**文档说明**

项目地址：

<https://github.com/121-hello/github_try>

[ExecuteTotal.py](https://github.com/121-hello/github_try/blob/master/ExecuteTotal.py) 用于执行sql文件

**unit 文件夹内放着读取sql文件以及初始化数据库的一些信息**

**log文件夹放执行后的log**

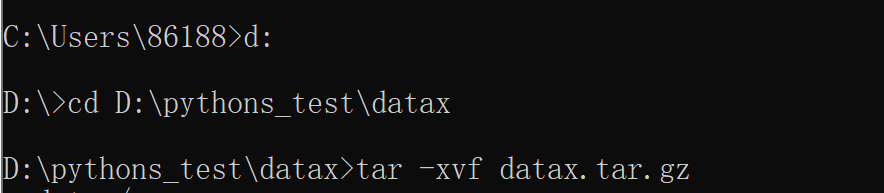
**sql文件夹放待执行的sql脚本**

**job文件夹放json文件(跨数据库取数)**

**当需要跨数据库取数时，如从oracle到mysql或从mysql的一个数据库到另一个数据库时，可以使用datax**

**下面简单介绍datax使用：**

1. **datax 的使用，离线数据同步工具/平台**
2. **下载压缩包：**<http://datax-opensource.oss-cn-hangzhou.aliyuncs.com/datax.tar.gz>
3. 移动至 D:\pythons\_test\datax



tar -xvf datax.tar.gz 解压

**注意：**

python3 需要将解压后bin 目录下的三个py文件修改成了符合 python3 语法要求的文件

文件在下地址下载

<https://github.com/TwoThreeWang/DataX_Python3>

环境：

* Linux/Windows
* JDK(1.8以上，推荐1.8)
* Python(推荐Python2.6.X，我用的python3,须替换文件)
* Apache Maven 3.x (Compile DataX)

之后，进入bin目录

cd D:\pythons\_test\datax\datax\bin

执行：

python datax.py -r streamreader -w streamwriter



演示从mysql的test数据库到mysql的test1数据库

D:\pythons\_test\jobs 新建文件jobs1

cd D:\pythons\_test\datax\datax\bin

python datax.py

python datax.py D:\pythons\_test\jobs\jobs1.json 执行

我这里由于mysql版本8.0 报错 连接失败，需要搜对应的jar驱动，放在下面文件夹内

如 mysql-connector-java-8.0.19 百度搜即可

替换后执行成功，dates表成功同步

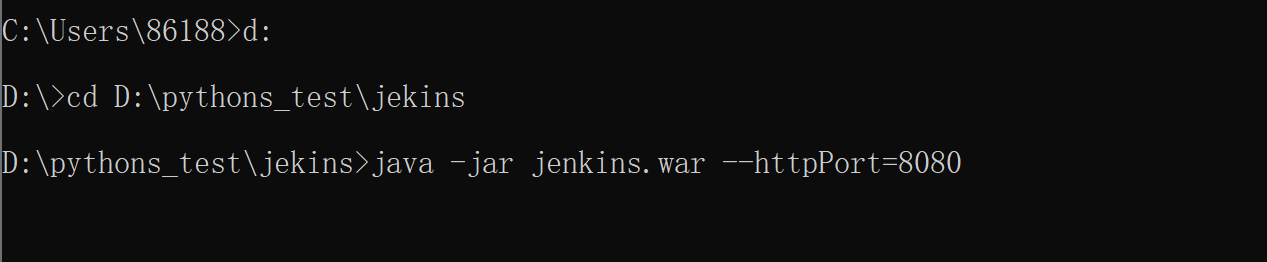
后面就可以根据python语句调用以及jekins自动部署执行

下面我注册本地JENKINS测试了一下

JEKINS安装

下载安装包[jenkins.war](http://mirrors.jenkins.io/war-stable/latest/jenkins.war)；

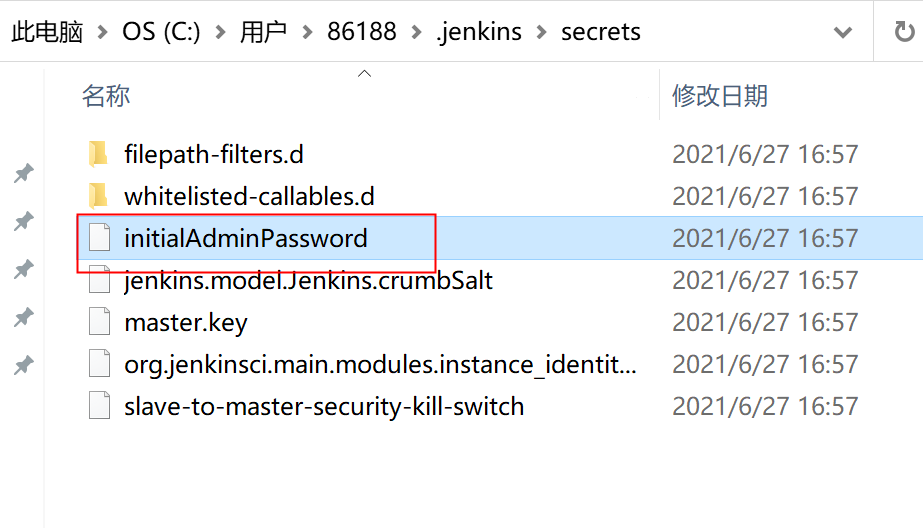
在安装包根路径下，运行命令 java -jar jenkins.war --httpPort=8080，（linux环境、Windows环境都一样）；



打开浏览器进入链接 [http://localhost:8080](http://localhost:8080/).



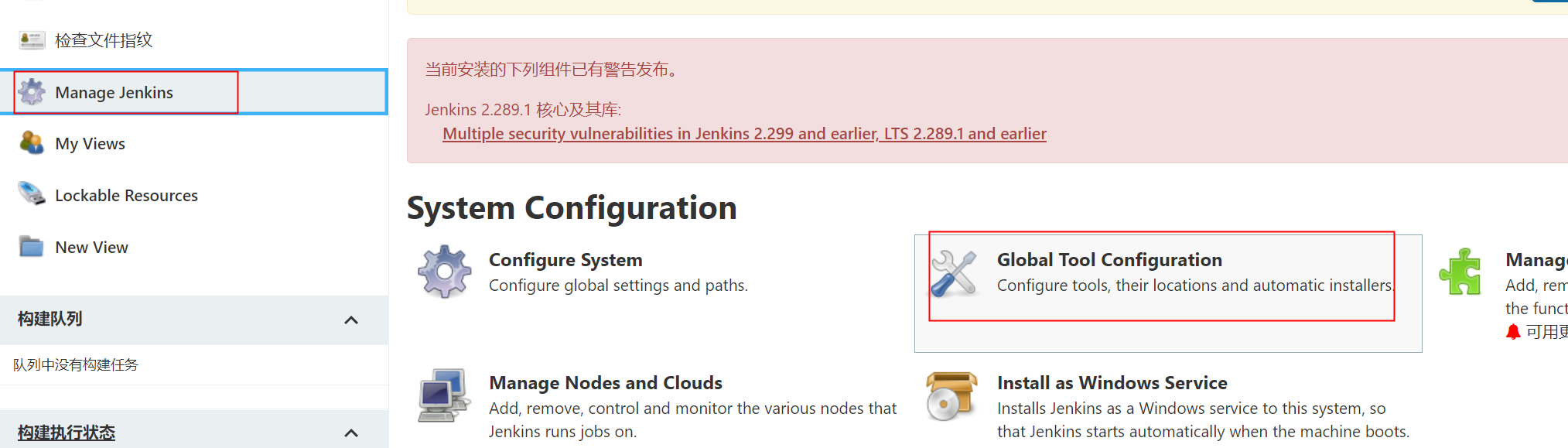
找文件 复制密码



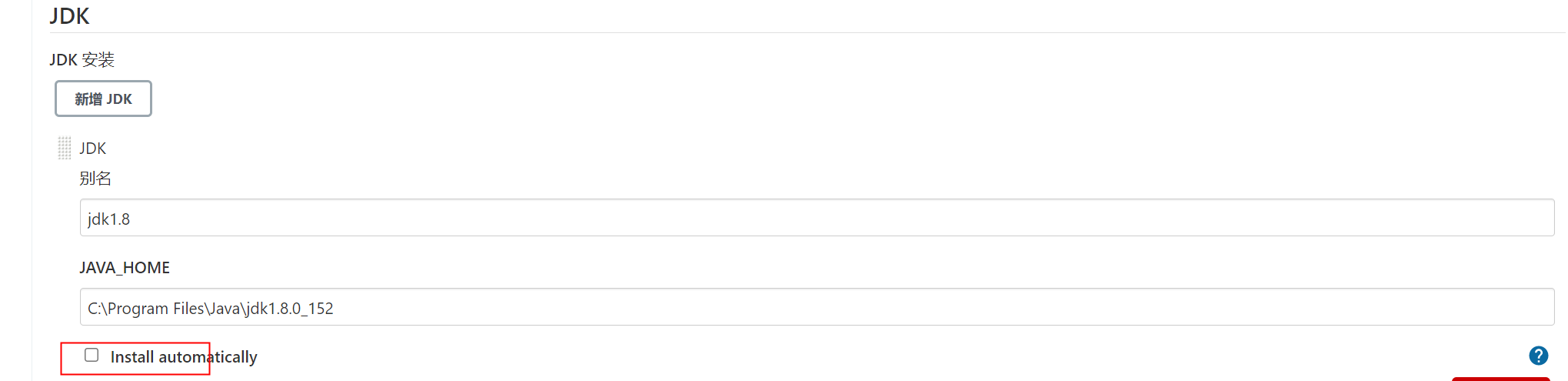
登录成功，默认配置后进入即可

下面需要配置运行环境

来设置jenkins的jdk,git和Maven



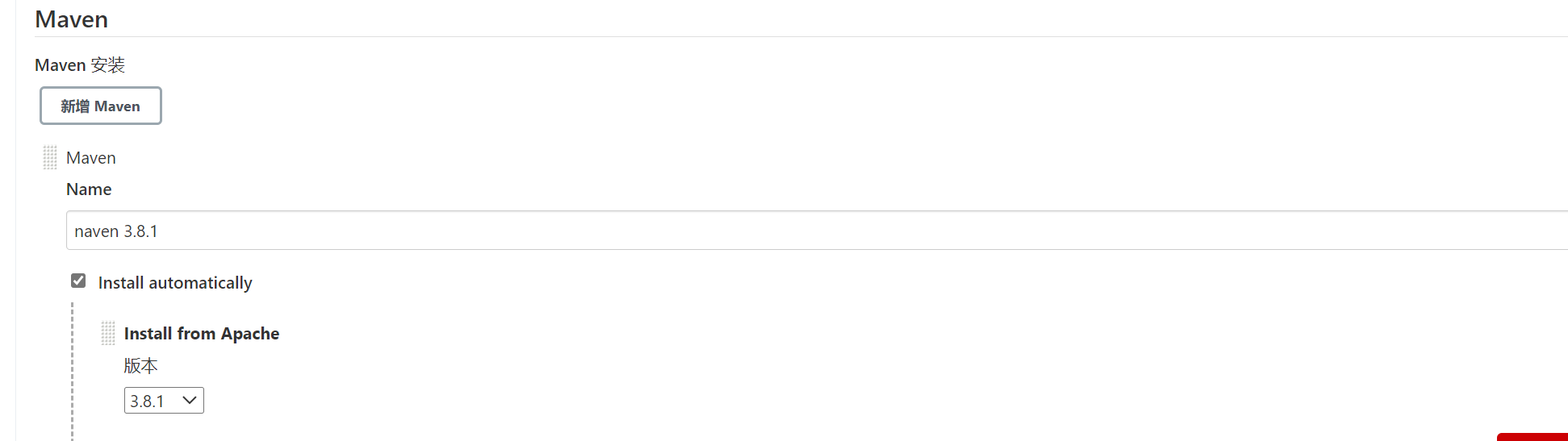
这里jdk可以自动安装，也可以自己设置安装路径



**Git一样**

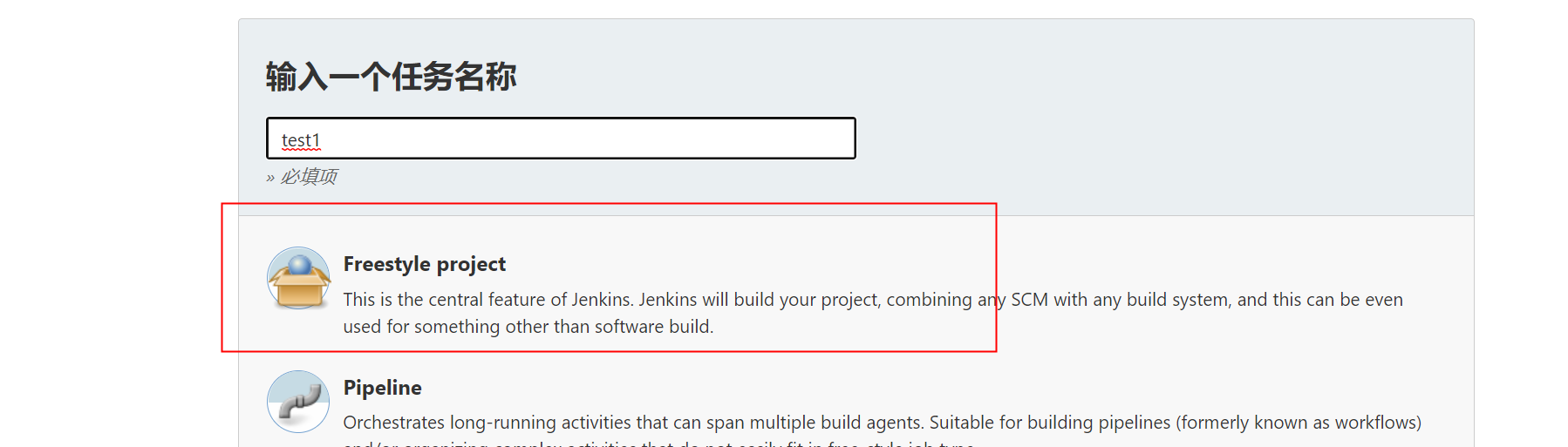


前两项为我自己目录，MAVEN选的是自动安装(git和jdk也可)



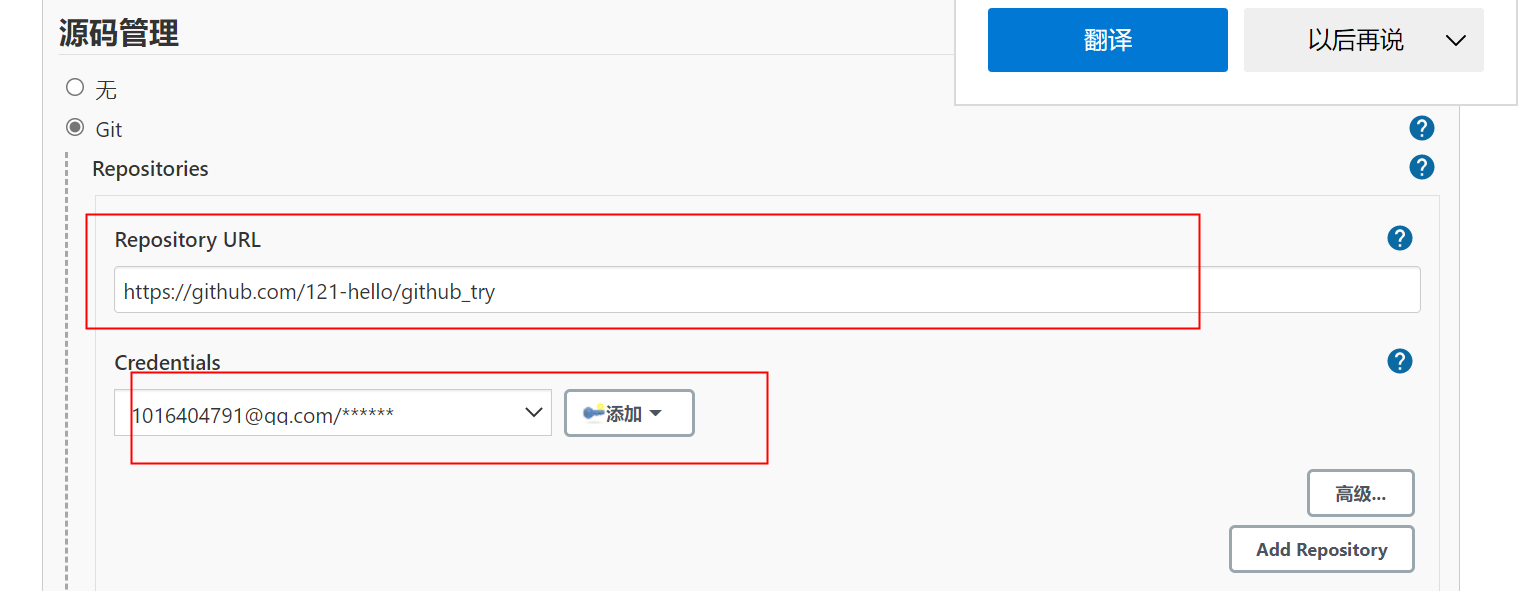
**即配置完成**

**新建一个项目试试：**





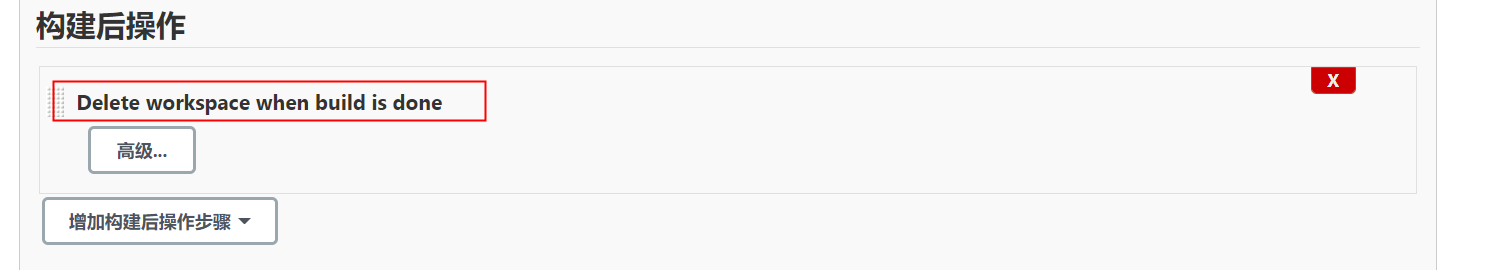
源码管理中配置项目在github的url以及登录账号密码



这里填写执行python脚本的语句



这里job所在的工作区目录里面的东西清空掉再进行构建

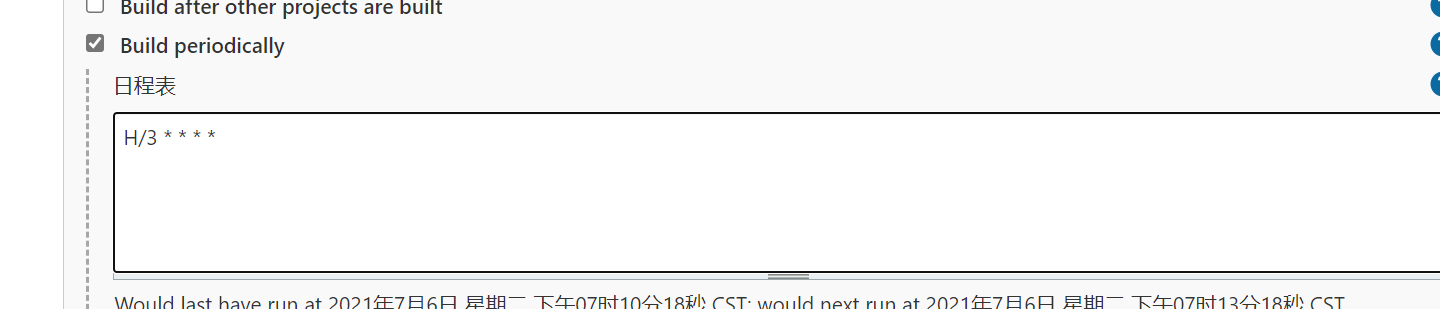


设置好后运行试试



脚本成功执行，数据库里面的dates表也进行了更新

如何定时，定时试试(每三分钟执行一次)



**确认之后**



**可以看到定时任务已经成功执行**

服务器参考：

[Jenkins自动构建部署项目到远程服务器上\_陌筱明的博客-CSDN博客\_jenkins远程部署](https://blog.csdn.net/ming19951224/article/details/80958761?utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-2.baidujs&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-2.baidujs)

**复杂的血缘关系也是基于此，在数据库里描述血缘关系，用python在jekins中自动执行**