cd /usr

mkdir java

mkdir maven

mkdir tomcat

在网上下载linux版本的jdk，maven，tomcat压缩包通过sftp分别上传到各自文件夹中，再分别进入每个文件夹中解压（tar -zxvf 压缩包名称）

cd /var

mkdirdockerfile

cd dockerfile

mkdir Jenkins

cd Jenkins

vi dockerfile

**内容如下:**

FROM jenkins

USER root

#清除了基础镜像设置的源，切换成阿里云的jessie源

RUN echo '' > /etc/apt/sources.list.d/jessie-backports.list \

&& echo "deb http://mirrors.aliyun.com/debian jessie main contrib non-free" > /etc/apt/sources.list \

&& echo "deb http://mirrors.aliyun.com/debian jessie-updates main contrib non-free" >> /etc/apt/sources.list \

&& echo "deb http://mirrors.aliyun.com/debian-security jessie/updates main contrib non-free" >> /etc/apt/sources.list

#更新源并安装缺少的包

RUN apt-get update && apt-get install -y libltdl7

ARG dockerGid=999

RUN echo "docker:x:${dockerGid}:jenkins" >> /etc/group \

USER Jenkins

docker build -t my\_jenkins .

docker run -p 8080:8080 -p 50000:50000 --restart=always -v /etc/timezone:/etc/timezone -v /etc/localtime:/etc/localtime -v /var/docker\_data/jenkins/jenkins\_home:/var/jenkins\_home -v /var/docker\_data/jenkins/settings:/var/settings -v /var/run/docker.sock:/var/run/docker.sock -v $(which docker):/usr/bin/docker -v /usr/java/jdk1.8.0\_144:/usr/java/jdk1.8.0\_144 -v /usr/maven/apache-maven-3.5.0:/usr/maven/apache-maven-3.5.0 -v /usr/tomcat/apache-tomcat-8.0.45:/usr/tomcat/apache-tomcat-8.0.45 --name local\_jenkins -d my\_jenkins

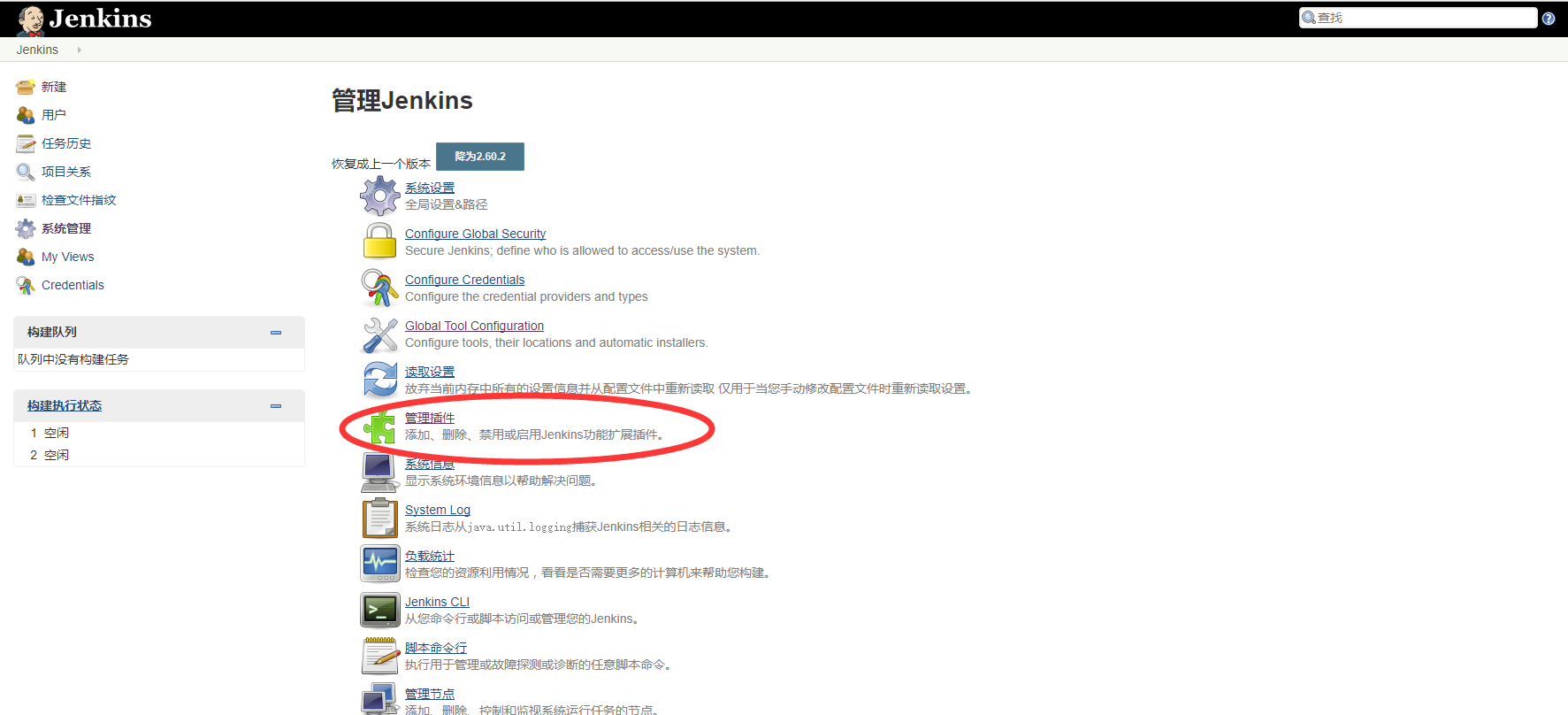
IP+8080进入页面

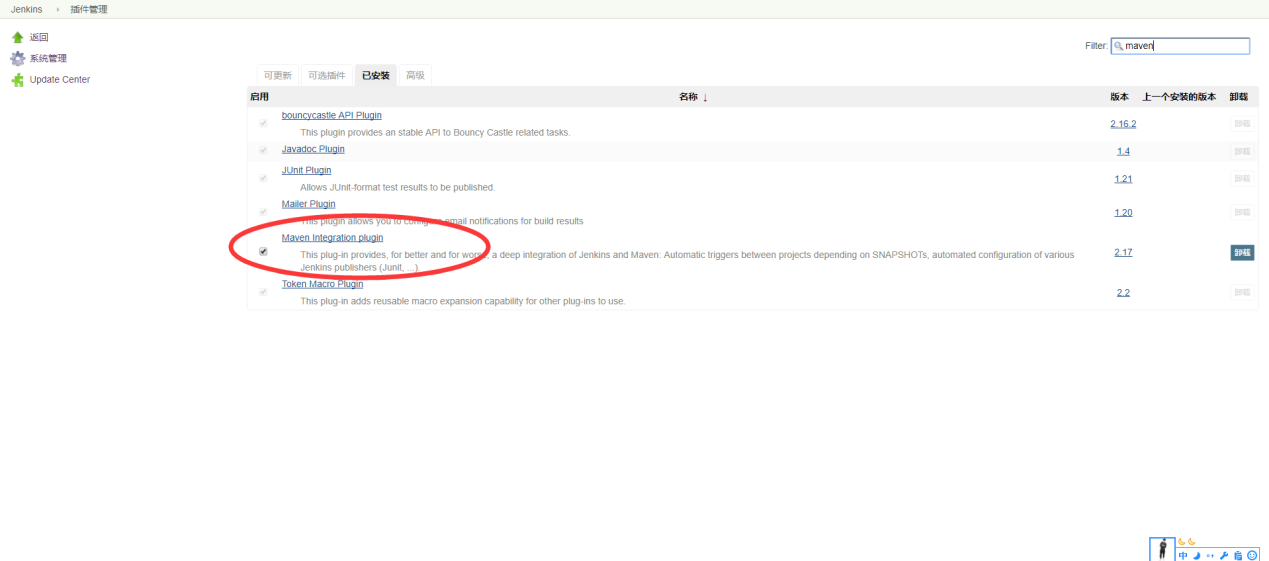
docker exec -it local\_jenkins /bin/bash(进入容器)

cat页面上初始密码位置

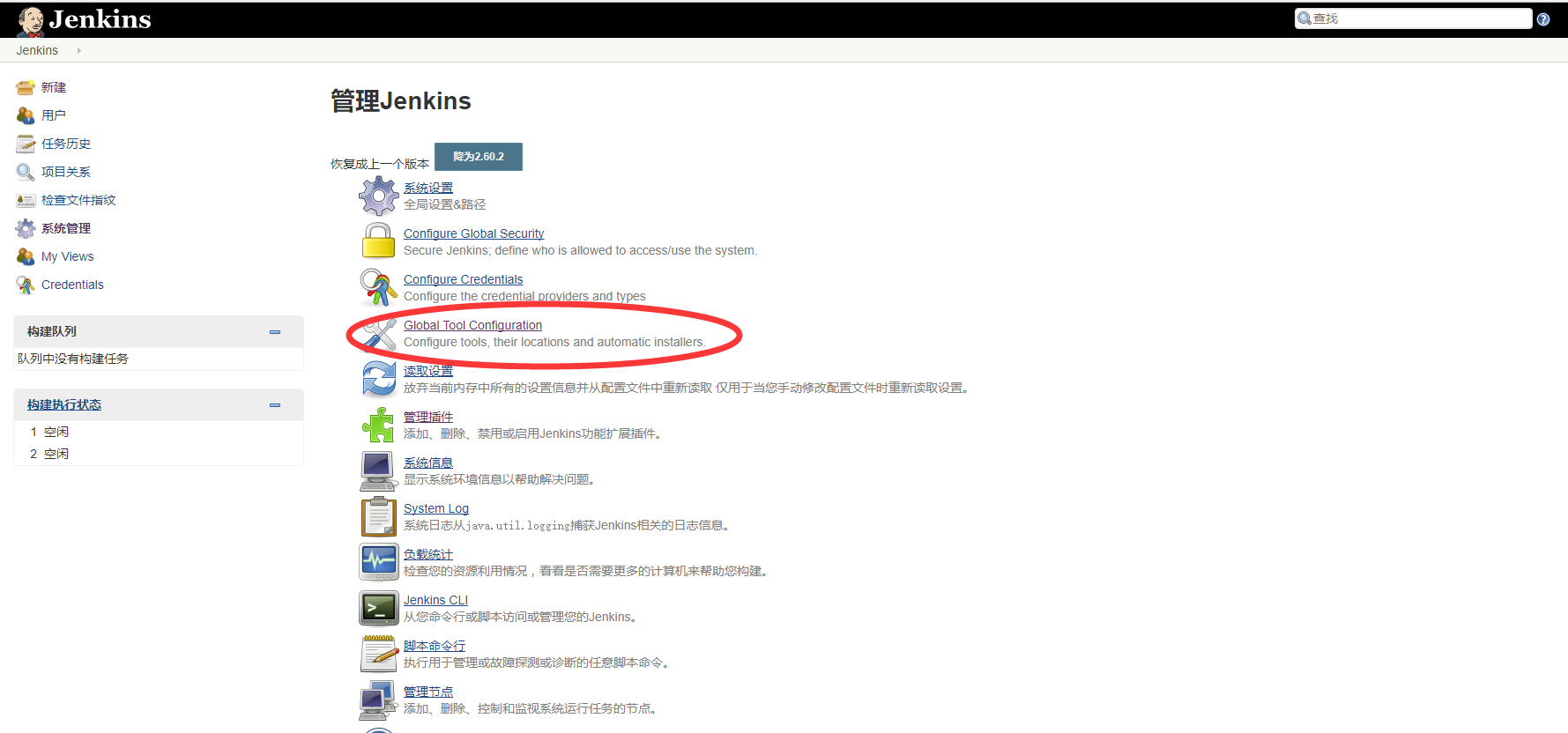
exit(退出容器)

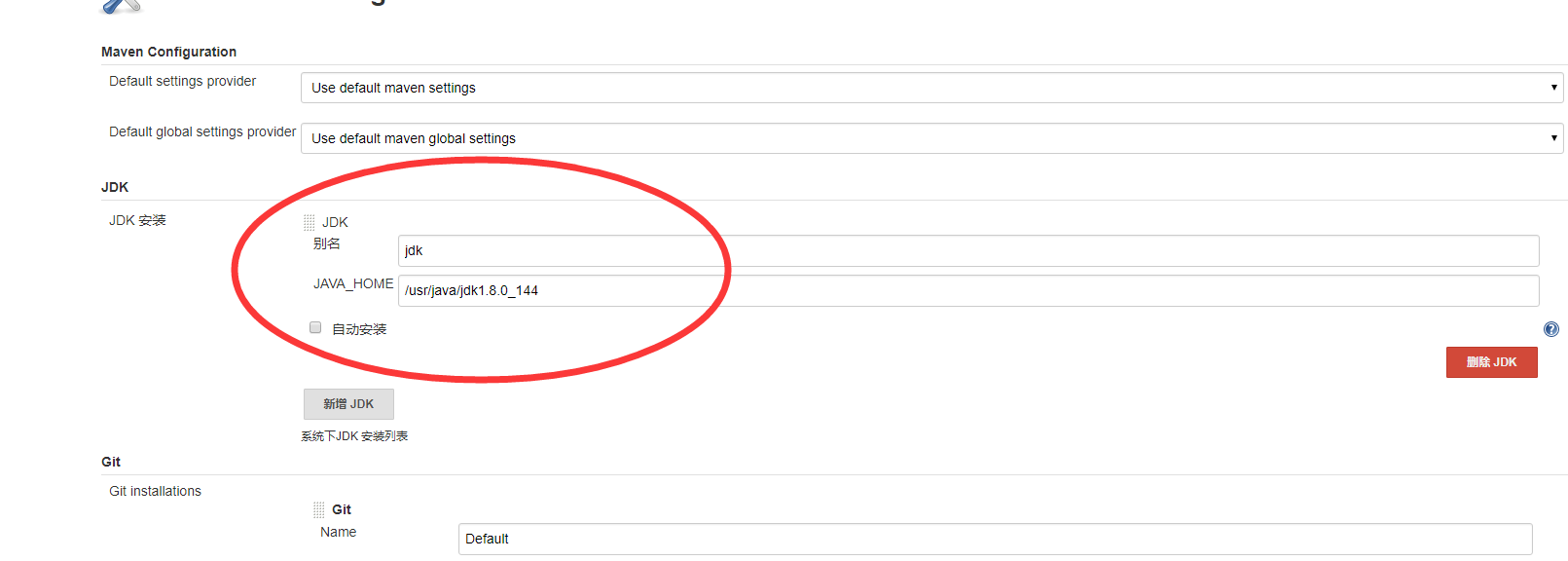
在jenkins中安装maven插件

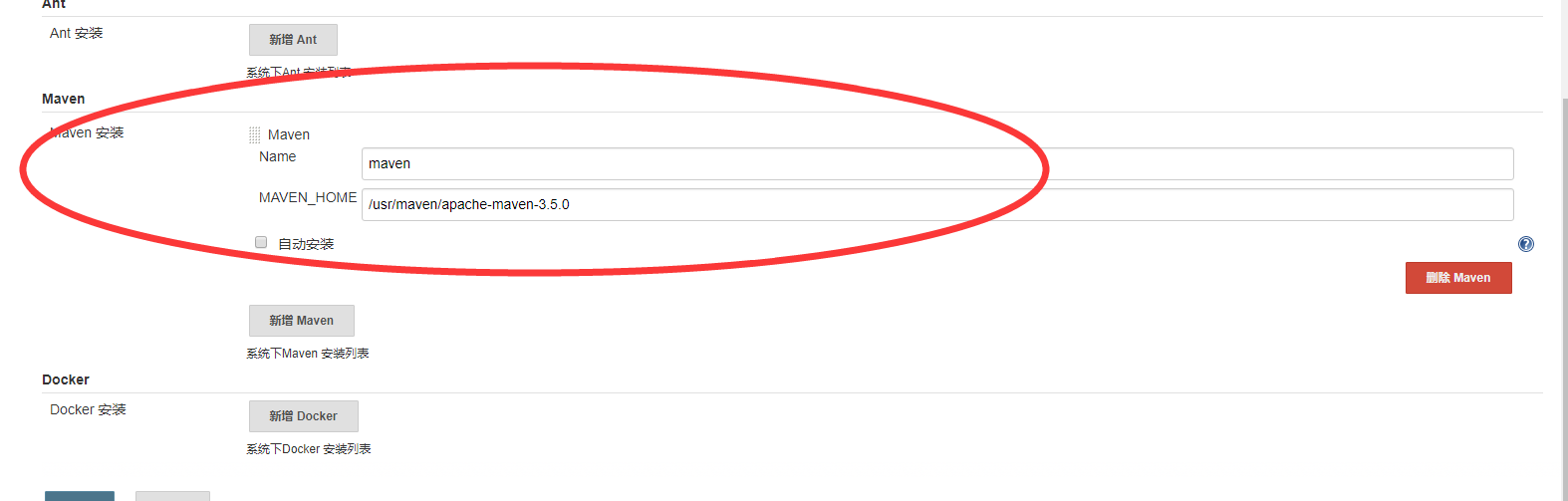




配置maven jdk环境变量







在 stu的下面目录创建\src\main\resources下新建dockerfile

内容如下：

FROM ubuntu:16.04

WORKDIR /usr

RUN mkdir java

RUN mkdir tomcat

WORKDIR /usr/java

RUN mkdirjdk

WORKDIR /usr/java/jdk

ADD /java/jdk1.8.0\_144 /usr/java/jdk

WORKDIR /usr/tomcat

ADD /tomcat/apache-tomcat-8.0.45 /usr/tomcat

ENV JAVA\_HOME=/usr/java/jdk

ENV JAVA\_BIN=/usr/java/jdk/bin

ENV PATH=$PATH:$JAVA\_HOME/bin

ENV CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

WORKDIR /usr/tomcat/webapps/ROOT

RUN rm -rf \*

WORKDIR /usr

ADD stu.war /usr/tomcat/webapps/stu.war

EXPOSE 9999

CMD /usr/tomcat/bin/catalina.sh run

打开pom.xml文件修改，增加如下配置节点：

<project>

[...]

<build>

[...]

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.6.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

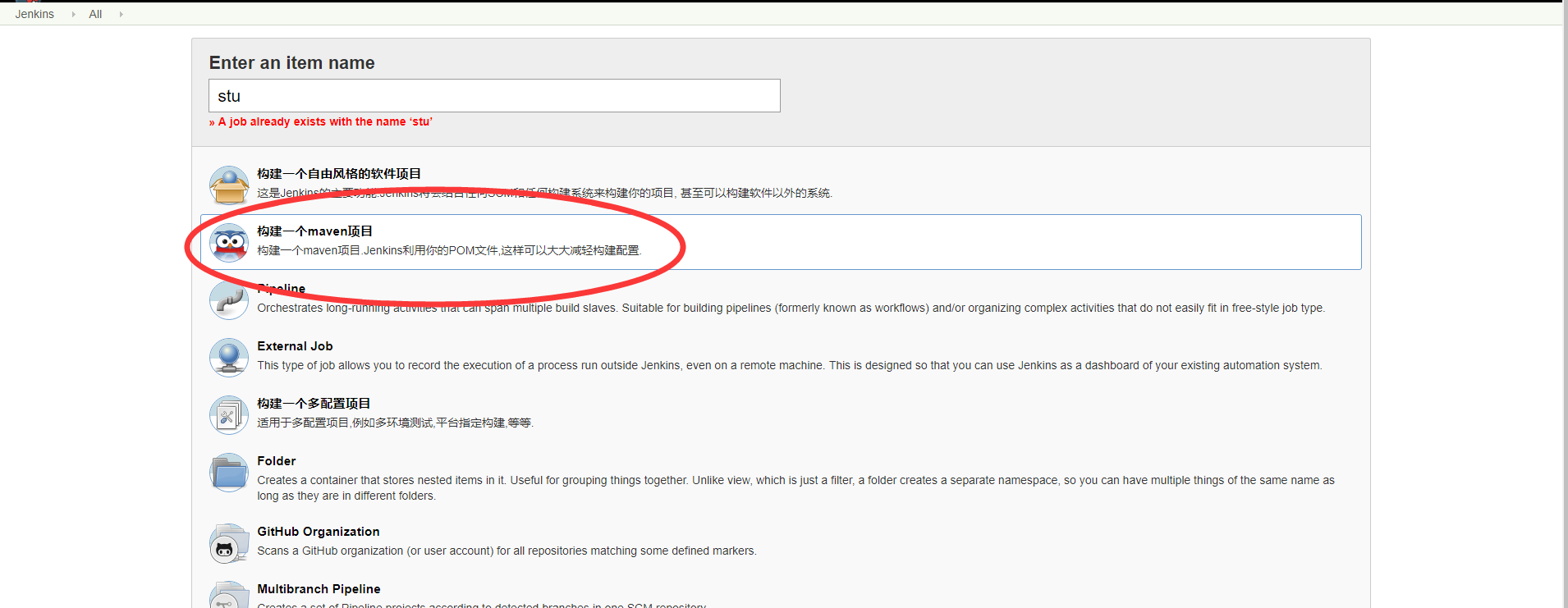
[...]

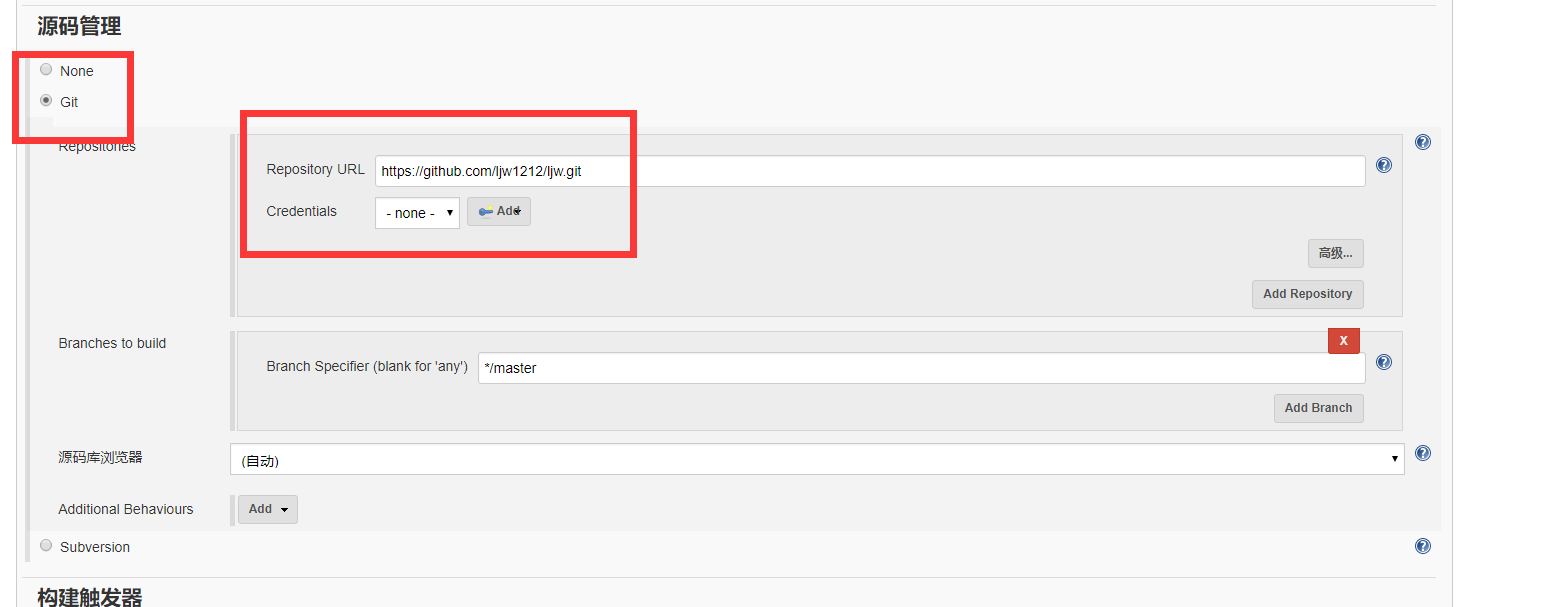
</build>

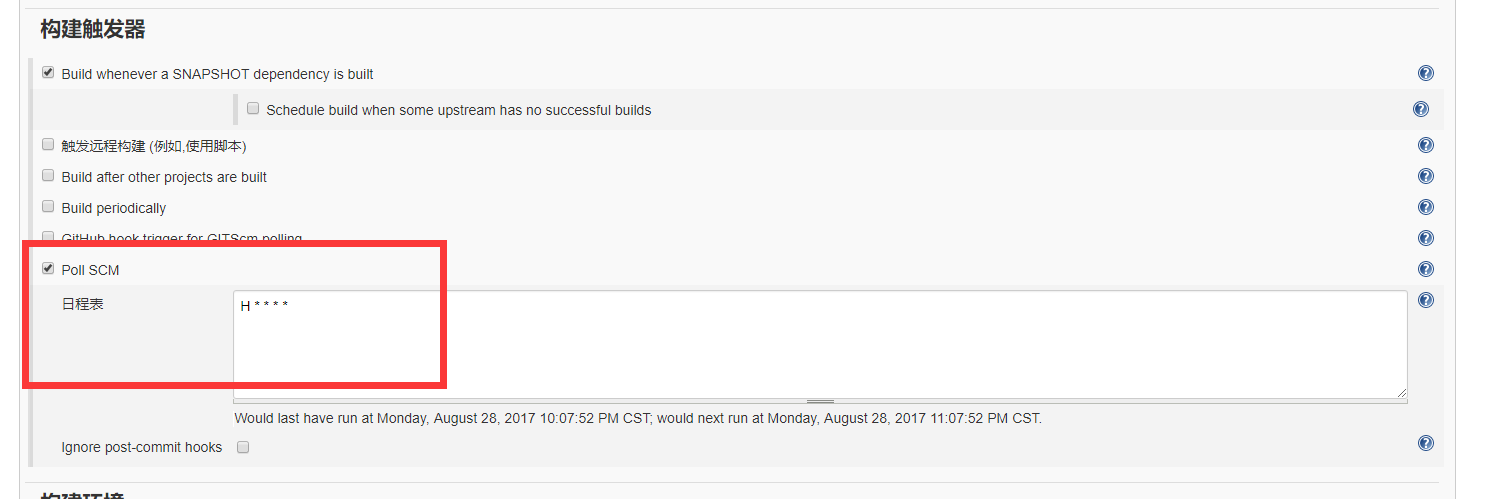
[...]

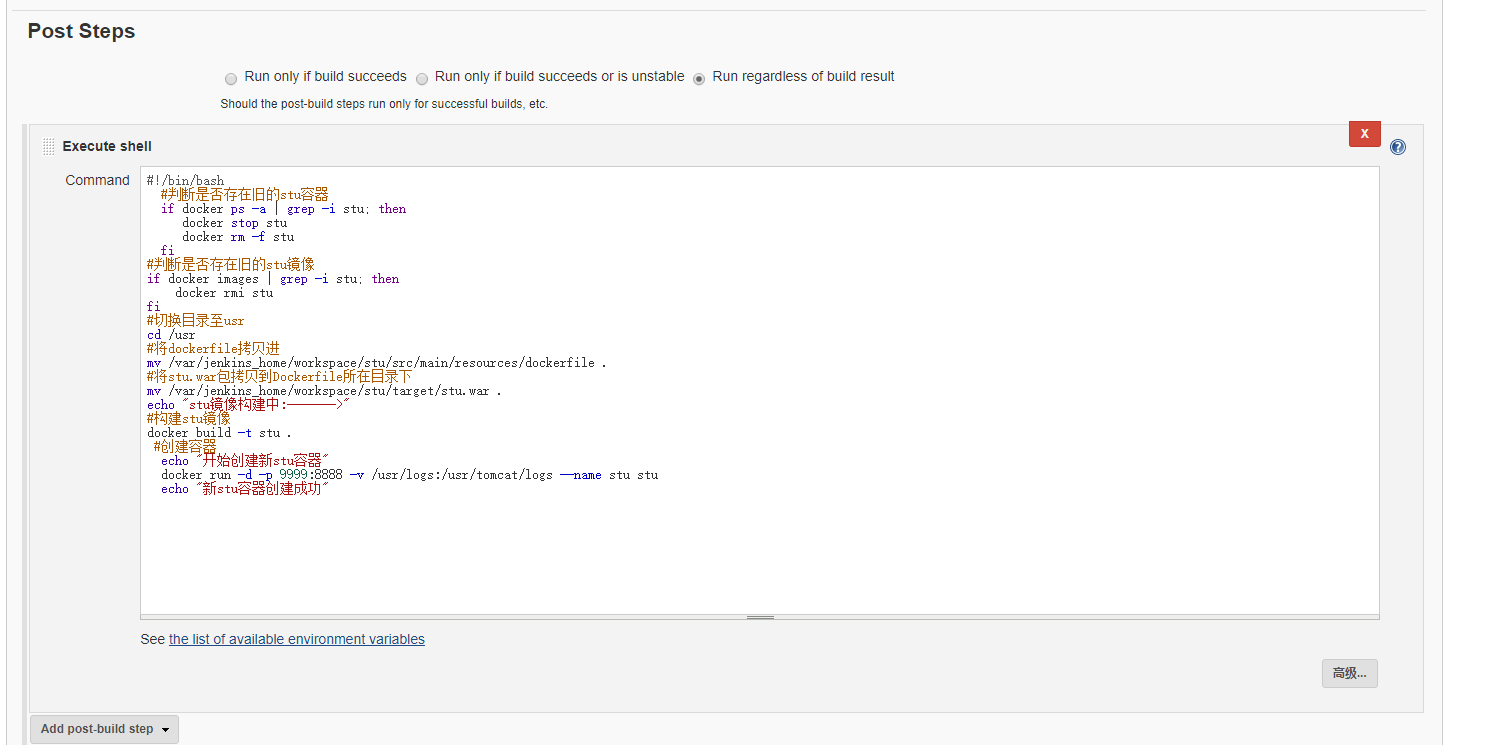
</project>

将stu上传至gitbub（<https://github.com/ljw1212/ljw.git>）









#!/bin/bash

#判断是否存在旧的stu容器

if docker ps -a | grep -istu; then

docker stop stu

docker rm -f stu

fi

#判断是否存在旧的stu镜像

if docker images | grep -istu; then

docker rmistu

fi

#切换目录至usr

cd /usr

#将dockerfile拷贝进

mv /var/jenkins\_home/workspace/stu/src/main/resources/dockerfile .

#将stu.war包拷贝到Dockerfile所在目录下

mv /var/jenkins\_home/workspace/stu/target/stu.war .

echo "stu镜像构建中:------->"

#构建stu镜像

docker build -t stu .

#创建容器

echo "开始创建新stu容器"

docker run -d -p 9999:8888 -v /usr/logs:/usr/tomcat/logs --name stustu

echo "新stu容器创建成功"

保存构建