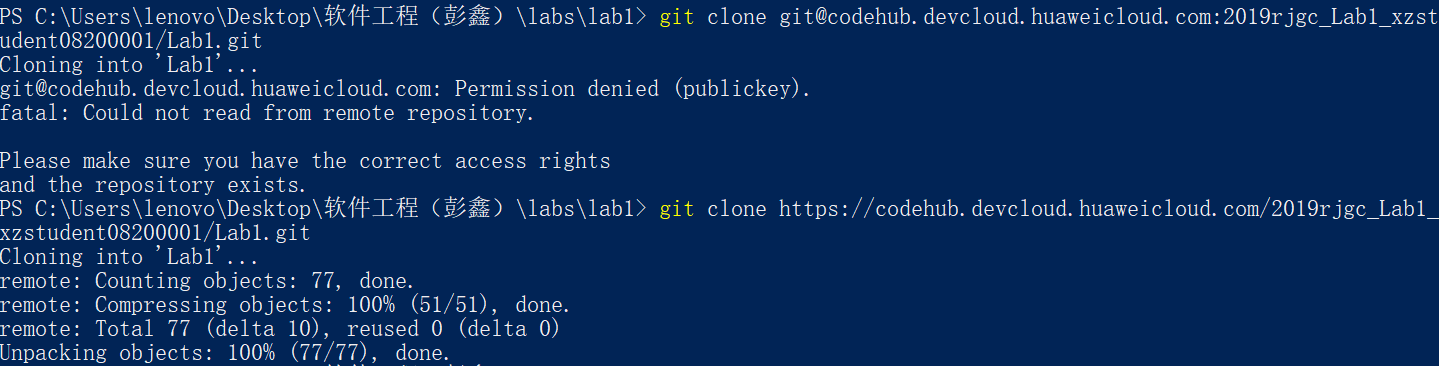
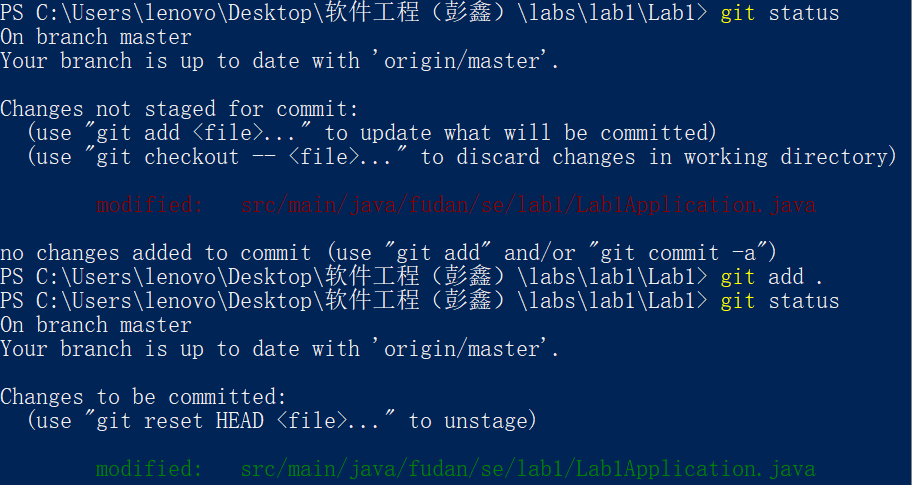
Lab1实验报告

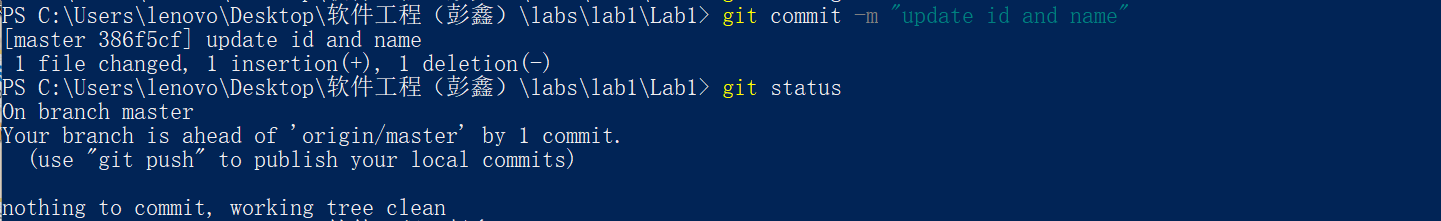
17302010079 宋怡景

1. 实验过程：
2. 按照TA给出的文档将lab1仓库克隆到本地，第一次我尝试使用SSL克隆，提示仓库不存在，于是改用HTTPS克隆，成功！（git clone后面加仓库的URL，支持多种协议，包括ssh和https等，此处之所以不可以使用SSH克隆，原因为没有添加密钥）

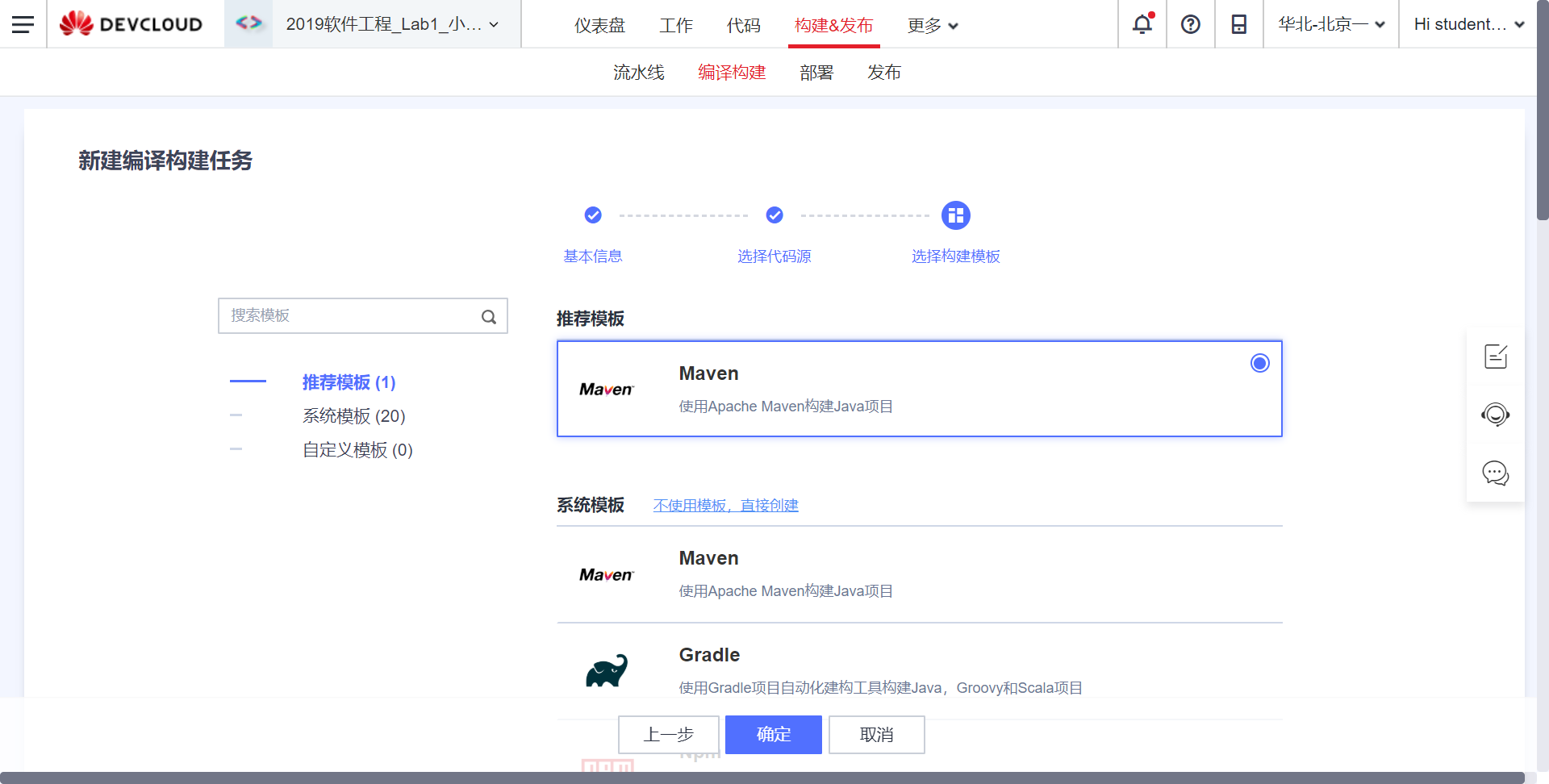


1. 在本地用编辑器打开lab1，改变其中的name和id，之后，使用git add .和git commit -m “”尝试提交更改（git add命令将修改保存到缓存区，git commit将缓存区的修改推送到本地仓库中，-m后面的参数为此次修改的描述）





1. 按照文档构建，成功！（此处选择Maven模版，Maven是一个项目管理工具，可以对Java项目进行构建，依赖管理）



1. 按照文档中部署，部署通过，但是url拒绝访问

在主机上安装JDK，使Java项目可以在主机上运行：



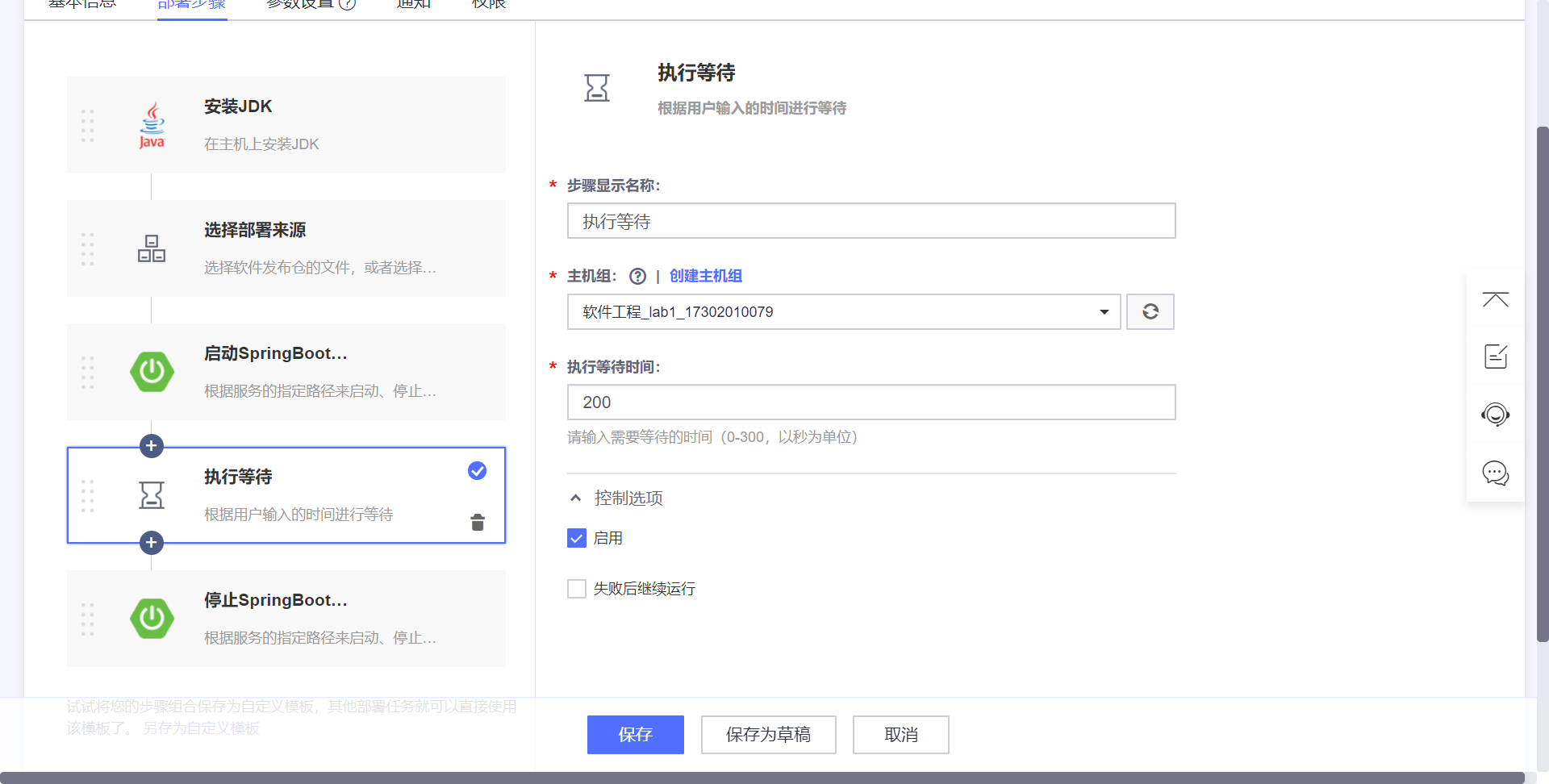
选择部署来源，选择构建任务的构建产物：



启动SpringBoot服务：



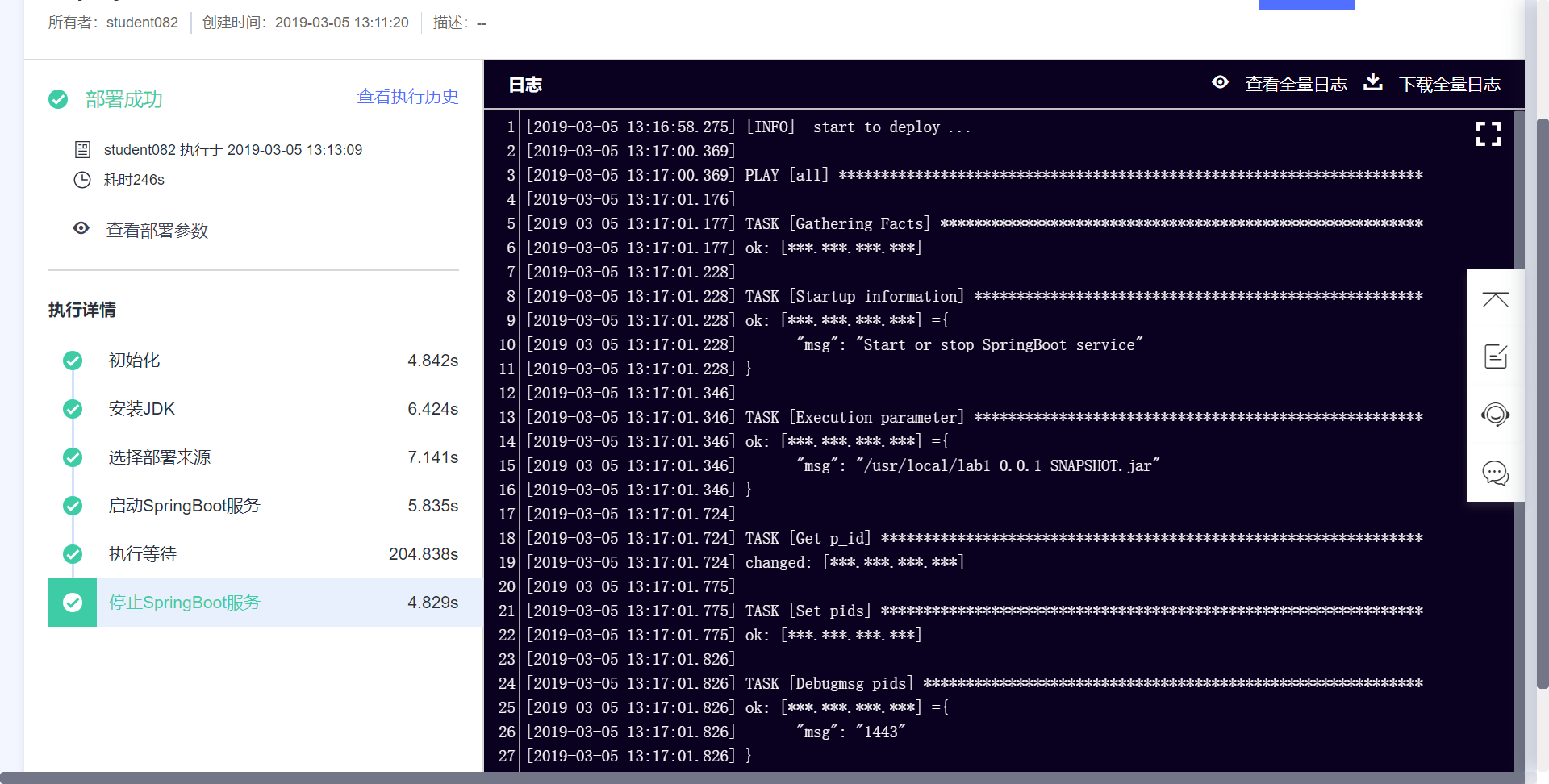
执行等待，其中的200s表示在服务开启至服务关闭有200s：



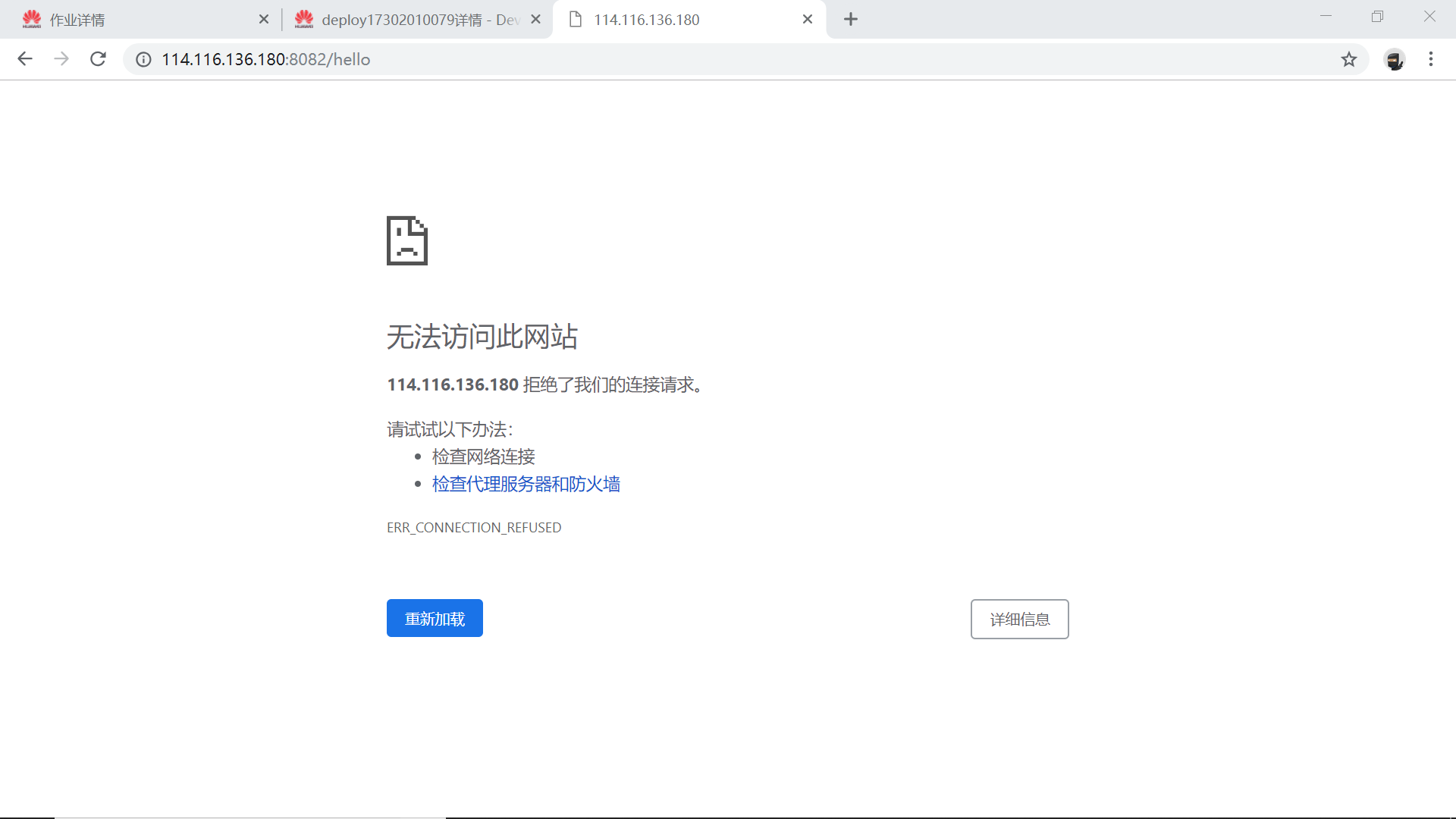
停止SpringBoot服务：



执行，部署成功：

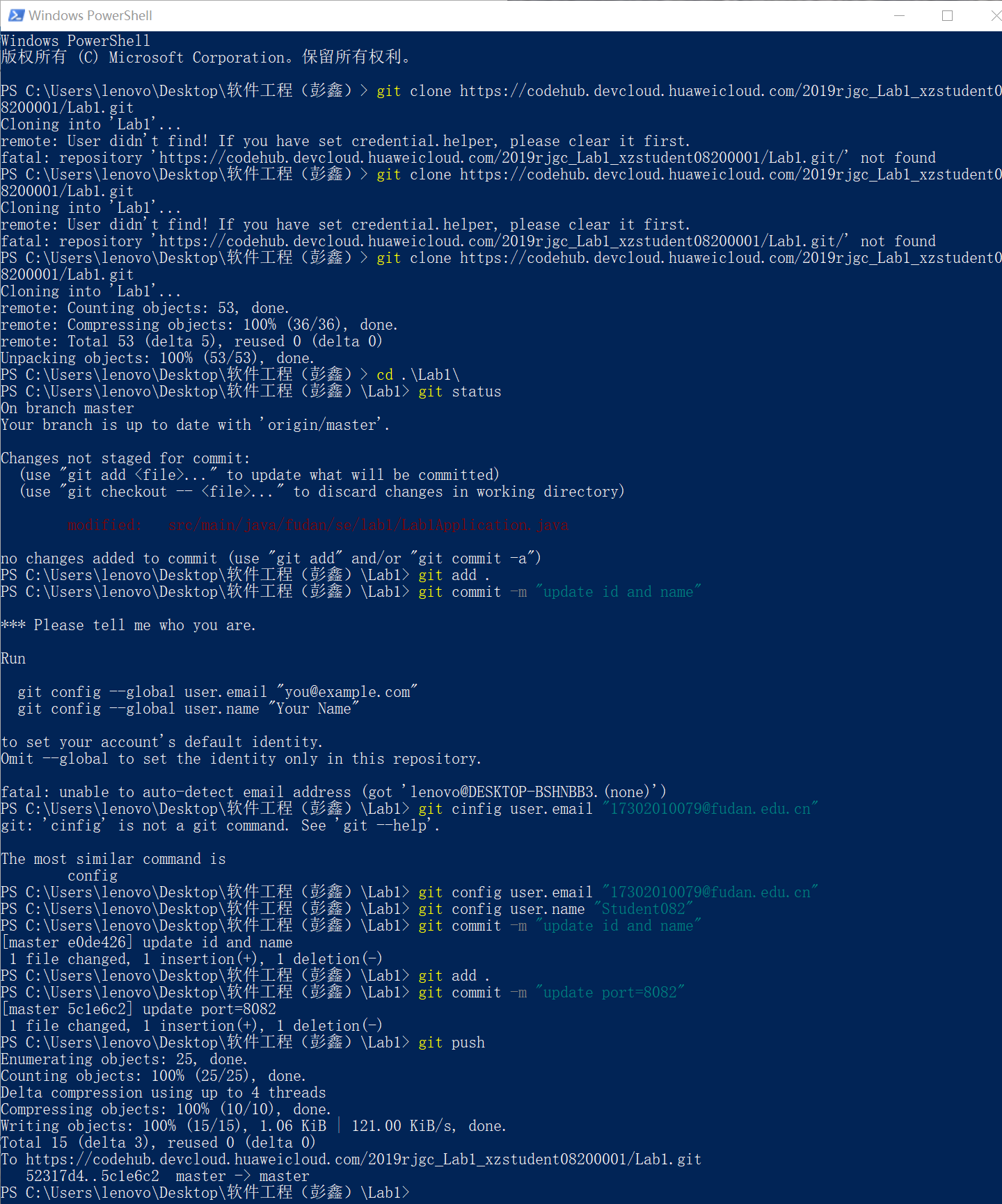


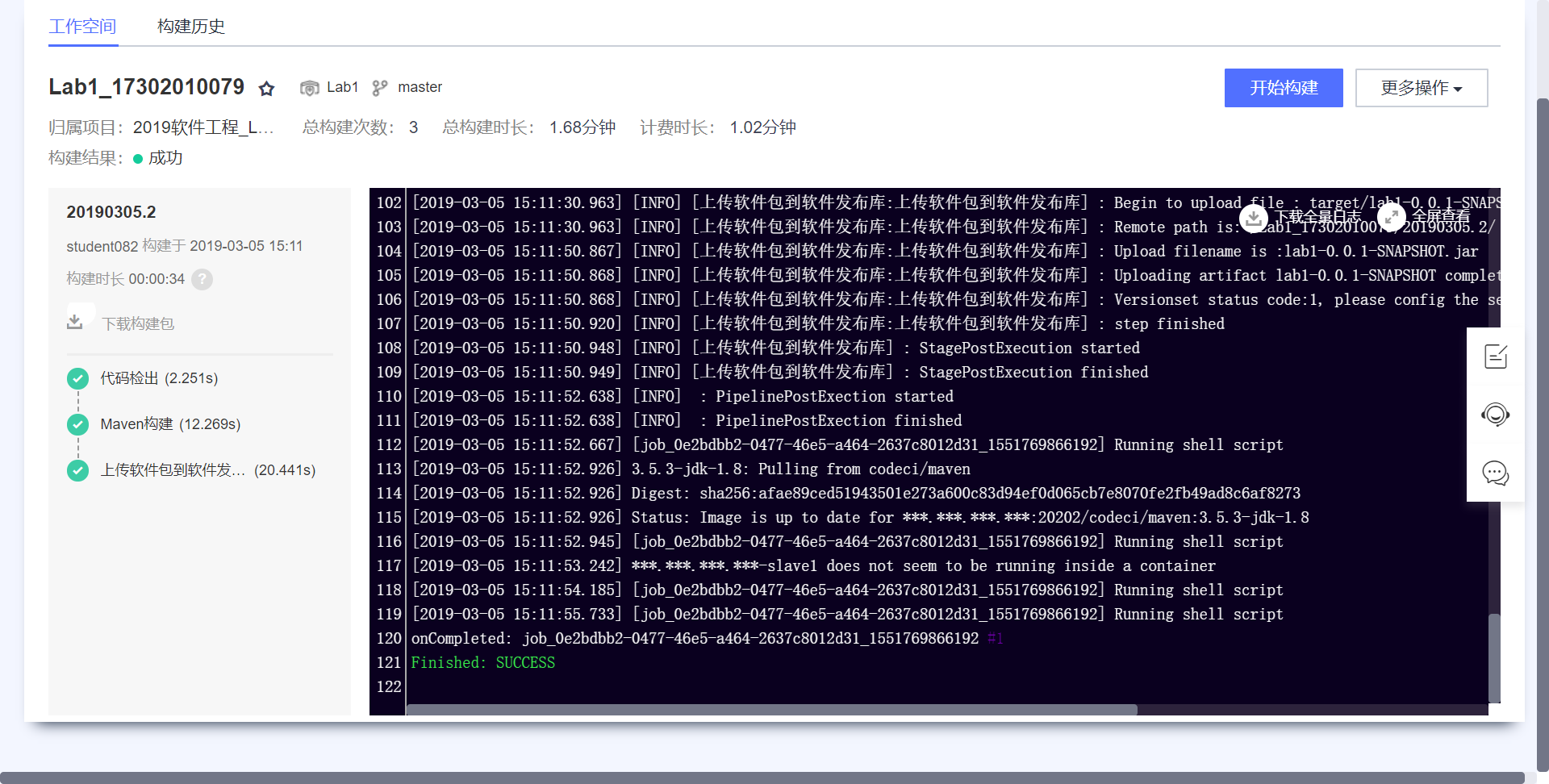
但是url无法访问：

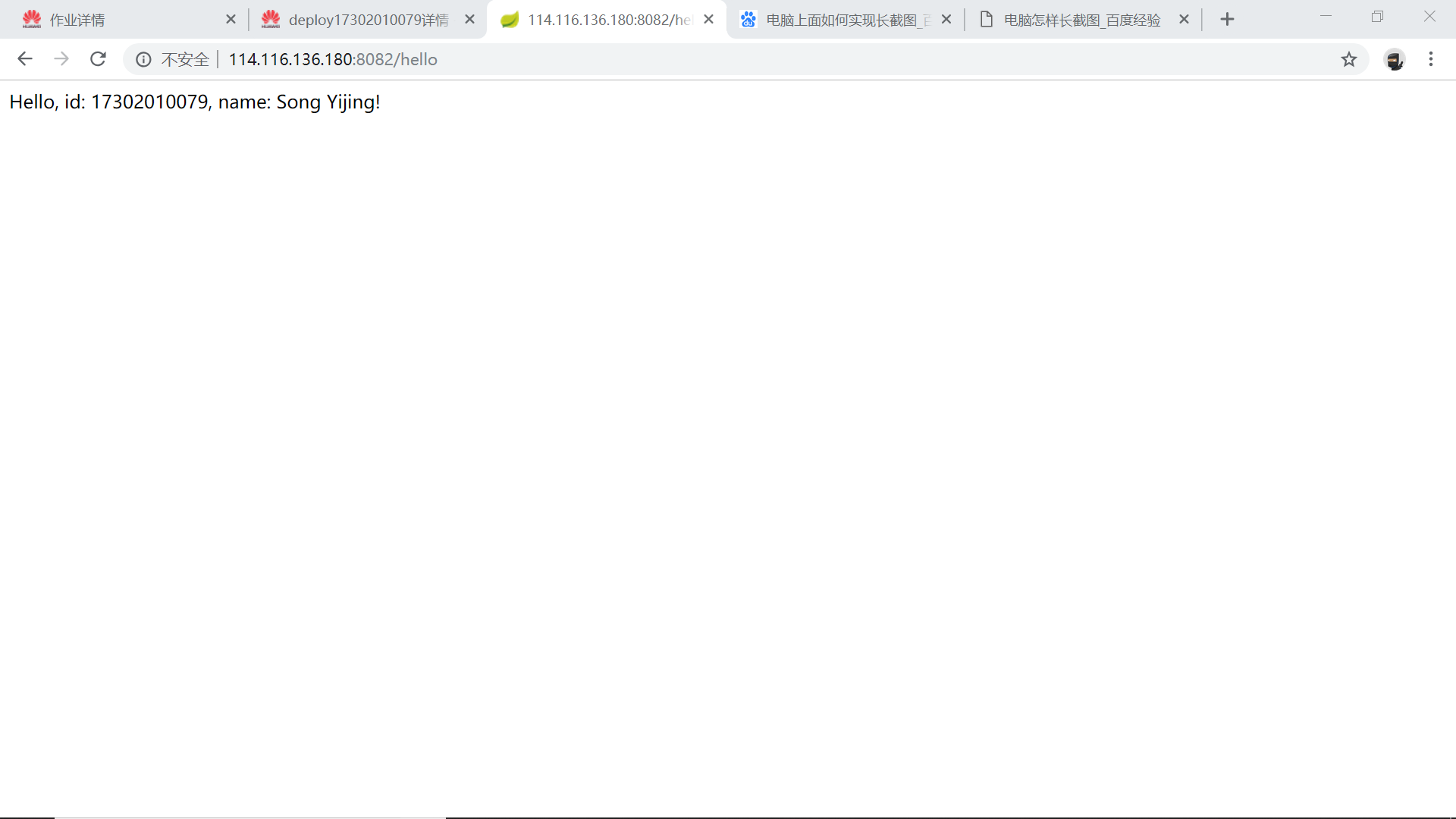


1. 反思之后，发现没有修改端口，修改端口之后，重复步骤2提交，尝试直接部署，url仍旧拒绝访问
2. 步骤回退，重新编译构建，然后再次部署，仍然拒绝访问
3. 于是想到是代码的问题，查看华为云中的代码，发现我在本地所做的修改在其中并没有体现，重新查看文档之后，每次在本地修改代码都没有push上去
4. 于是从克隆开始，重新做，成功！

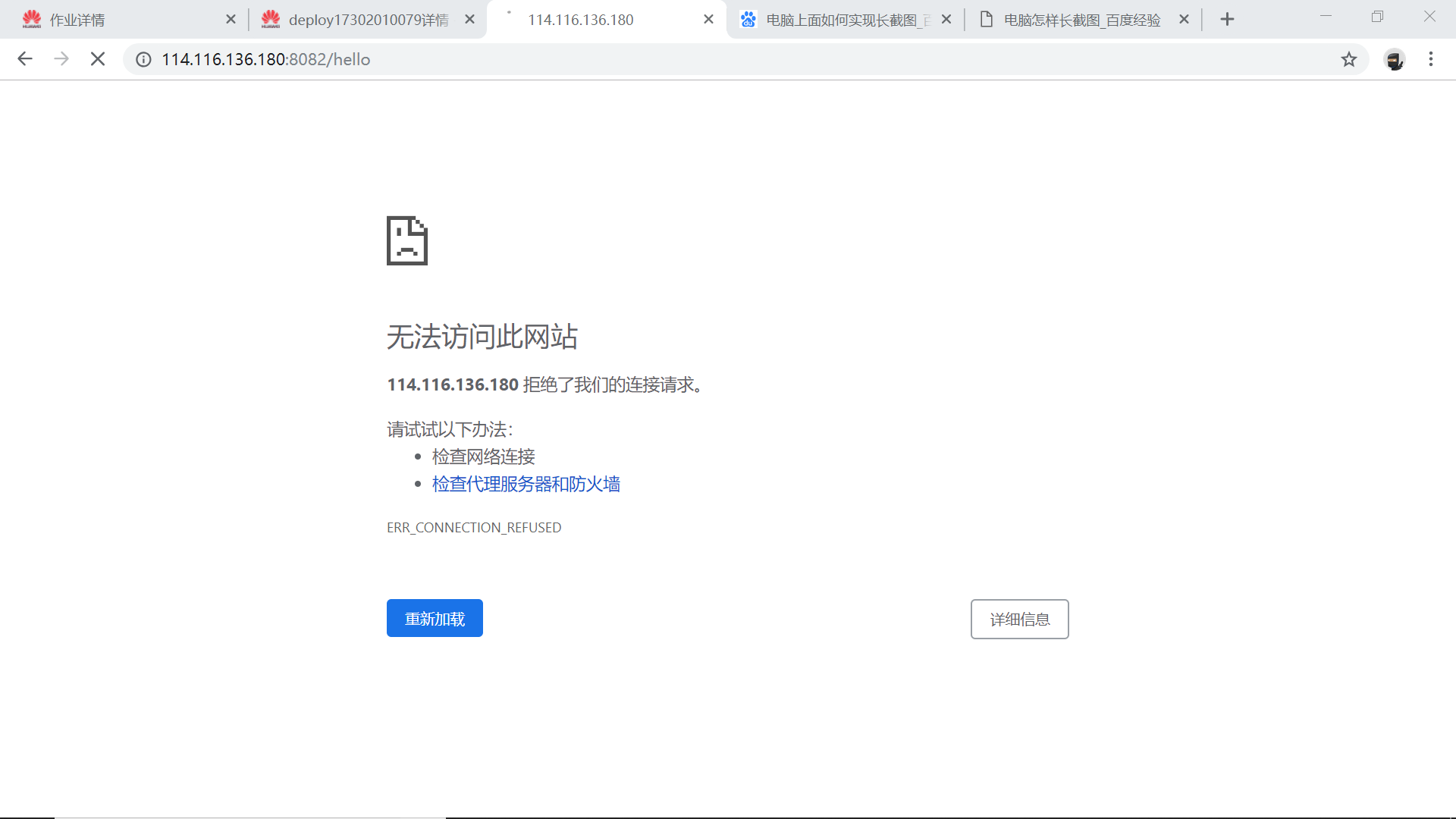
最后一次过程的截图：



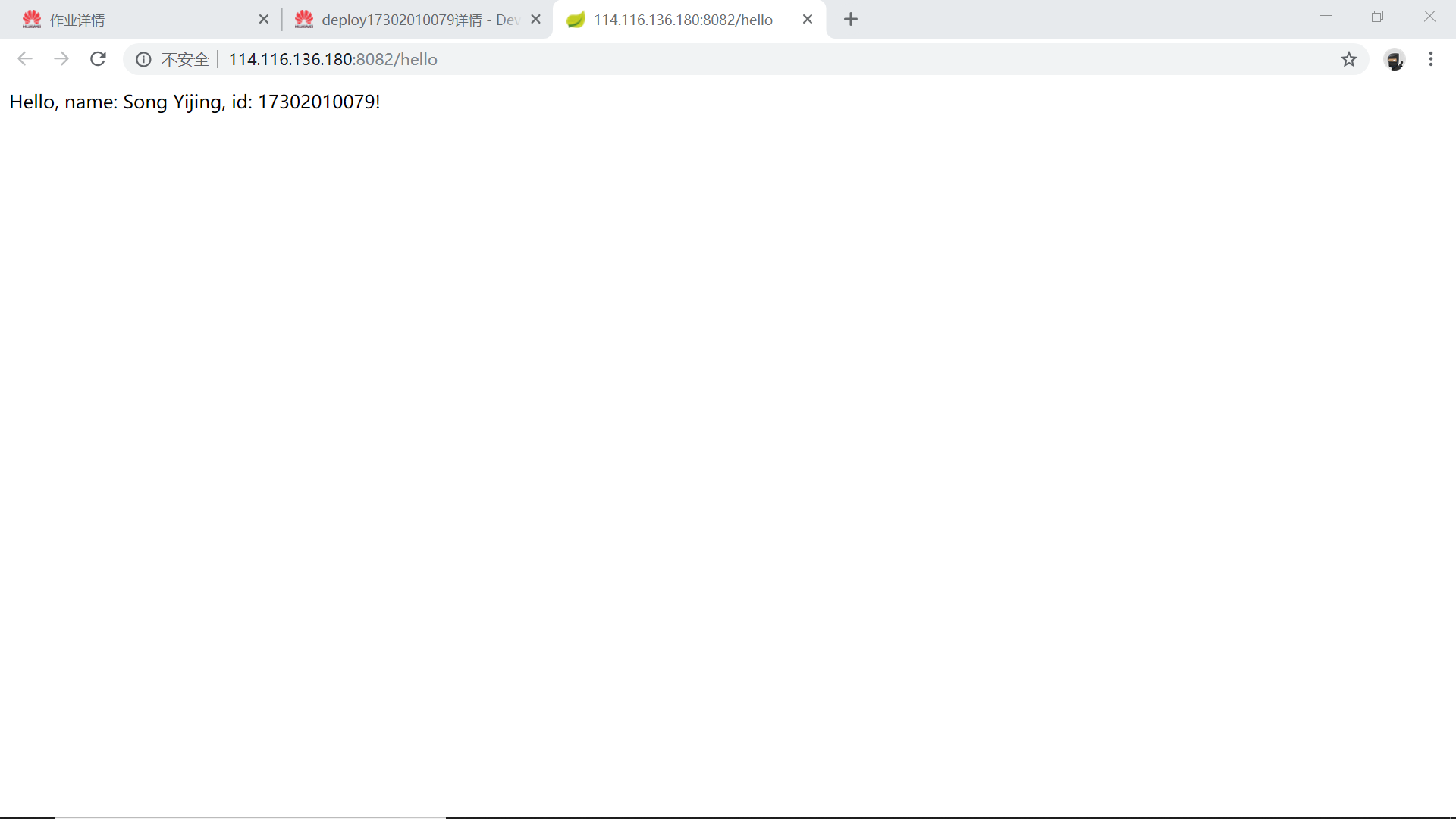


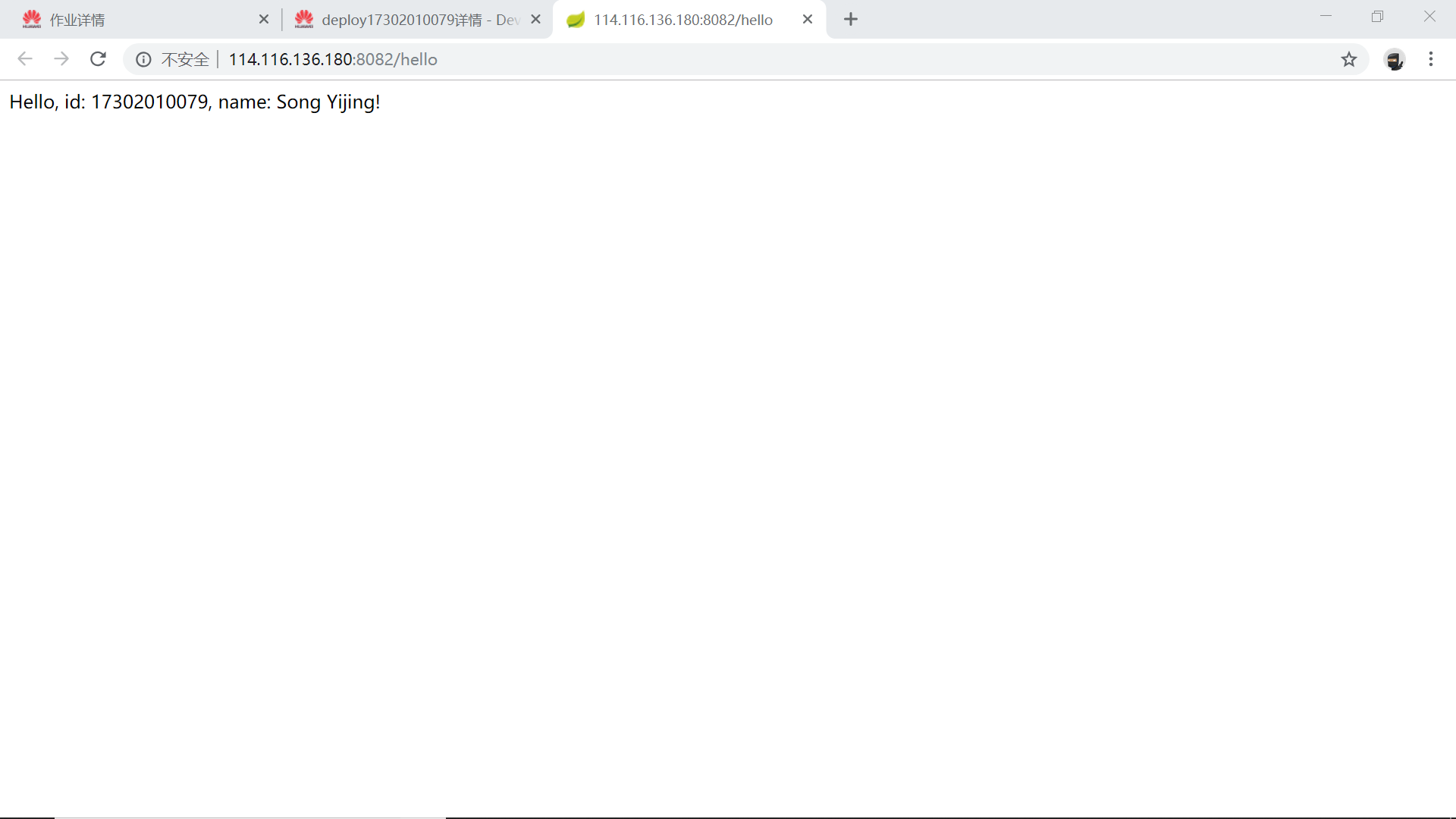


1. 200s之后，再次访问url，拒绝访问：



1. 实验中的体会与对实验的理解
2. 通过这次实验，我已经掌握了如何使用git将仓库复制到本地，以及如何在本地修改后提交，尤其对push步骤印象尤为深刻
3. 在上面实验步骤5-6步过程中，想到，如果将项目修改后，不经过编译构建直接部署，那么结果会改变吗？于是，将项目代码中的id和name的顺序调换，提交之后，直接部署，发现结果并没有发生变化，如下图所示：（上为重新build之后部署的结果，下为没有build直接部署的结果）





1. 为了更好的体会git在VCS方面优势，我查看了代码托管中的修改历史，如下：



虽然对git对版本管理的作用体会并不是很深刻，但是在历史中，但是对git对每次修改的记录这一功能，十分好评

1. 通过此次实验，我并没有对Maven和SpringBoot产生深刻的理解