# Docker+Jenkins镜像构建以及

# 项目自动构建

目录

[1. docker安装](#_Toc32255_WPSOffice_Level1) [3](#_Toc32255_WPSOffice_Level1)

[2. 使用Docker搭建仓库](#_Toc31213_WPSOffice_Level1) [4](#_Toc31213_WPSOffice_Level1)

[3. docker安装Jenkins](#_Toc3305_WPSOffice_Level1) [4](#_Toc3305_WPSOffice_Level1)

[4. 使用Github+Docker hub自动构建](#_Toc31086_WPSOffice_Level1) [5](#_Toc31086_WPSOffice_Level1)

[5. 使用Github+阿里云自动构建](#_Toc10463_WPSOffice_Level1) [7](#_Toc10463_WPSOffice_Level1)

[6. Jenkins构建项目](#_Toc19260_WPSOffice_Level1) [10](#_Toc19260_WPSOffice_Level1)

[7. 仓库初始化](#_Toc25352_WPSOffice_Level1) [13](#_Toc25352_WPSOffice_Level1)

### docker安装

1. <http://mirrors.aliyun.com/docker-toolbox/windows/docker-toolbox/>

2 .安装

傻瓜式安装

初始化的时候会获取不到镜像，或者下载失败，可以使用翻墙软件，我用手机下载，在github上找小明vpn，下载就可以使用。下载结束，复制到cache目录下，这个时候一定要断开网络。

3.启动：

我的启动是失败的，查了一下日志

00:12:51.00:10:04.129200 GUI: UIMediumEnumerator: Medium-enumeration finished!

00:11:35.703328 ERROR [COM]: aRC=E\_UNEXPECTED (0x8000ffff) aIID={7844aa05-b02e-4cdd-a04f-ade4a762e6b7} aComponent={SessionWrap} aText={The session is not locked (session state: Unlocked)}, preserve=false aResultDetail=0

00:13:07.414974 GUI: UISelectorWindow: Geometry saved as: Origin=709x371, Size=933x306

网上的解决方法：

原因：

原因因为大多人使用的ghost系统都会破解uxtheme.dll文件

导致virtualbox启动失败验证uxtheme.dll

下载microsoft的sigcheck工具http://technet.microsoft.com/en-us/sysinternals/bb897441.aspx

运行命令sigcheck -i -a -h c:\windows\system32\uxtheme.dll显示结果第一行是 Verified:Unigned 说明是破解的

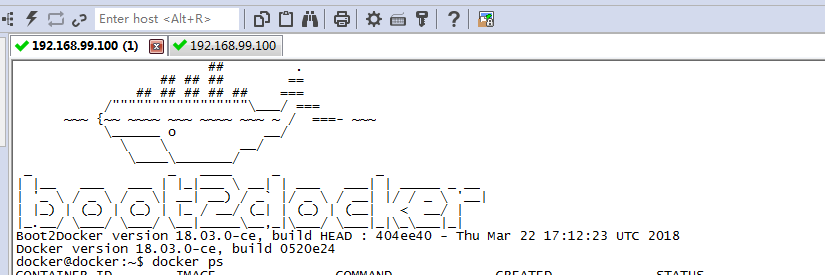
显示结果第一行是 Verified:Signed 说明是原版的

使用原版的uxtheme.dll替换c:\windows\system32\uxtheme.dll即可正常使用virtualbox

替换掉windows/system/uxtheme.dll文件

安装如果有问题，私信我，我的替换掉就已经好了。

看到这个东西基本就可以了。



我这是使用工具连接docker这个机器，没有用他们安装的时候提供的工具，因为我感觉很难以使用，docker安好之后，默认有个用户名和密码：docker ctuser，登录使用就可以了。

简单介绍几个命令

1. docker search XX，查询镜像
2. docker pull XX 下载
3. Docker run XX 启动，当然有很多参数，这个可以查文档
4. Docker push 上传至仓库

### 使用Docker搭建仓库

1.sudo mkdir -pv /opt/data/registry 创建所需的目录

2. docker run -d -p 5000:5000 -v /opt/data/registry:/var/lib/registry registry

启动容器

说明：

（1）本地没有会自己下载，输入命令就可以了

3.上传【因为现在里面没有东西】

上传这里需要注意，将镜像改为自己的

比如我将其改为127.0.0.1:5000/XX

Docker tag xxx 127.0.0.1:5000/XX 修改

docker push 127.0.0.1:5000/XX 上传

说明：

（1）这个因为是在本机上，每一个应用只是他的一个端口，所以使用localhost是不会有问题的

（2）可能会说，我在window下，为什么是127.0.0.1，因为你连的是boot2docker，所以这就不是window。

4.测试下载

docker pull 127.0.0.1:5000/XX

### docker安装Jenkins

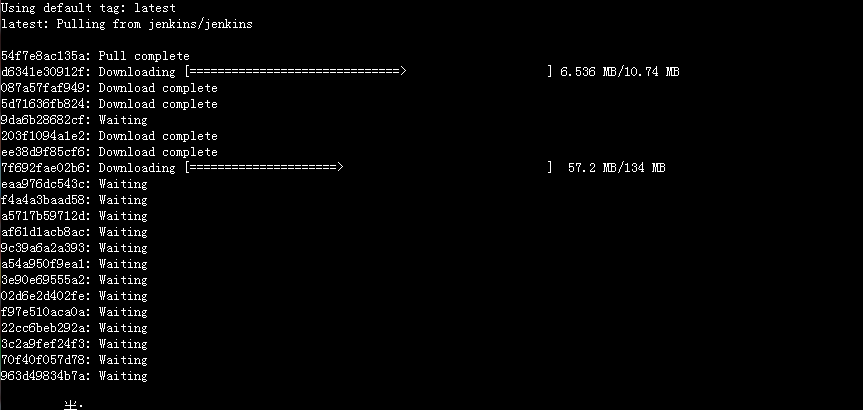
创建Jenkins

拉取镜像

（1）docker search Jenkins查询镜像

（2）下载镜像

docker pull Jenkins/jenkins



（3）启动镜像

docker run -p 8080:8080 Jenkins/jenkins

（4）打开浏览器进行访问

http://168.192.1.101:8080

（5）将日志中的密码复制过来,也是一长串字符串

（6）安装插件

（7）修改密码

（8）设置全局环境

（9）设置安全，允许注册

（10）注册用户

（11）发布项目

### 使用Github+Docker hub自动构建

1. 写Dockerfile文件

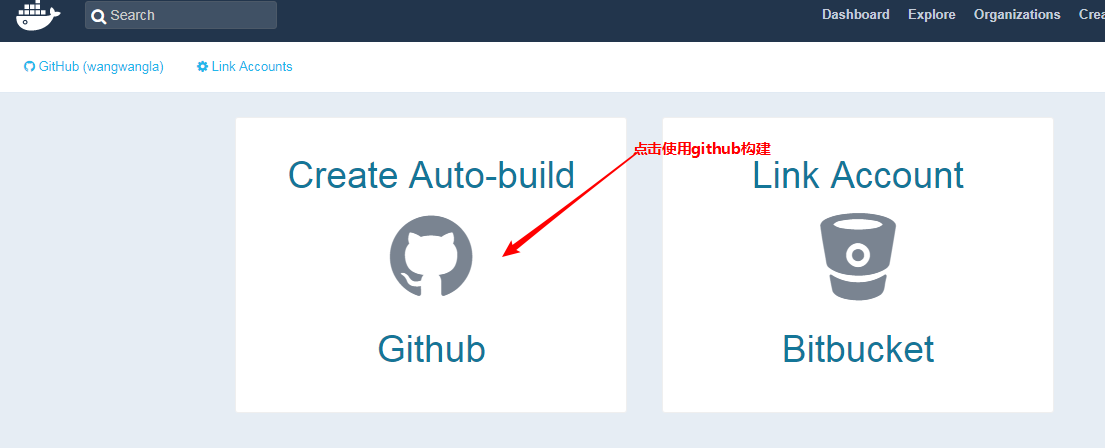
|  |
| --- |
| FROM openjdk:8-jre  MAINTAINER dfengwei@163.com  ENV JAVA\_HOME /docker-java-home  ENV CATALINA\_HOME /usr/local/tomcat  ENV PATH $CATALINA\_HOME/bin:$PATH  ENV TIME\_ZONE Asia/Shanghai  RUN mkdir -p "$CATALINA\_HOME"  WORKDIR $CATALINA\_HOME  RUN set -x \  \  # 下载Tomcat压缩文件  && wget -O tomcat.tar.gz 'https://www.apache.org/dyn/closer.cgi?action=download&filename=tomcat/tomcat-8/v8.5.23/bin/apache-tomcat-8.5.23.tar.gz' \  # 解压  && tar -xvf tomcat.tar.gz --strip-components=1 \  # 删除供Windows系统使用的.bat文件  && rm bin/\*.bat \  # 删除Tomcat压缩文件  && rm tomcat.tar.gz\* \  \  # 更改时区  && echo "${TIME\_ZONE}" > /etc/timezone \  && ln -sf /usr/share/zoneinfo/${TIME\_ZONE} /etc/localtime \  \  # 处理tomcat启动慢问题（随机数产生器初始化过慢）  && sed -i "s#securerandom.source=file:/dev/random#securerandom.source=file:/dev/./urandom#g" $JAVA\_HOME/jre/lib/security/java.security  EXPOSE 8080  CMD ["catalina.sh", "run"] |

使用自动构建，根据github中的代码进行自己构建

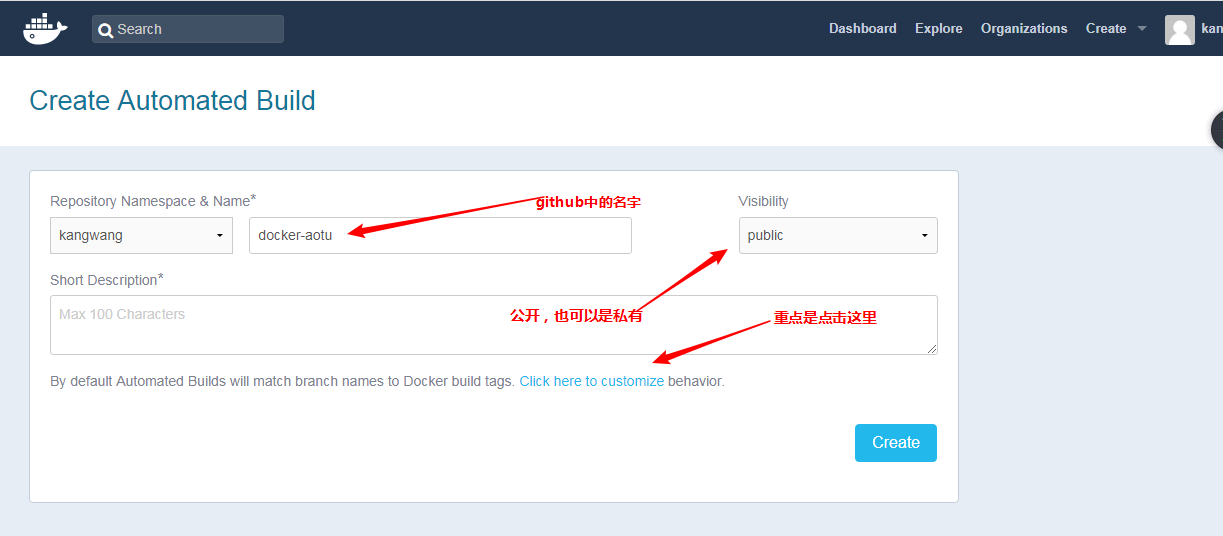
（1）在本地写好Dockerfile文件上传到github上，然后登陆自己的docker账号

（2）使用docker 账号关联github

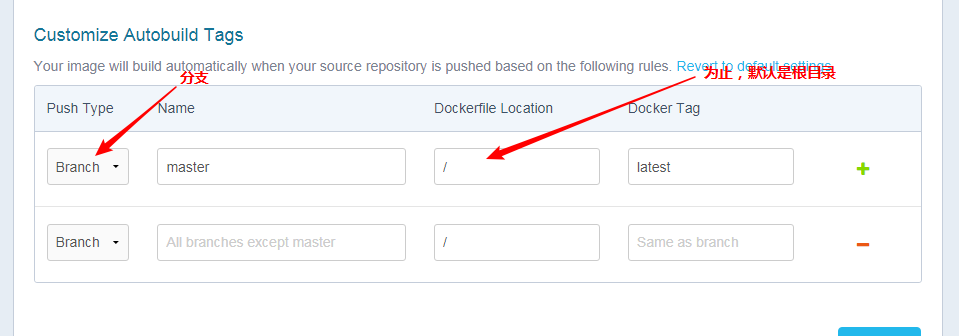
创建自动构建，选中github

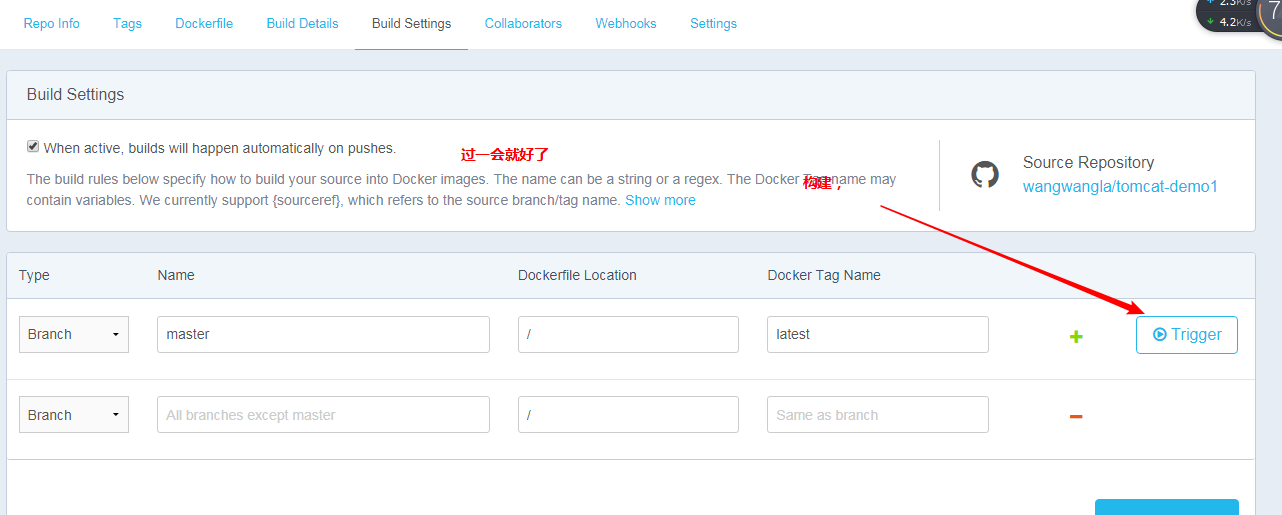


1. 选中github中的文件



1. 然后点击Click here to customize



1. 点击Create
2. 查看镜像
3. 
4. 构建成功



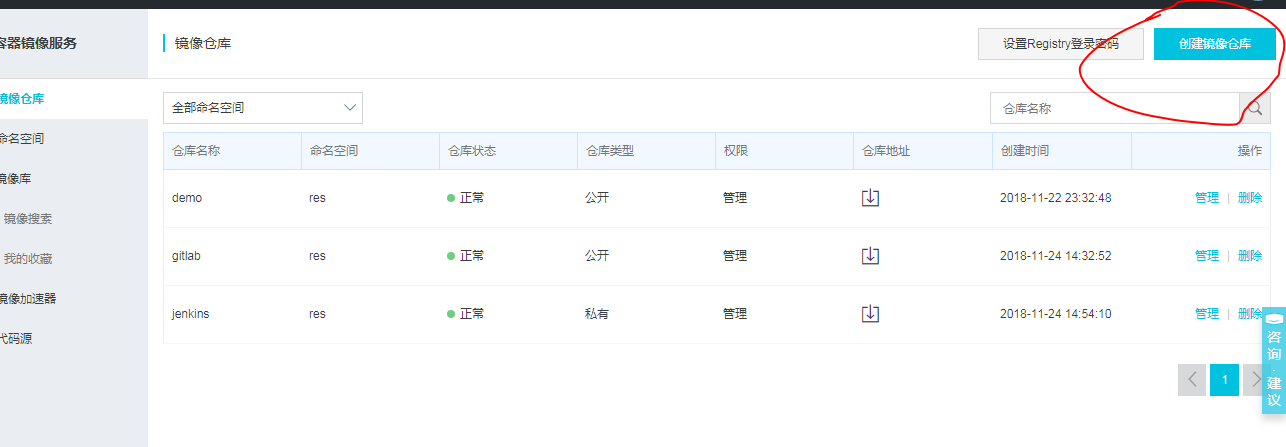
想要使用就可以下载使用了，不过还是别使用了，下载速度很无奈。

### 使用Github+阿里云自动构建

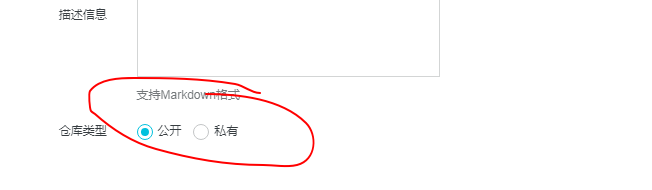
1. [https://cr.console.aliyun.com注册登录](https://www.aliyun.com/?utm_content=se_1000301881注册登录)
2. 看到这页面



1. 下来办正事





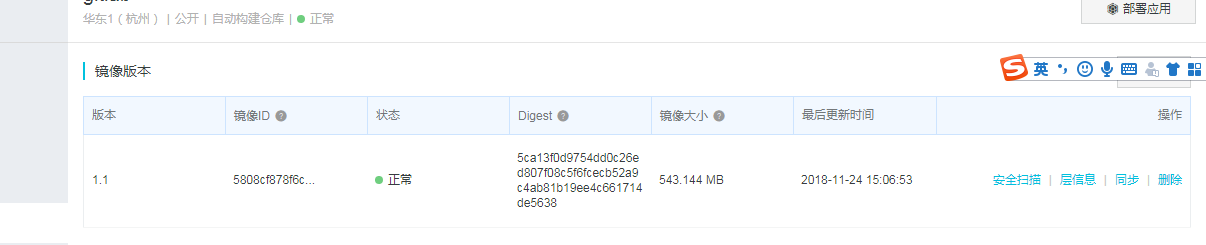








1. 基本成功了，下来可以进行扫描等操作，这当然不是重点



1. 下来说怎样下载吧

sudo docker pull registry.cn-hangzhou.aliyuncs.com/res/gitlab:版本 版本不写那就默认最新版

1. 上传【hub.docker没有上传，因为太慢了，我上传了很多次就没有上传完】

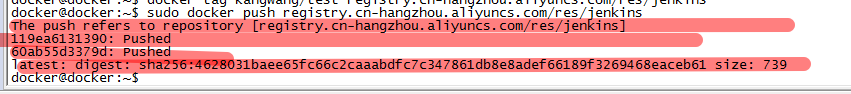
上传之前需要先登录

sudo docker login --username=18392608112 registry.cn-hangzhou.aliyuncs.com

sudo docker tag [ImageId] registry.cn-hangzhou.aliyuncs.com/res/gitlab:[镜像版本号]

sudo docker push registry.cn-hangzhou.aliyuncs.com/res/gitlab:[镜像版本号]

1. 来一个上传成功

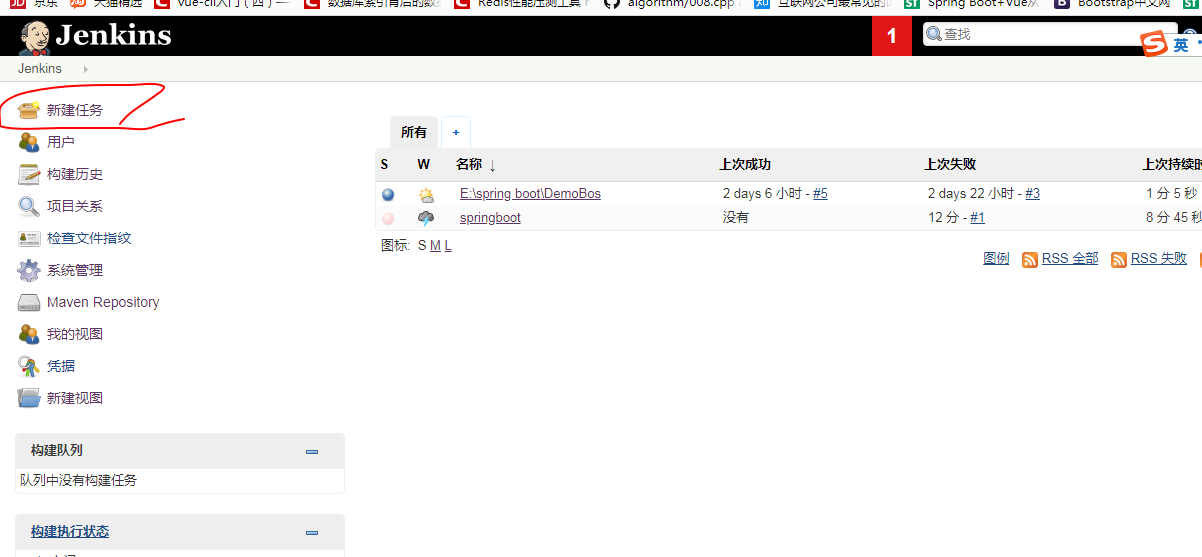




图片上写错了，应该是上传成功。

### Jenkins构建项目

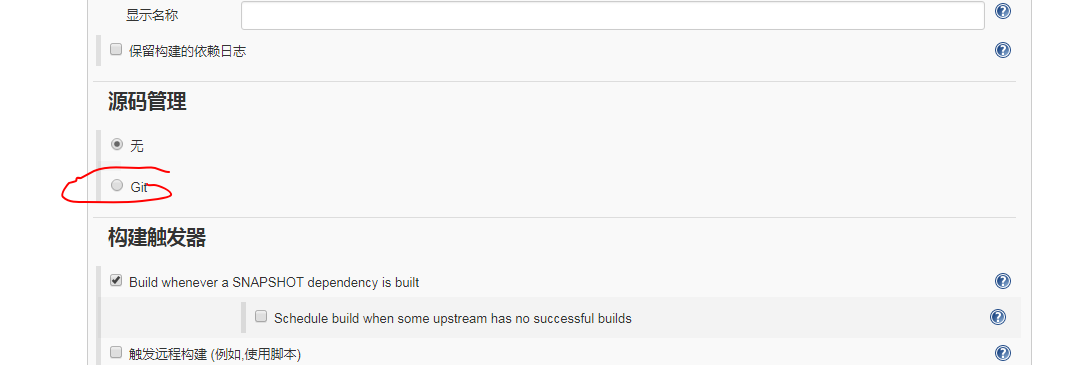
1. 登录自己的账号
2. 主页面，开始新建任务

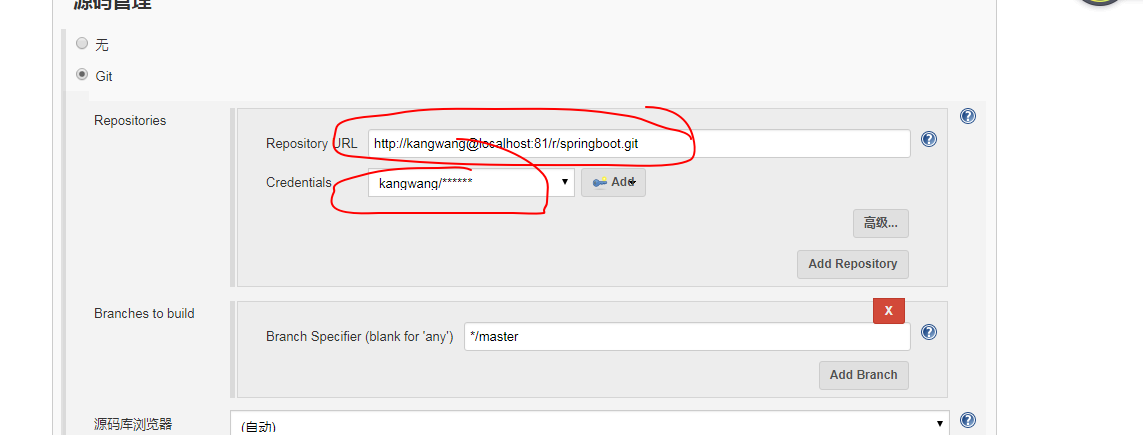


1. 填写名称，选择maven

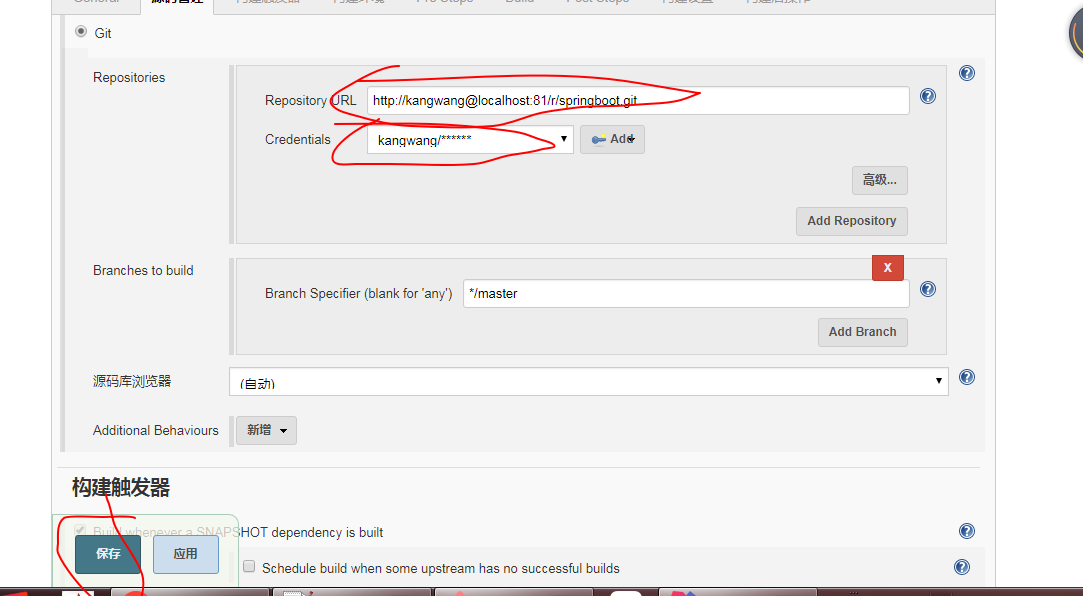


1. 使用仓库的代码





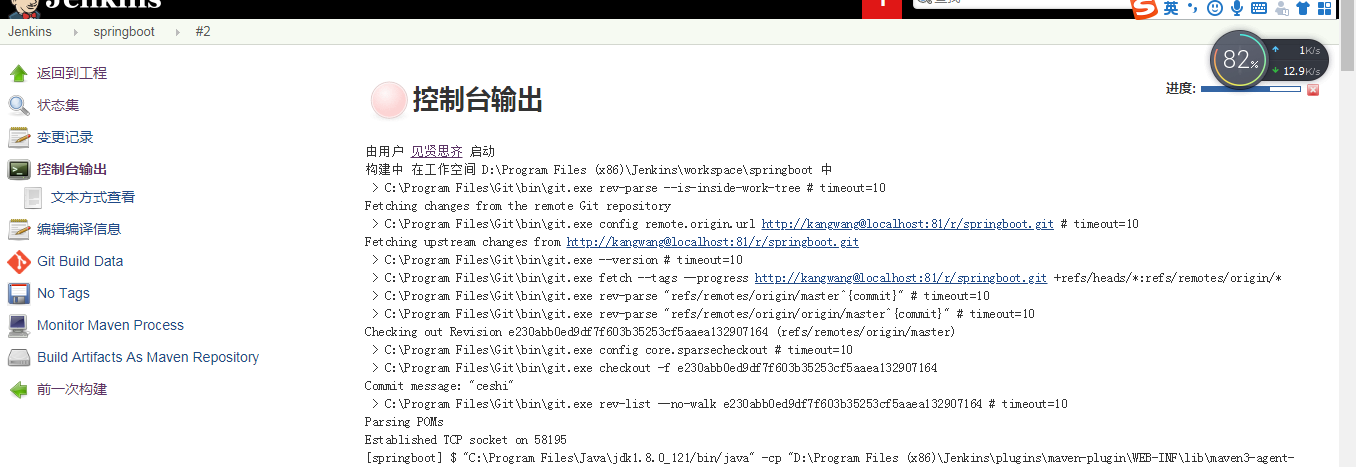
1. 设置仓库



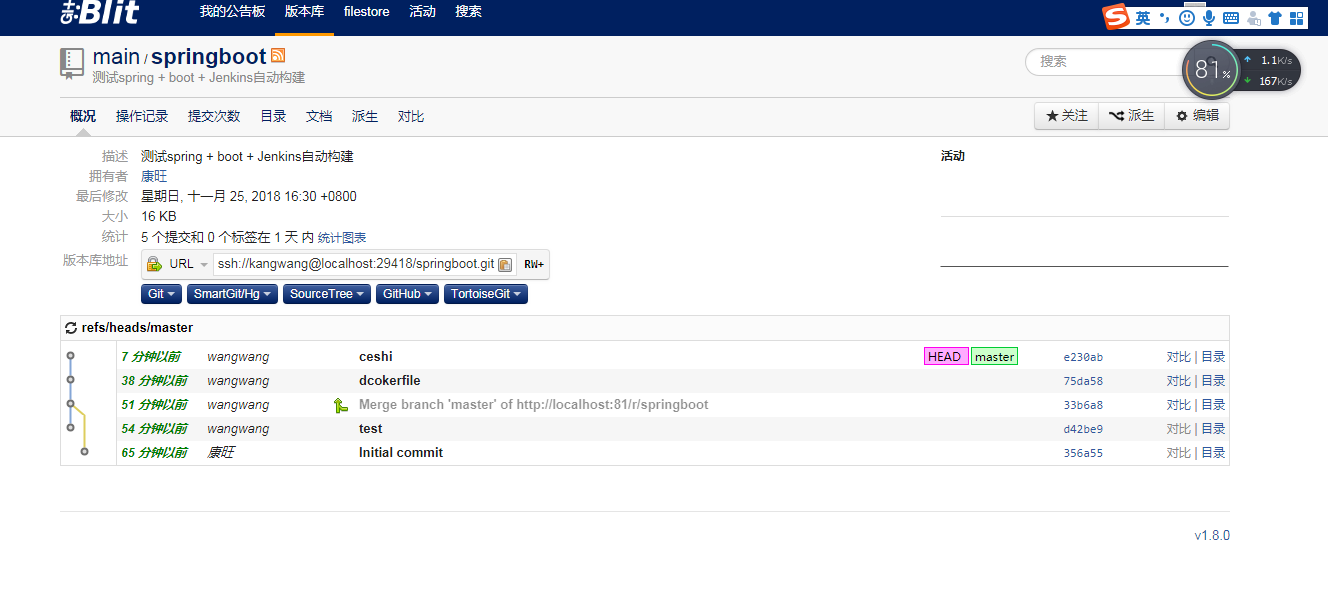
1. 点击立即构建



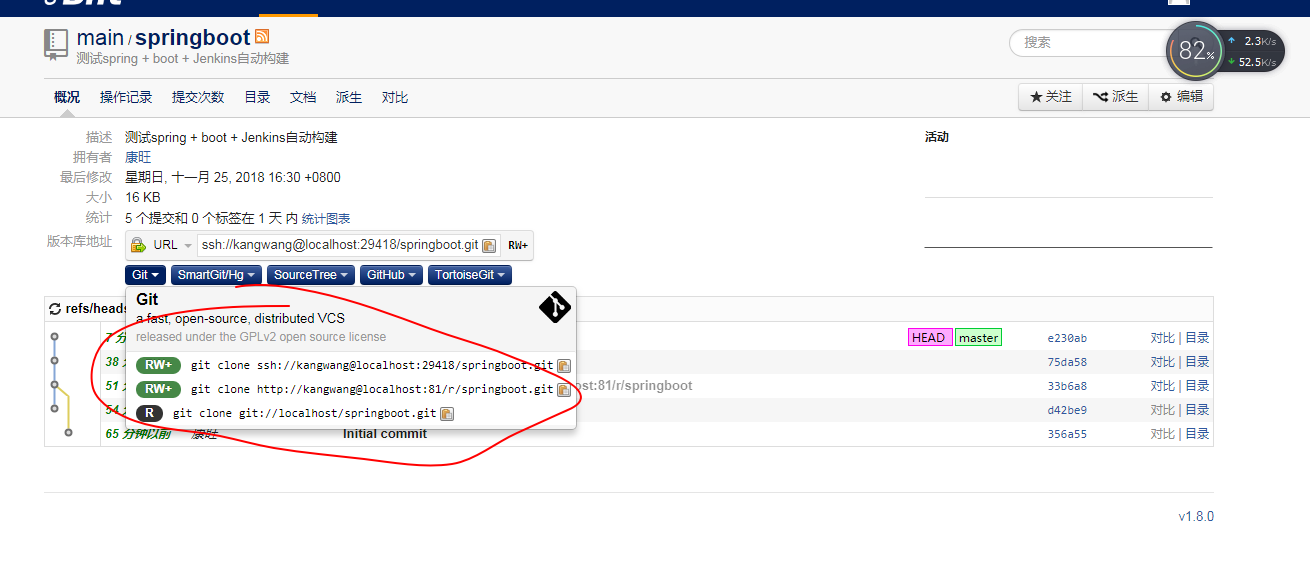
1. 控制台信息



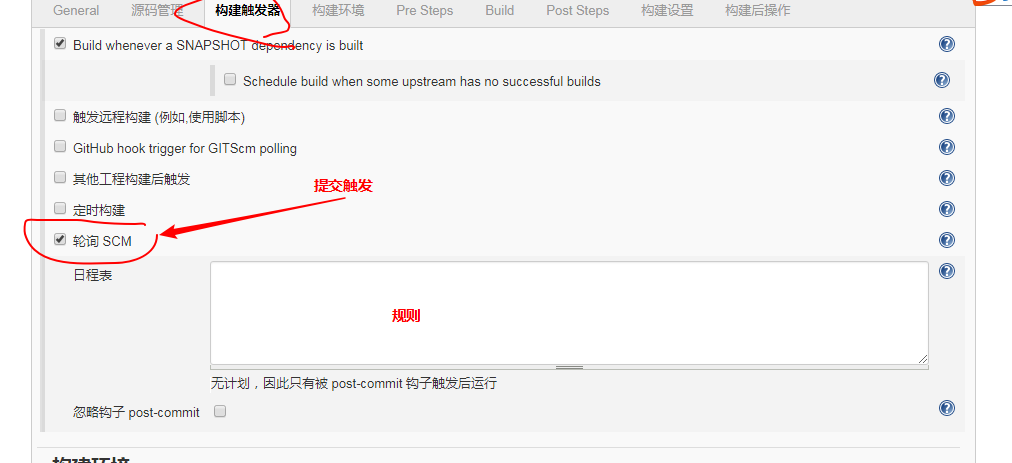
1. 仓库页面



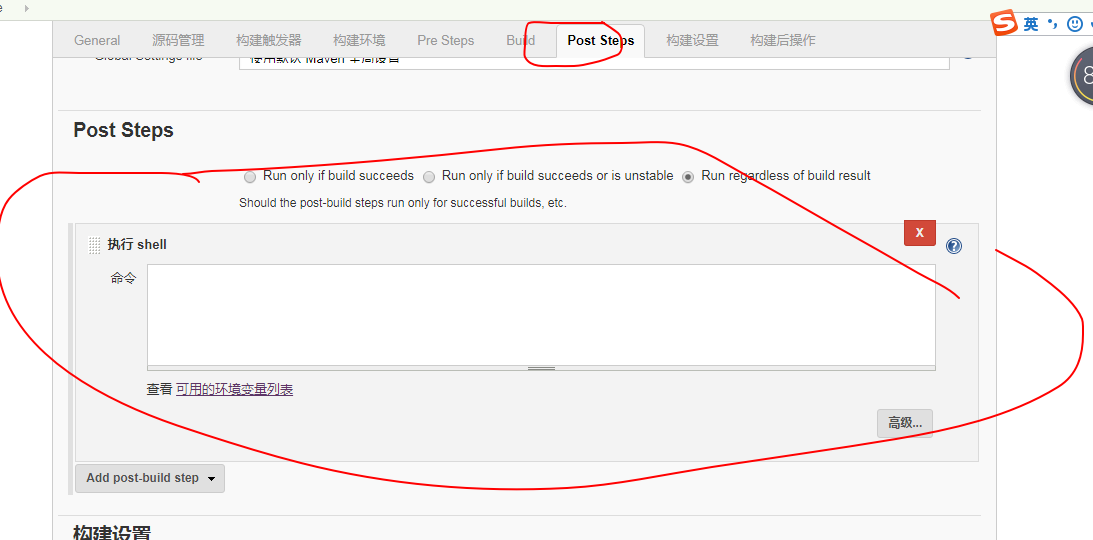
1. 仓库的地址



1. 自动触发，比如上传代码就触发



1. 构建之后执行



### 仓库初始化

1.初始化项目

git init

2.添加

git add .

3.增加源

git remote add origin http://kangwang@localhost:81/r/springboot.git

4.将远程的拉取

git pull origin master

5.将本地的上传

git push -u origin master

Docker安装建议去阿里云，上面很详细，可以将自己的镜像上传方便以后使用。