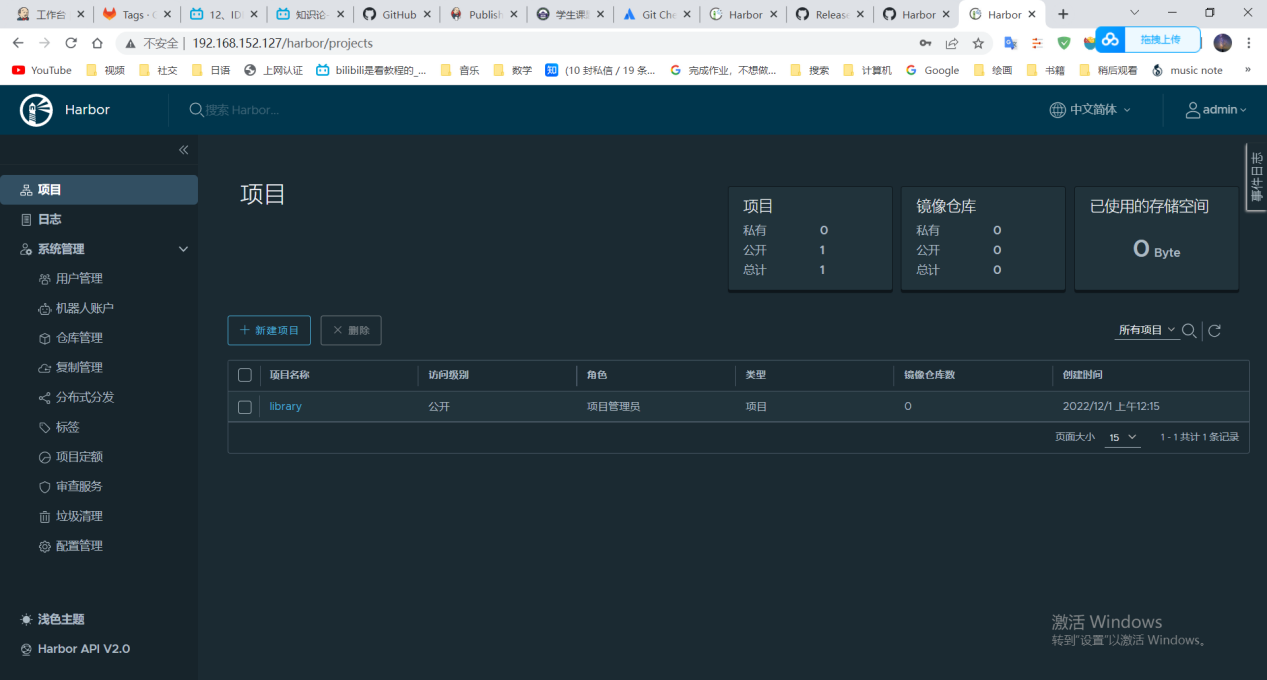
使用mv命令移动文件会损失内容？

上下两个分别是使用mv和cp命令复制harbor.gz文件的结果

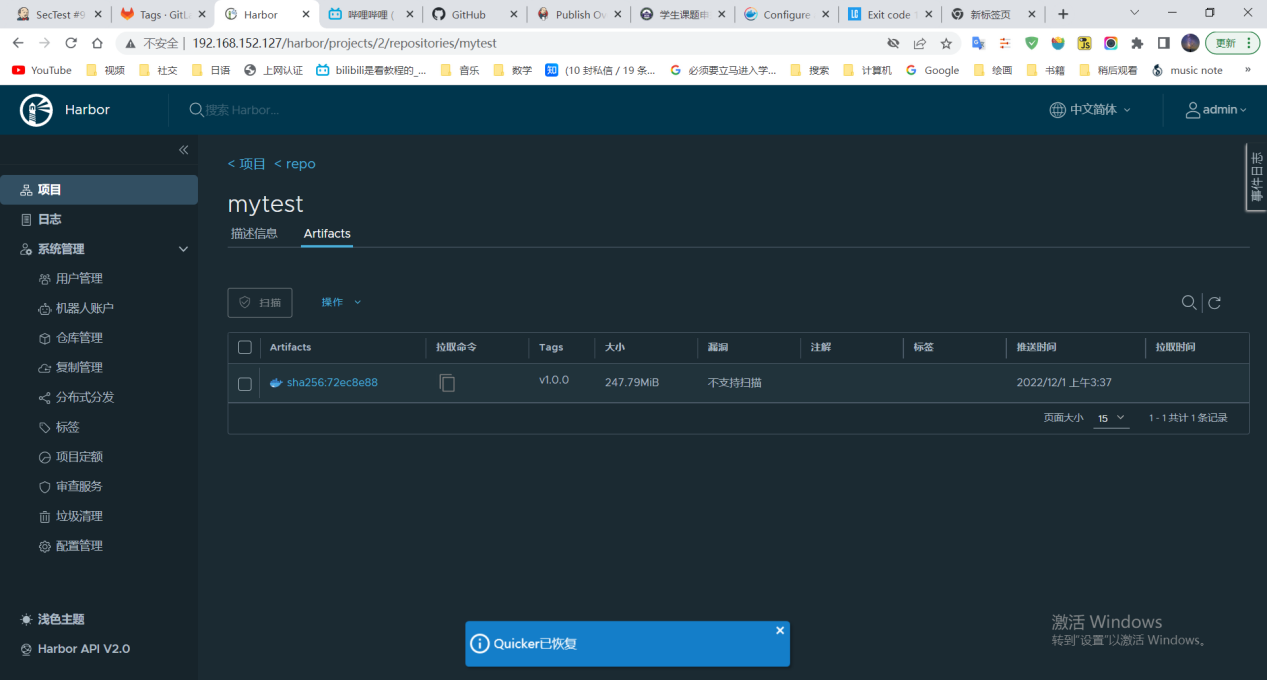


明显可以发现上面一个的gz文件相比下面的缺失了内容造成解压出现错误

成功安装harbor



配置docker.daemon.json中，harbor的地址,然后制作镜像并push到harbor仓库中



配置jenkins能够在jenkins容器内部使用docker制作镜像

配置 /var/run/docker.sock 的用户组和读写权限，以及jenkins容器的yml文件，增加volumes。

进入容器内部执行docker命令，执行成功





将制作的镜像推送到harbor仓库中

配置服务器脚本自动拉取镜像：

1. 应该拉取哪个镜像
2. 判断当前服务器是否正在运行容器，需要删除
3. 如果目标服务器已经存在当前镜像，删除
4. 目标服务器拉取镜像，运行成容器

脚本文件配置

|  |
| --- |
| tag=`docker images |grep ${project} | awk '{print $2}'`  echo $tag  if [[ "$tag" =~ "$version" ]]; then  docker rmi -f $imageName  fi  docker login -u admin -p Harbor12345 $harbor\_addr  docker pull $imageName  docker run -d -p $port:$port --name $project $imageName  echo "success" |

部署成功。



# 4 总结

## 4.1.小结

主要实现了devops的整个流程，对于项目的开发和部署有了更深一层的了解，同时增强了自己对linux排查错误的能力。

## **4.**2 项目存在问题与不足

工程并不是很完善，只能用来测试部署的结果。Jenkins使用的是自由风格的项目管理，这并不是一个好的方案。

## **4.**3 项目展望

完善工程项目，可以使用更多的工具完善部署流程，如k8s和jenkins的pipeline风格项目。

# 5学习体会

## 5.1项目相关技术学习体会

了解了一个项目的devops核心流程。Push代码到Gitlab仓库中，通过jenkins拉取，使用maven构建，然后制作镜像上传到harbor仓库中，最后目标服务器拉取镜像运行成容器。整个流程以jenkins为核心，一键部署，可以节省开发者大量的时间。

## 5.2云计算技术的行业应用体会

未来若能实现元宇宙，云计算将是其中非常重要的部分，所以云计算必是未来发展的一个趋势。

**参考文献**

1. <https://www.bilibili.com/video/BV13Y411E7nd/?p=1&vd_source=7261e37adc33f0ab8f2703fb95098e5f>
2. <https://blog.csdn.net/qq_37950196/article/details/125297203>
3. <https://blog.csdn.net/lee_yanyi/article/details/125912314>
4. <https://blog.csdn.net/zytmaster/article/details/106170908>
5. <https://zhuanlan.zhihu.com/p/81213673>