**Docker之jenkins 持续部署**

**先按环境可以参考如下，尽量将jdk、tomcat、maven、按在同一目录下，方便后面部署。**

**ubuntu 安装jdk 的两种方式:**

**1:通过ppa(源) 方式安装.**

**2:通过官网下载安装包安装.**

这里推荐第1种,因为可以通过 apt-get upgrade 方式方便获得jdk的升级

使用ppa/源方式安装

1.添加ppa

sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

2.安装oracle-java-installer

　jdk7

sudo apt-get install oracle-java7-installer

（这个地方如果出现问题，可以直接先下载好包放在相应的位置）

　jdk8

sudo apt-get install oracle-java8-installer

安装器会提示你同意 oracle 的服务条款,选择 ok

然后选择yes 即可

如果你懒,不想自己手动点击.也可以加入下面的这条命令,默认同意条款:

JDK7 默认选择条款

echo oracle-java7-installer shared/accepted-oracle-license-v1-1 select true | sudo /usr/bin/debconf-set-selections

JDK8 默认选择条款

echo oracle-java8-installer shared/accepted-oracle-license-v1-1 select true | sudo /usr/bin/debconf-set-selections

 接下会是等待(依个人网速定)

如果你因为防火墙或者其他原因,导致installer 下载速度很慢,可以中断操作.然后下载好相应jdk的tar.gz 包,放在:

   /var/cache/oracle-jdk7-installer             (jdk7)

   /var/cache/oracle-jdk8-installer              (jdk8)

下面,然后安装一次installer. installer 则会默认使用 你下载的tar.gz包

3.设置系统默认jdk

JDk7

sudo update-java-alternatives -s java-7-oracle

JDK8

sudo update-java-alternatives -s java-8-oracle

如果即安装了jdk7,又安装了jdk8,要实现两者的切换,可以:

　　jdk8 切换到jdk7

sudo update-java-alternatives -s java-7-oracle

　　jdk7 切换到jdk8

sudo update-java-alternatives -s java-8-oracle

**4.测试jdk 是是否安装成功:**

java -version

javac -version

**直接下载jdk压缩包方式安装(这里只介绍jdk7的,jdk8 的原理完全一致)**

　分为下面5个步骤

   1.官网下载JDK

   2.解压缩,放到指定目录

   3.配置环境变量

   4.设置系统默认JDK

　5. 测试jdk

1.官网下载JDK

     地址: http://www.oracle.com/technetwork/articles/javase/index-jsp-138363.html

　　选择相应的 .gz包下载 （ps：使用wget下载可能解压的时候会出错，原因是那个网页好像有个什么验证哦）

2. 解压缩,放到指定目录(以jdk-7u60-linux-x64.gz为例)

　　创建目录:

sudo mkdir /usr/lib/jvm

　加压缩到该目录:

sudo tar -zxvf jdk-7u60-linux-x64.gz -C /usr/lib/jvm

(ps:如果出现问题，可以先用file+包名 检验压缩包是否正常)

**3.修改环境变量:**

sudo vim ~/.bashrc

　文件的末尾追加下面内容:

#set oracle jdk environment  
export JAVA\_HOME=/usr/lib/jvm/jdk1.7.0\_60 ## 这里要注意目录要换成自己解压的jdk 目录

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

export PATH=${JAVA\_HOME}/bin:$PATH

　使环境变量马上生效

source ~/.bashrc

**4.设置系统默认jdk 版本**

sudo update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk1.7.0\_60/bin/java 300

sudo update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk1.7.0\_60/bin/javac 300

sudo update-alternatives --install /usr/bin/jar jar /usr/lib/jvm/jdk1.7.0\_60/bin/jar 300

sudo update-alternatives --install /usr/bin/javah javah /usr/lib/jvm/jdk1.7.0\_60/bin/javah 300

sudo update-alternatives --install /usr/bin/javap javap /usr/lib/jvm/jdk1.7.0\_60/bin/javap 300

　然后执行:

sudo update-alternatives --config java

    若是初次安装jdk,会有下面的提示

   There is only one alternative in link group java (providing /usr/bin/java):   
    /usr/lib/jvm/jdk1.7.0\_60/bin/java

**否者,选择合适的jdk**

**5.测试jdk**

java -version  
  
java version "1.7.0\_60"

  Java(TM) SE Runtime Environment (build 1.7.0\_60-b19)  
  Java HotSpot(TM) 64-Bit Server VM (build 24.60-b09, mixed mode)

**2.安装Tomcat**

1.新建tomcat文件夹(路径：/usr/tomcat)

makdir tomcat

2.将tomcat安装包移到安装目录/usr/tomcat，解压

tar -zxvf  apache-tomcat-7.0.27.tar.gz

(堵上下载地址:[http://tomcat.apache.org/download-70.cgi](http://blog.csdn.net/xuexiaoxu1990/article/details/ ))

3.启动Tomcat

  # cd /usr/tomcat/apache-tomcat-7.0.27/bin

  # ./startup.sh

4.测试Tomcat

打开防火墙,使外部能访问

 # /sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT

 # service iptables save

 # service iptables restart

或直接修改文件/etc/sysconfig/iptables.

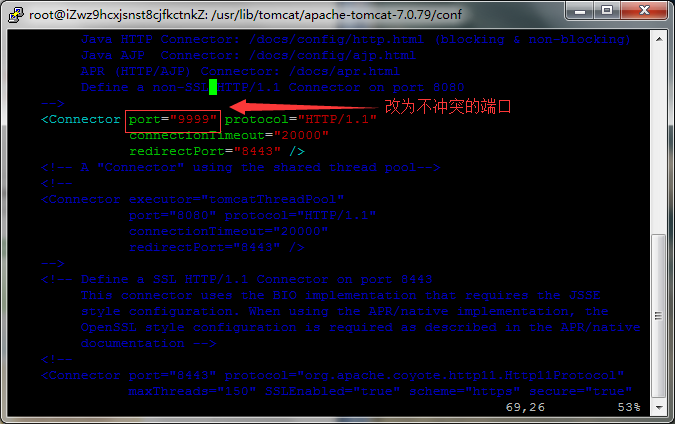
 # vi /etc/sysconfig/iptables

-A INPUT -p tcp -m tcp --dport 8080 -j ACCEPT

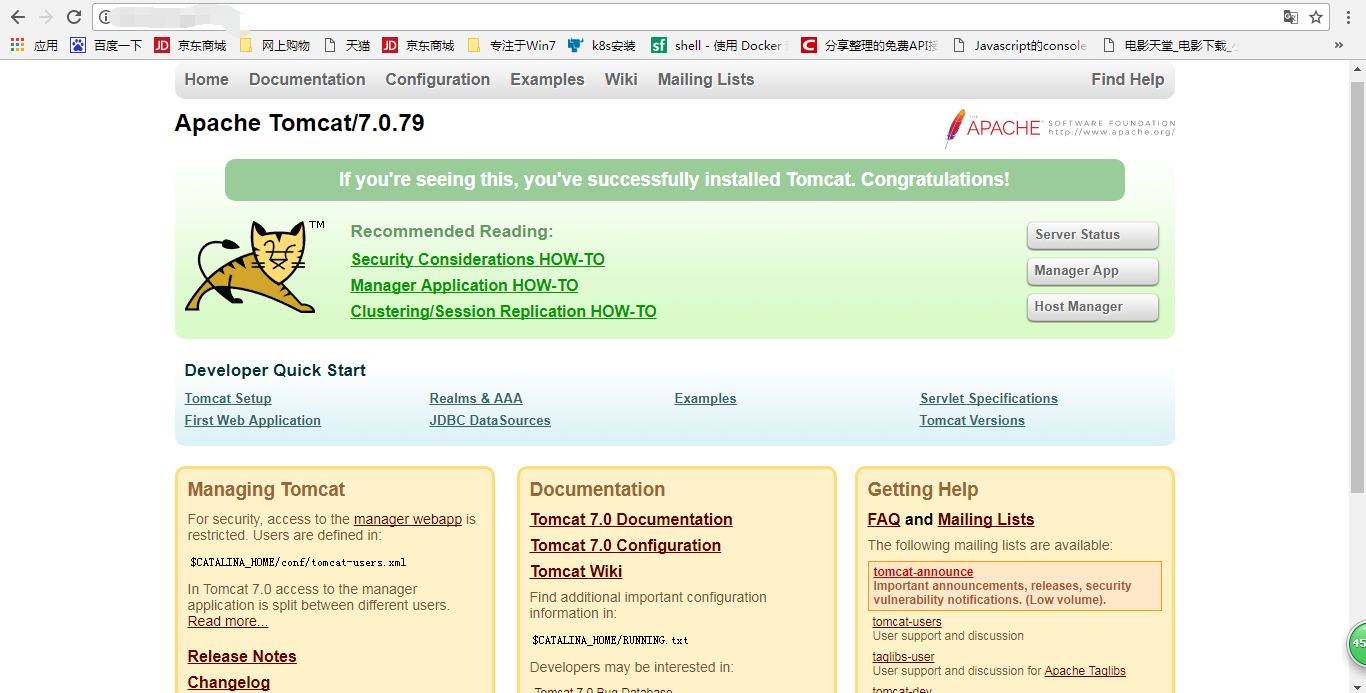
# service iptables restart

在本机浏览器地址栏中输入：http://localhost:8080

(PS:tomcat默认端口为8080，如和其他应用端口号冲突，可在/usr/tomcat/apache-tomcat-7.0.27/conf/server.xml中修改，我改成了9999.)



出现tomcat的页面表示安装成功！



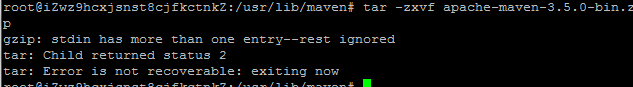
**4.安装Maven**

1.将Maven安装包移到安装目录/usr/，解压

tar -zxvf  apache-maven-3.3.9-bin.zip

(附上下载地址：<http://maven.apache.org/download.cgi>)

遇到的问题



解决：原来tar也是间接调用了gzip，而gzip的使用是有限制的：

也就是说单文件用deflation压缩的包才可以用gzip解压，好在有workaround ---- unzip

unzip用法也非常简单，直接敲unzip 就有提示，最后一句命令搞定：

unzip apache-maven-3.5.0-bin.zip

-d为指定输出文件夹，不指定貌似就放到当前文件夹了。

2.配置环境变量

vim  /etc/profile

末尾添加以下变量：

export MAVEN\_HOME=/usr/apache-maven-3.3.9  
export PATH=$PATH:$MAVEN\_HOME/bin

保存退出！

 source /etc/profile

刷新一下以便生效！

3.验证安装效果

输入mvn -v

出现类似下面的信息，说明maven安装成功！

Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-10T08:41:47-08:00)  
Maven home: /usr/apache-maven-3.3.9  
Java version: 1.7.0\_79, vendor: Oracle Corporation  
Java home: /usr/jdk/jdk1.7.0\_79/jre  
Default locale: en\_US, platform encoding: UTF-8  
OS name: "linux", version: "2.6.32-642.el6.x86\_64", arch: "amd64", family: "unix"

**一、创建jenkins容器:**

1.拉取jeknin镜像

sudo docker pull jenkins

2.创建一个jenkins目录

sudo mkdir /jenkins

3.在jenkins目录下创建一个Dockerfile

sudo touch Dockerfile

cp/usr/lib/x86\_64-linux-gnu/libltdl.so.7./

4.为了在jenkins容器中具有执行sudo命令的权限，在Dockerfile添加如下内容:

FROM jenkins:latest

USER root

RUN apt-get update

RUN apt-get install -y sudo

RUN rm -rf /var/lib/apt/lists/\*

RUN echo "jenkins ALL=NOPASSWD: ALL" >> /etc/sudoers

ADD libltdl.so.7 /usr/lib/x86\_64-linux-gnu/libltdl.so.7

#加这一句是因为jenkins容器里好像少了这个东西，少了就不能执行docker语句

USER jenkins  
5.创建新的jenkins镜像

sudo docker build -t zss/jenkins:1.0 .

6.创建jenkins容器下面主机目录改成你自己安装目录

sudo docker run -p 8080:8080 --name jenkins

-v /usr/bin/docker:/usr/bin/docker

-v /var/run/docker.sock:/var/run/docker.sock                                              //将宿主机中docker挂载到jenkins容器中，以便在jenkins容器中执行docker命令

-v /usr/jdk/jdk1.7.0\_25:/usr/java/jdk1.7.0\_25                                            //将宿主机中jdk挂载到jenkins容器中

-v /usr/tomcat/apache-tomcat-7.0.29:/usr/tomcat/apache-tomcat-7.0.29     //将宿主机中tomcat挂载到jenkins容器中

-v /usr/maven/apache-maven-3.3.9:/usr/maven/apache-maven-3.3.9           //将宿主机中tomcat挂载到jenkins容器中

zss/jenkins:1.0

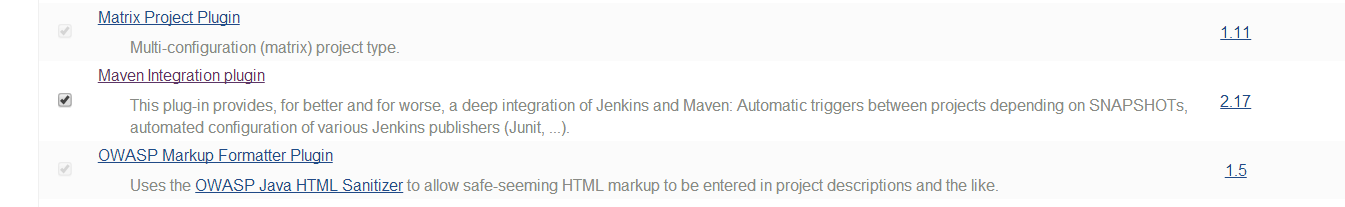
7.在浏览器中输入http://localhost:8080,填写启动jenkins时出现的密码

会提示你初始密码位置，容器里面cat即可；

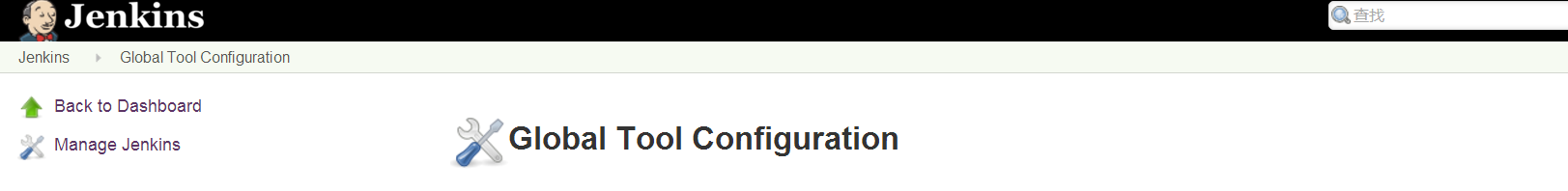
**二、配置jenkins:**

1.安装maven插件安装这个名字插件Maven Integration plugin

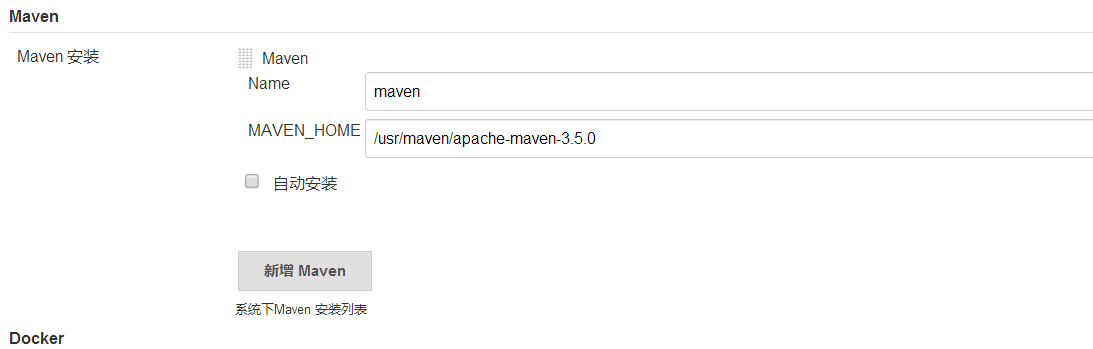




 2.在jenkins中配置jdk、maven如图：

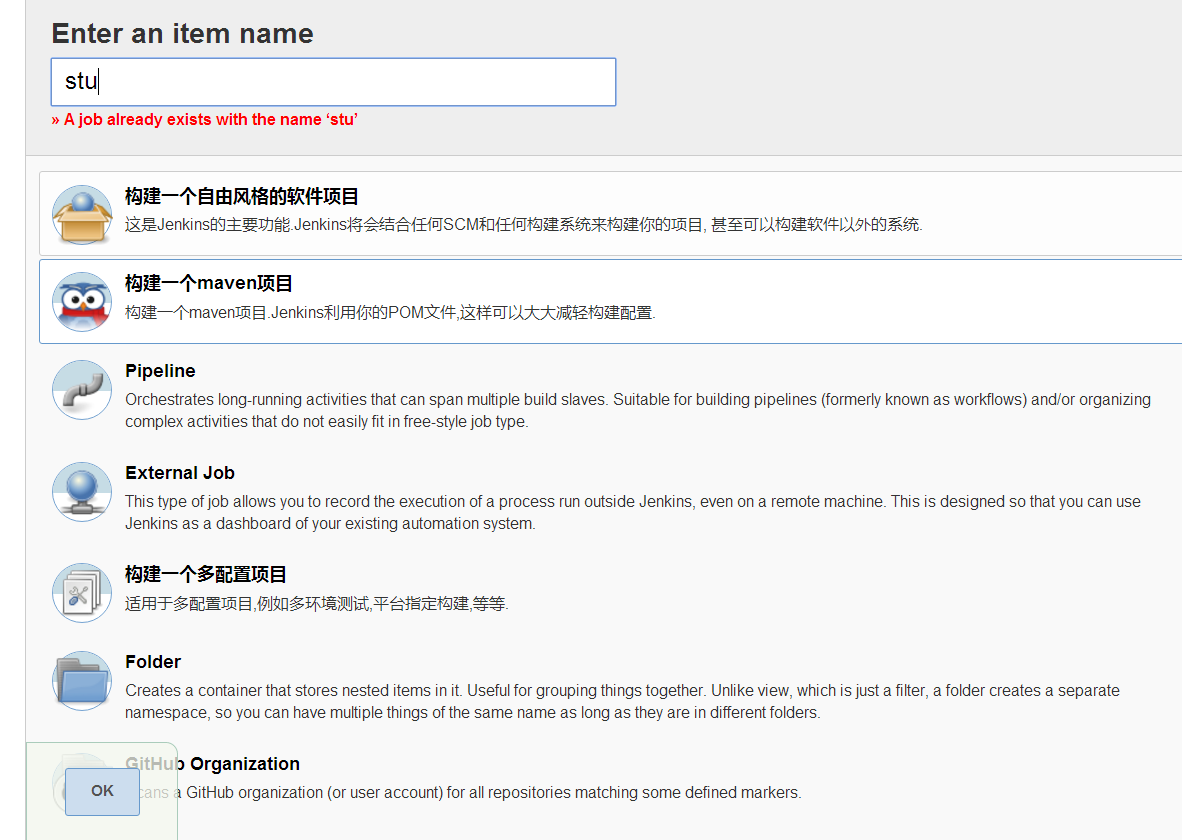


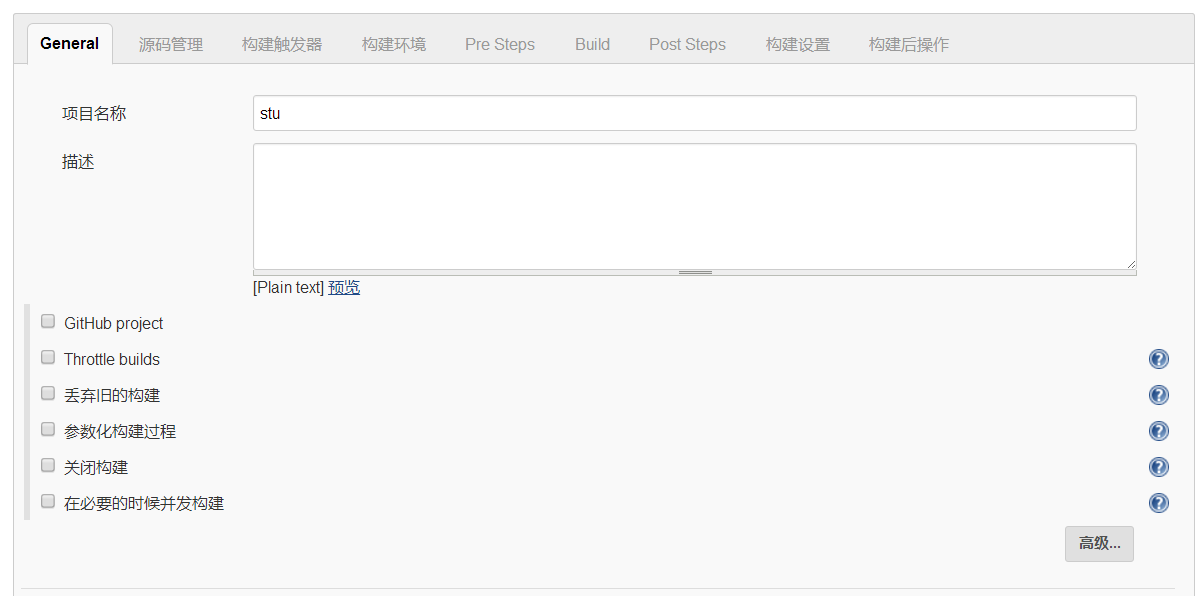


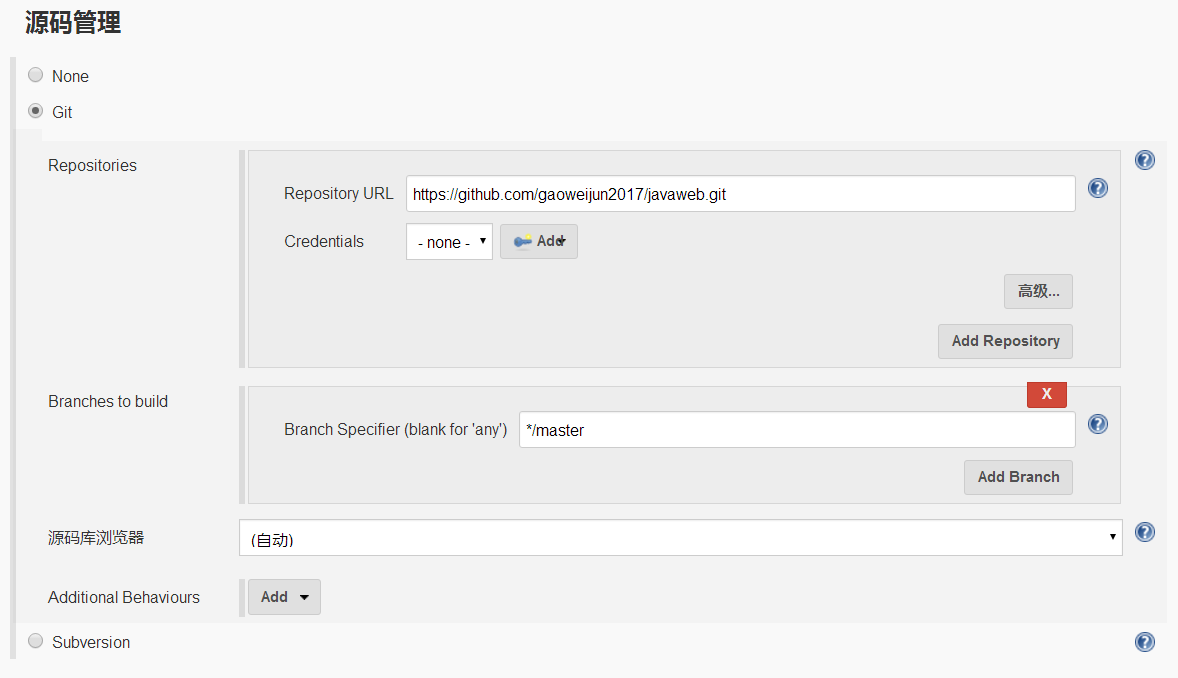


**三、配置Java web项目:**

1.创建一个maven风格的stu项目







<https://github.com/gaoweijun2017/javaweb.git>

这个是老师那个stu那个应用，本来这个是要做相应的添加和修改，这个git是改好之后的

数据库也是我的数据库的地址，可自行修改

详细如下

在 stu的下面目录创建\src\main\resources\docker文件夹 在里面创建Dockerfile

#使用ubuntu:16.04作为基础镜像

FROM ubuntu:16.04

#指定镜像创建者信息

MAINTAINER test xxxx@xxx.com

#创建日期

ENV REFRESHED\_AT 2017-2-15

#切换镜像目录，进入/usr目录

WORKDIR /usr

#在/usr/下创建jdk目录,用来存放jdk文件

RUN mkdir jdk

#在/usr/下创建tomcat目录，用来存放tomcat

RUN mkdir tomcat

#切换镜像的目录至/usr/jdk

WORKDIR /usr/jdk/

#将docker的jdk安装目录下的文件拷至镜像的/usr/

ADD jdk/jdk1.8.0\_131 /usr/jdk

#切换镜像的目录至/usr/tomcat

WORKDIR /usr/tomcat

#将docekr的tomcat安装目录下的文件拷至镜像的/usr/tomcat目录下

ADD tomcat/apache-tomcat-8.5.16 /usr/tomcat

#设置环境变量

ENV JAVA\_HOME=/usr/jdk

ENV JAVA\_BIN=/usr/jdk/bin

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

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

#切换工作目录到ROOT下

WORKDIR /usr/tomcat/webapps/ROOT

#删除tomcat默认的项目文

#将自己的xxx.war系统添加到docker镜像中tomcat下的ROOT目录下

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

#公布tomcat的9999端口

EXPOSE 9999

#启动tomcat

ENTRYPOINT ["../../bin/catalina.sh","run"]

还需修改pom.xml文件，不然构建会出错

# Maven错误 diamond operator is not supported in -source 1.5 (use -source 7 or higher to enable diamond operator)

添加 红色部分

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>

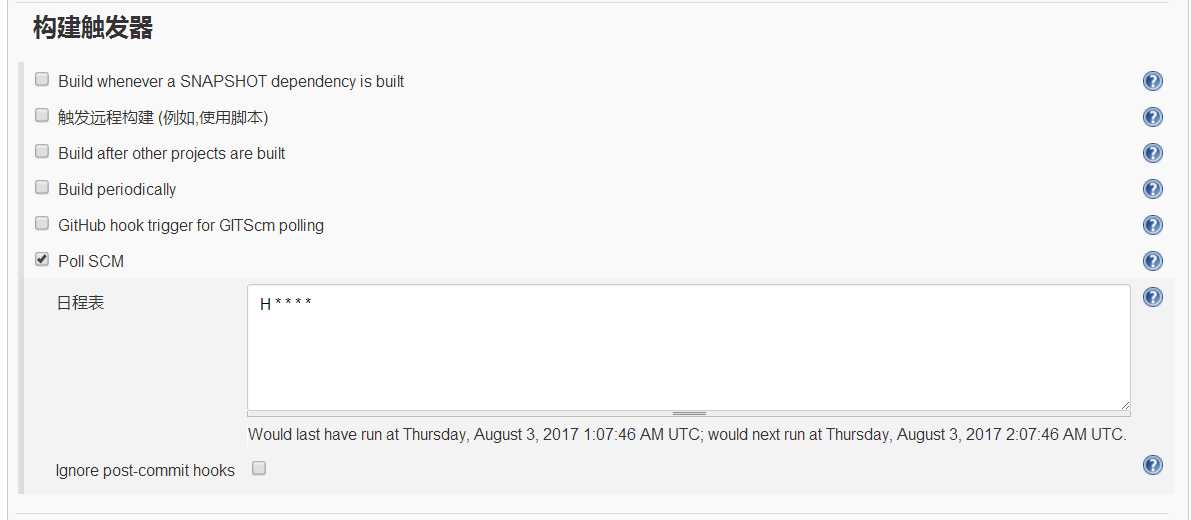
</plugins>

[...]

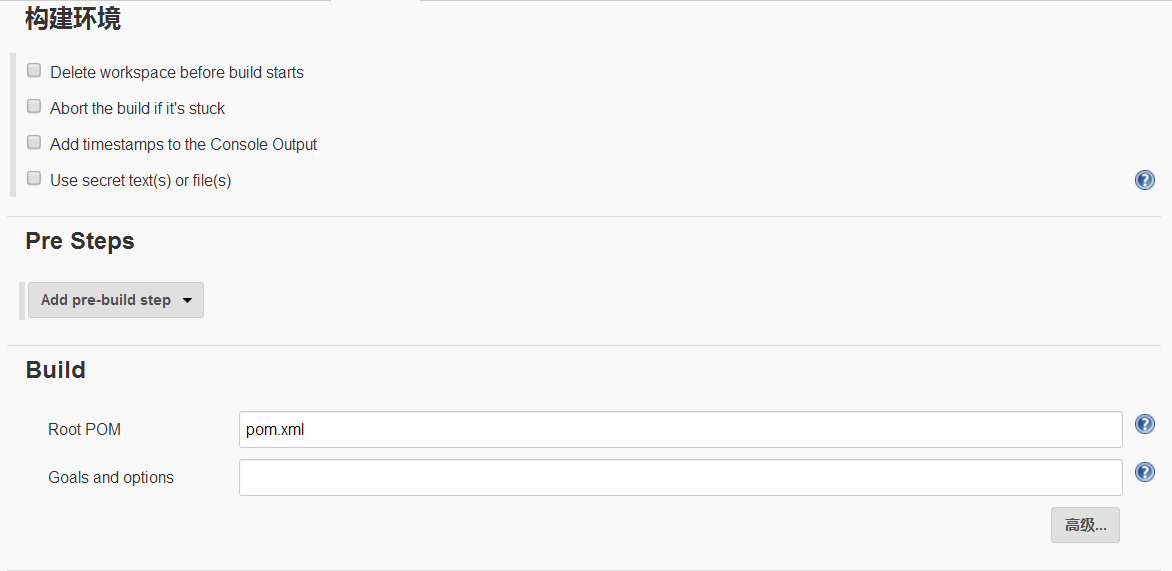
</build>

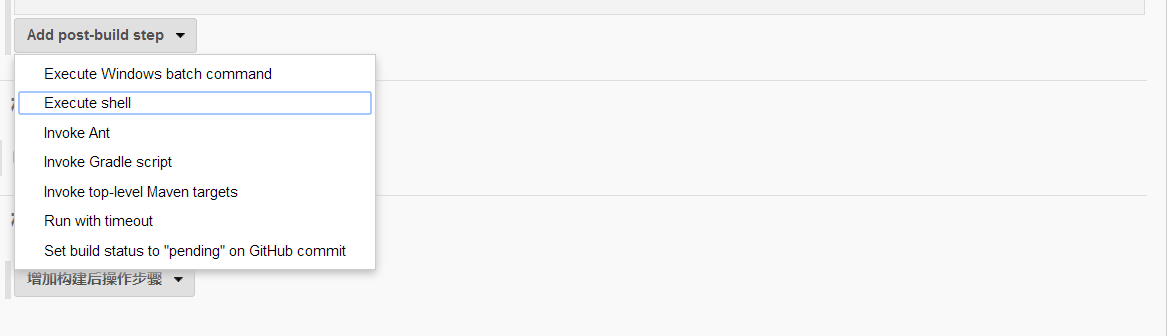
[...]

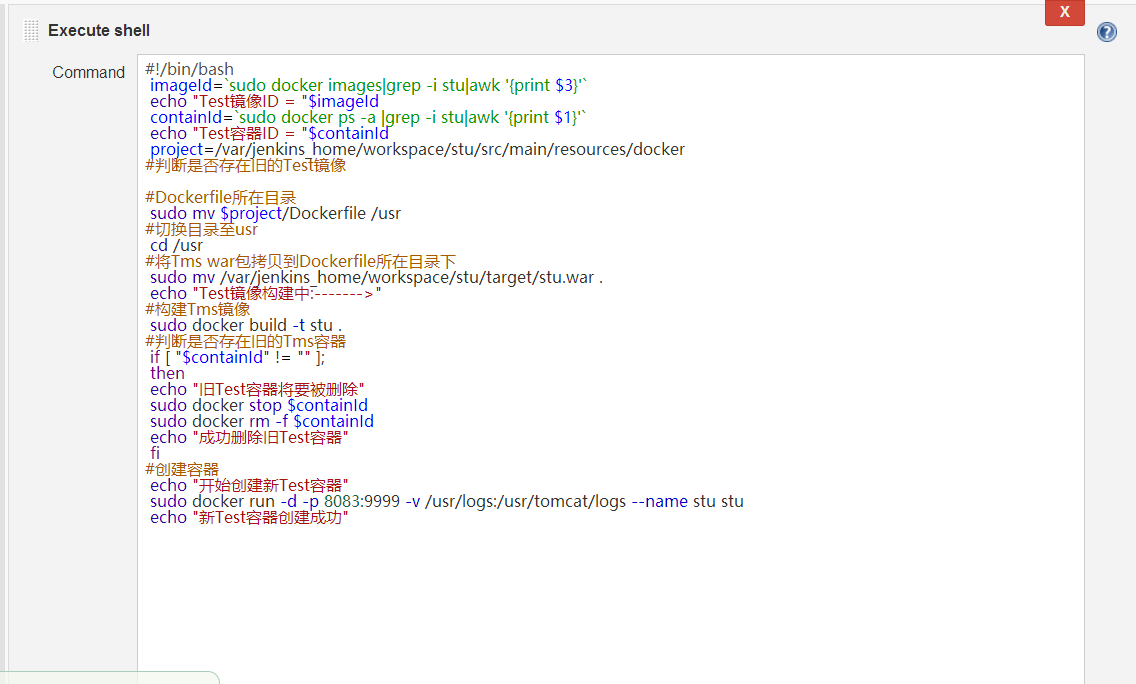
</project>



H \* \* \* \* 表示触发构建的时间间隔 代表1个小时检查一次是否有更新 需要构建H/5代表间隔5分钟







保存；

构建。