**一、Jenkins 部署**

**1.1 部署**

#wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo --no-check-certificate

#rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

yum install -y jenkins

service jenkins start

docker部署

|  |
| --- |
| docker run \  -u root \  --name=jenkins \  -d \  -p 8080:8080 \  -p 50000:50000 \  -v jenkins-data:/var/jenkins\_home \  -v /etc/localtime:/etc/localtime:ro \  -v /var/run/docker.sock:/var/run/docker.sock \  --restart=always \  jenkinsci/blueocean |

<https://pkg.jenkins.io/redhat-stable/>

**启动报错**

Starting Jenkins bash: /usr/bin/java: 没有那个文件或目录

[失败]

**用软链接解决**

ln -s /usr/java/jdk1.8.0\_144/bin/java /usr/bin/java

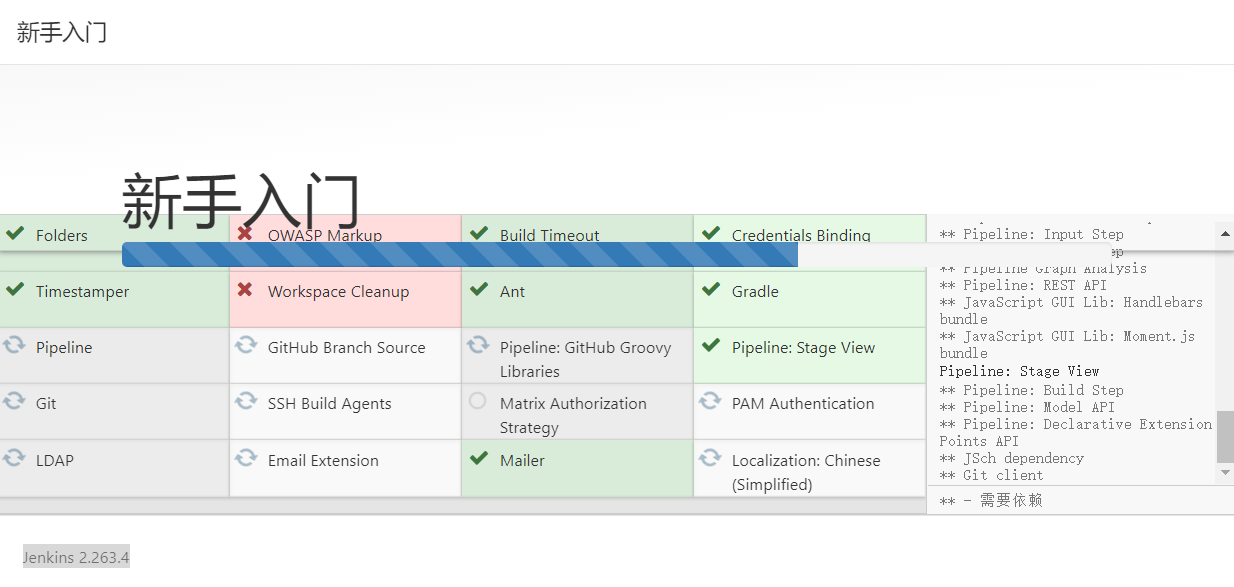
停止/启动  
service jenkins start/stop/restart  
chkconfig jenkins on

在浏览器中访问  
首次进入会要求输入初始密码如下图，  
初始密码在：/var/lib/jenkins/secrets/initialAdminPassword

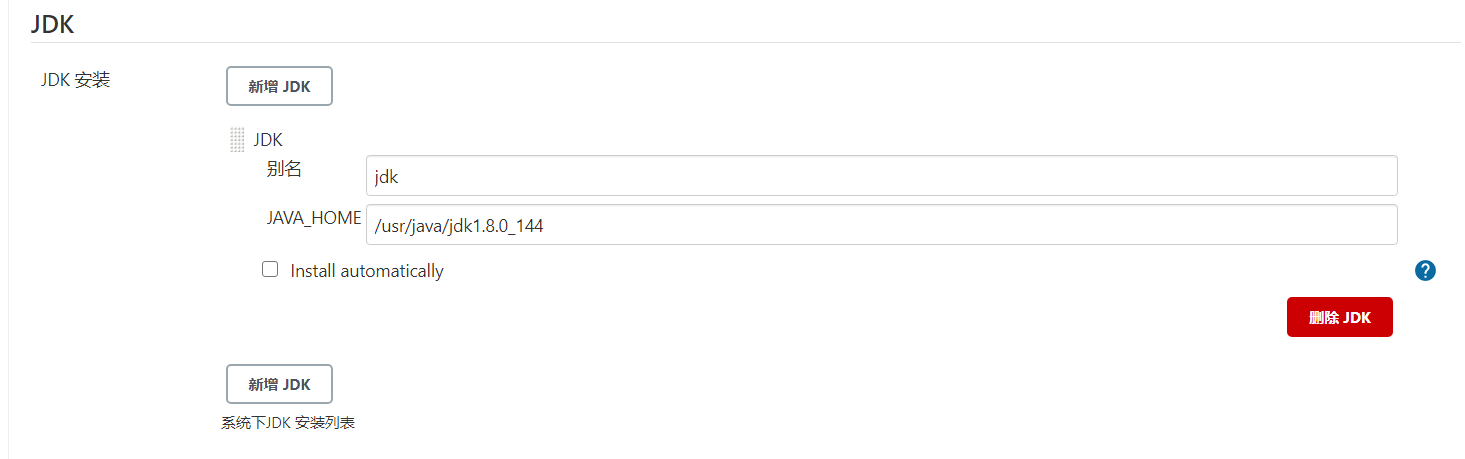


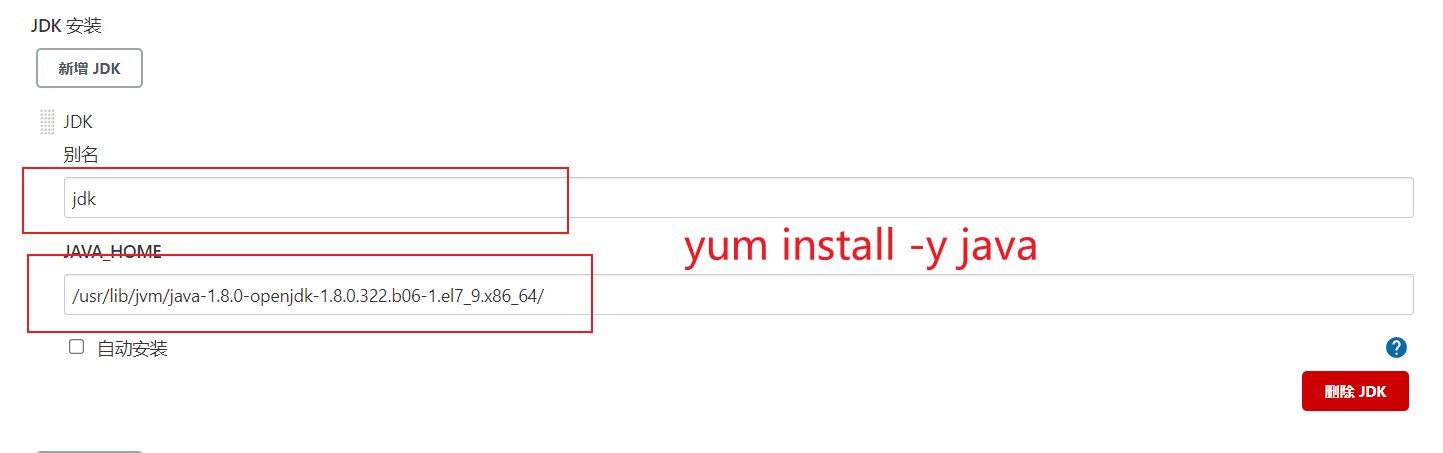


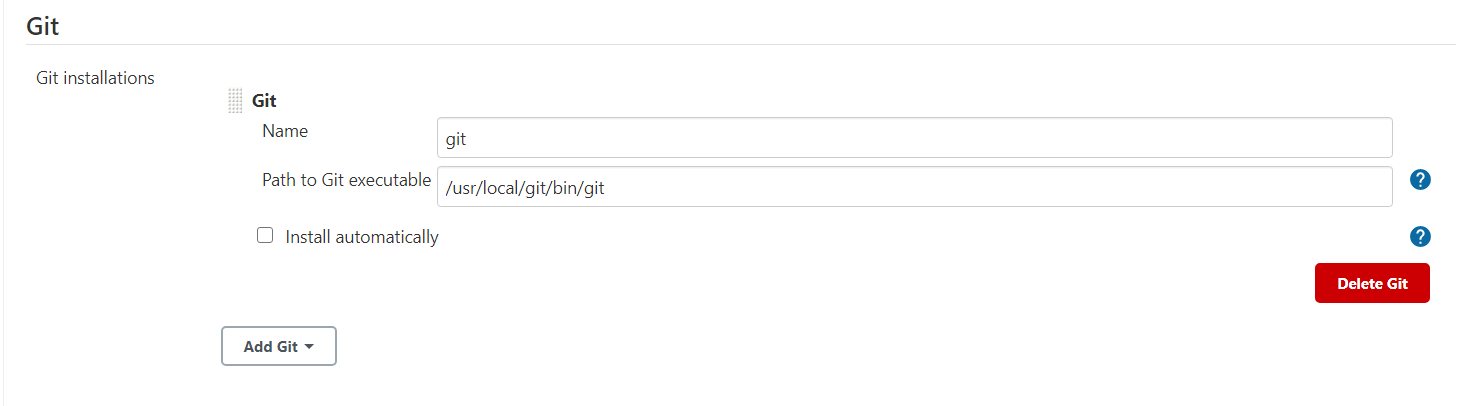
选择“Install suggested plugins”安装默认的插件，下面Jenkins就会自己去  
下载相关的插件进行安装。

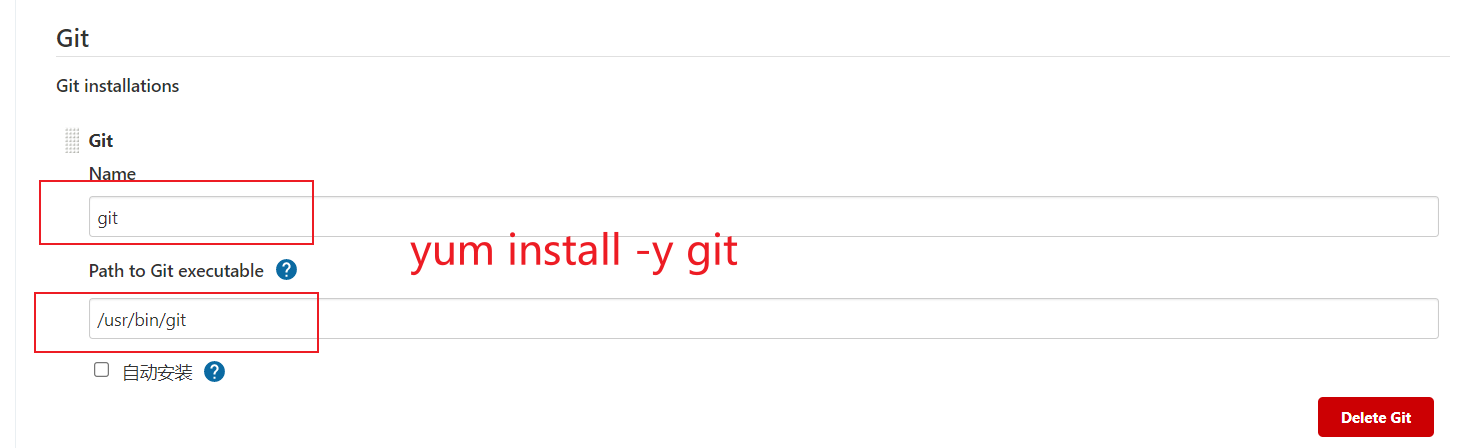




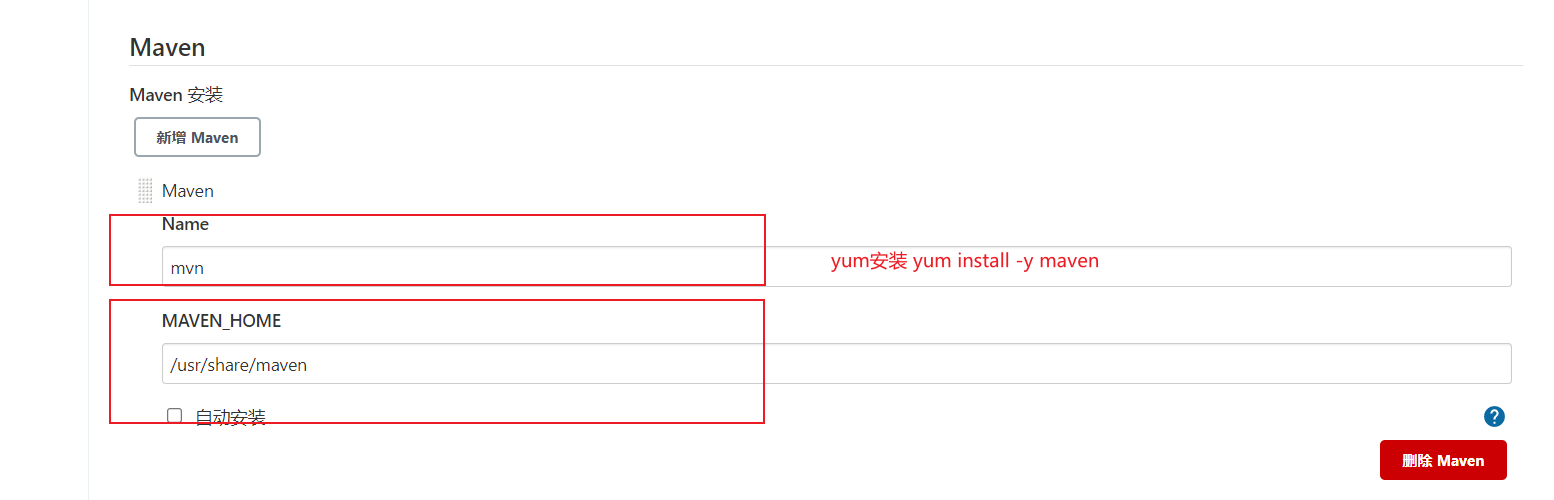


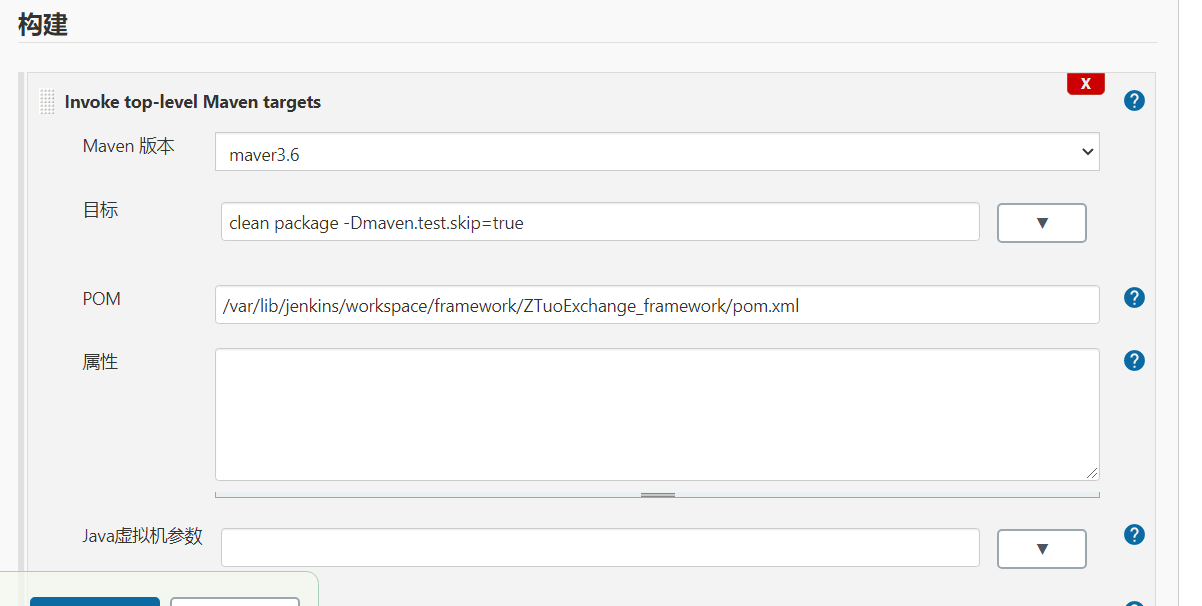












**Maven安装本地jar包**

mvn install:install-file -DgroupId=com.spark.bc -DartifactId=bitcoin-rpc -Dversion=1.2.0 -Dpackaging=jar -Dfile=/tmp/bitcoin-rpc-1.2.0.jar

|  |
| --- |
| **step1:安装依赖** yum -y install libcurl-devel expat-devel curl-devel gettext-devel openssl-devel zlib-devel gcc perl-ExtUtils-MakeMaker **step2:下载最新版**[git](https://github.com/git/git/releases) wget <https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.25.0.tar.gz>  tar xf git-2.25.0.tar.gz  cd git-2.25.0 **Step3:编译&安装** make prefix=/usr/local/git all  make prefix=/usr/local/git install **Step4:配置环境变量** echo 'export GIT\_HOME=/usr/local/git' >> /etc/profile  echo 'export PATH=$GIT\_HOME/bin:$PATH' >> /etc/profile  source /etc/profile  git --version |

**Maven安装**

|  |
| --- |
| wget <https://mirrors.sonic.net/apache/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.zip>  unzip apache-maven-3.6.3-bin.zip  mv apache-maven-3.6.3 /usr/local/  --------  vim /etc/profile  export MAVEN\_HOME=/usr/local/apache-maven-3.6.3  export PARH=$PATH:$MAVEN\_HOME/bin:$PATH  --------  mvn -v |

**1.2安装插件**

AnsiColor #构建输出颜色

Build With Parameters #参数化构建

Pipeline #pipeline

Configuration Slicing Plugin #批量修改job

Role-based Authorization Strategy#用户权限

Email Extension Plugin #发邮件的

Git Parameter Plug-In #git参数化构建

ThinBackup #jenkins备份用的

Job Config History #记录任务作业和系统配置的全部变更有历史的插件

Audit Trail #跟踪用户行为插件，相当于用户的操作日志

BuildResultTrigger Plugin #根据其他的job的成功或失败来启动此build

Parameterized Trigger Plugin #这是一个扩展型的插件，使各个job连接的时候可以传递一些job相关的信息

Maven Integration

Build Name and Description Setter #这个插件支持设置构建的显示名字，而不是默认的为#1

Localization: Chinese (Simplified)

Publish Over SSHe

Git Parameter

Persistent Parametere

Extended Choice Parameter

Blue Ocean Pipeline Editor （可选）

Blue Ocean （可选）

<https://www.cnblogs.com/honeybee/p/7877875.html>

<https://blog.csdn.net/pansaky/article/details/80755739>

**推荐安装的插件**

★Docker Pipeline && Docker

安装Docker Pipeline会自动安装docker相关的

这个允许我们自定义agent使用docker环境

★Git Parameter

解析git参数，允许我们选择分支进行构建

★Active Choices

可以做到参数的级联选择

★Generic Webhook Trigger

通用的webhook触发器，构建更强大的webhook功能

★Role-based Authorization Strategy

RBAC权限指定，给一个用户精确指定权限

★List Git Branches Parameter

列出分支参数

★Build With Parameters

基于自定义参数构建

Url <https://blog.csdn.net/weixin_44729138/article/details/108342538>

**二、使用技巧**

### **2.1 jenkins部署任务提示“Host key verification failed.”**

<https://blog.csdn.net/weixin_43557605/article/details/103804426>

<https://blog.csdn.net/gongmeiyan/article/details/104046497>

### **2.2 Linux（Centos7）安装Maven仓库（配置本地仓库、配置阿里镜像源）**

<https://blog.csdn.net/hu1010037197/article/details/110389389>

**2.3 NGINX 代理Jenkins**

|  |
| --- |
| server {  listen 80;  server\_name jenkins.wetm.top; # server\_name localhost;  access\_log /data/nginx/logs/jenkins/access.log main;  charset utf-8;  location / {  proxy\_pass http://192.168.91.11:8088;#设置主机头和客户端真实地址，以便服务器获取客户端真实IP  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-Ip $remote\_addr;  proxy\_set\_header X-Forwarded-For $remote\_addr;  }  } |

**2.4 插件加速**

#cd /var/lib/jenkins/updates && cp -a default.json default.json.bak

#sed -i 's#https://updates.jenkins.io/download#https://mirrors.tuna.tsinghua.edu.cn/jenkins#g' default.json

#sed -i 's#http://www.google.com#https://www.baidu.com#g' default.json

#service jenkins restart

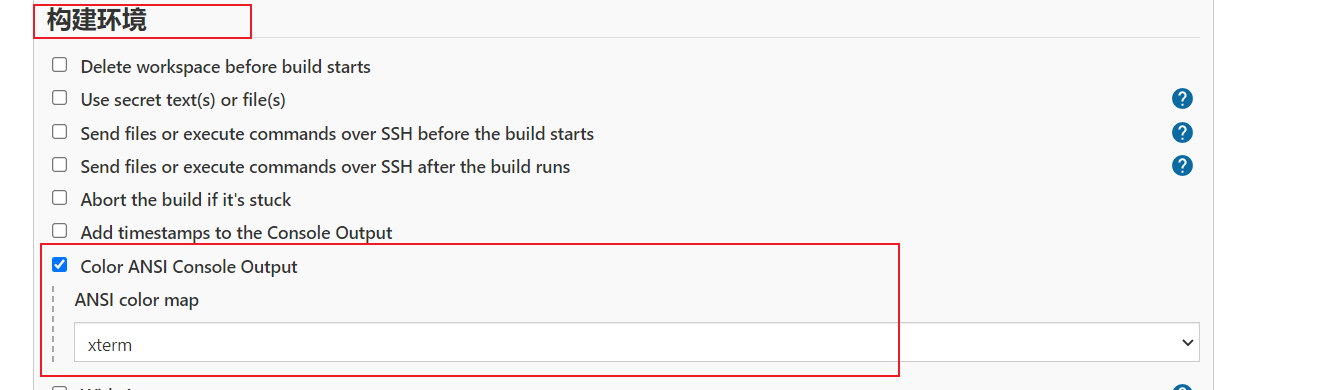
<https://www.cnblogs.com/hellxz/p/jenkins_install_plugins_faster.html>

<https://blog.csdn.net/JikeStardy/article/details/105606150>

### **2.5 Jenkins 控制台输出显示颜色**

jenkins 安装 AnsiColor 插件

jenkins输出变色插件:AnsiColor(在系统管理--->插件管理---->可选插件 搜索AnsiColor下载安装即可)

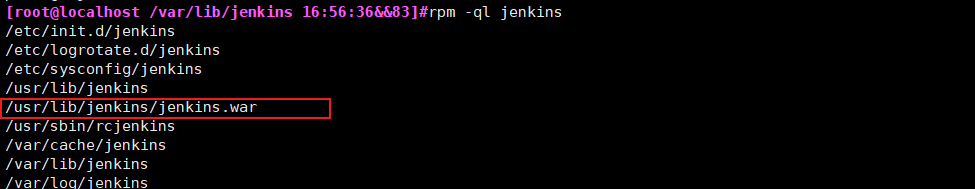


**2.6 jenkins升级版本**

**<https://mirrors.tuna.tsinghua.edu.cn/jenkins/war-stable>**

使用替换jenkins.war的方法使用最新版本

#rpm -ql jenkins



#cd /usr/lib/jenkins && mv jenkins.war jenkins.war.bak

#wget https://mirrors.tuna.tsinghua.edu.cn/jenkins/war-stable/2.332.2/jenkins.war --no-check-certificate

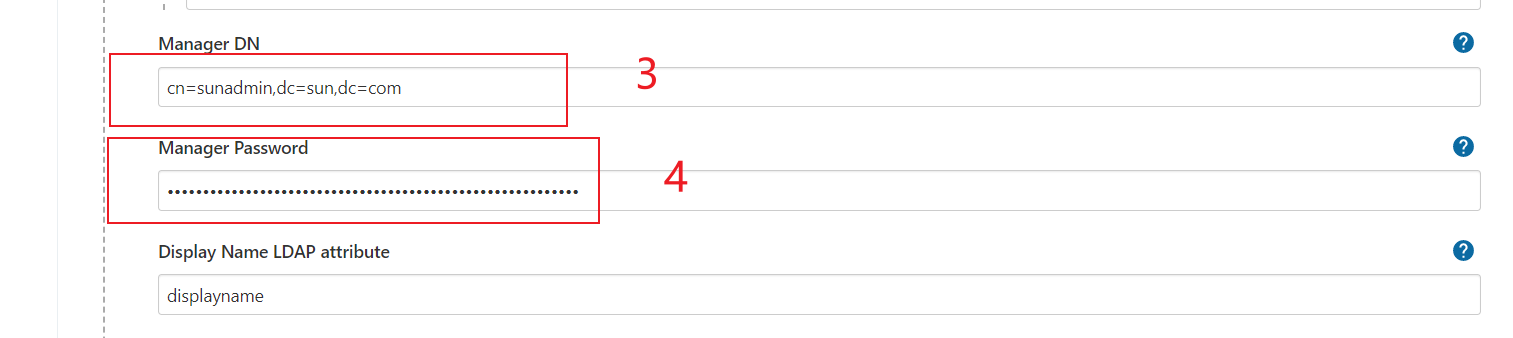
#service jenkins restart

**http://t.zoukankan.com/shengulong-p-9978763.html**

**2.6 Jenkins集成ldap**

**先备份 cp /var/lib/jenkins/config.xml{,.bak}**





主要修改以上四个配置，保存后。原先不是ldap的用户就会无法登陆。

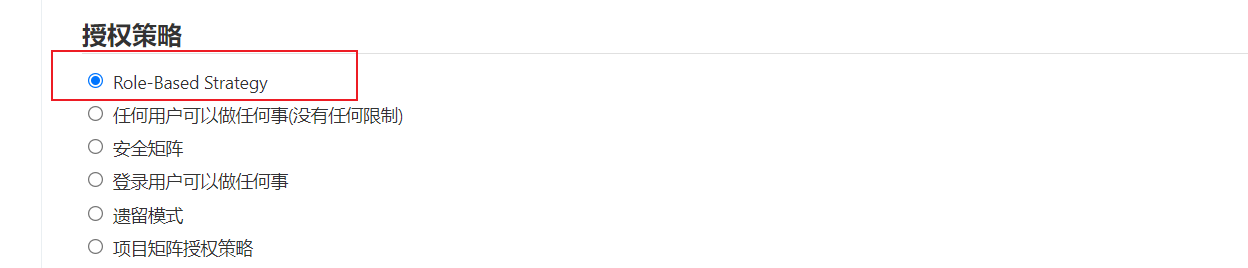
没有配置User search base和Group search base，这样ldap用户登录时，会对ldap 服务器上所有组所有用户进行匹配查询，如果指定了某一个组或某一个用户，那么其他组其他用户就算在ldap服务器上存在，也无法通过认证。LDAP只是做认证的，jenkins的的权限需要在授权策略中配置。

<https://blog.51cto.com/zengestudy/1911358>

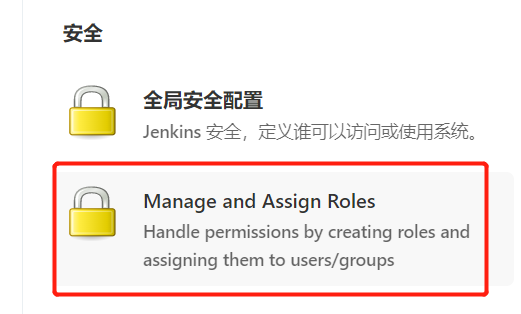
2.6.1 **配置ldap的账户权限**

安装Role-based Authorization Strategy插件

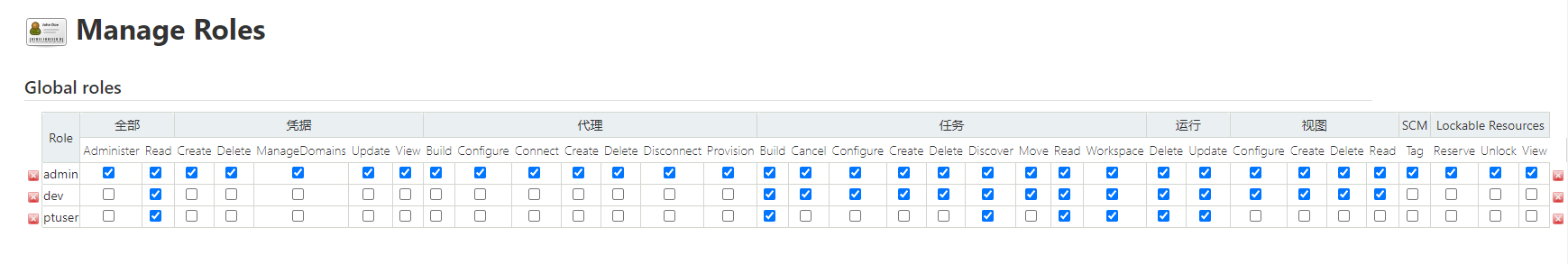
在系统管理——>全局安全配置,可以看到下面选项

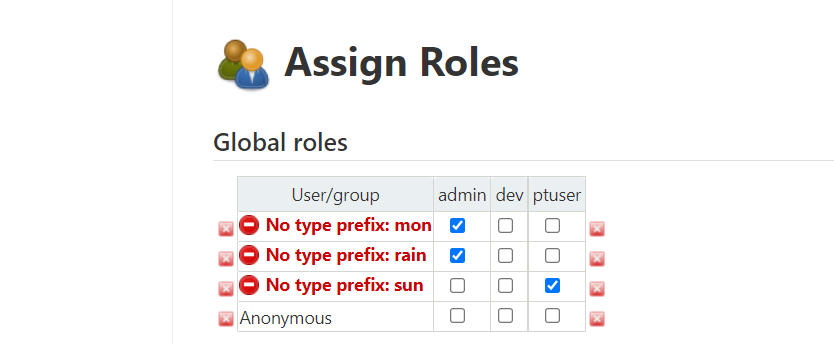


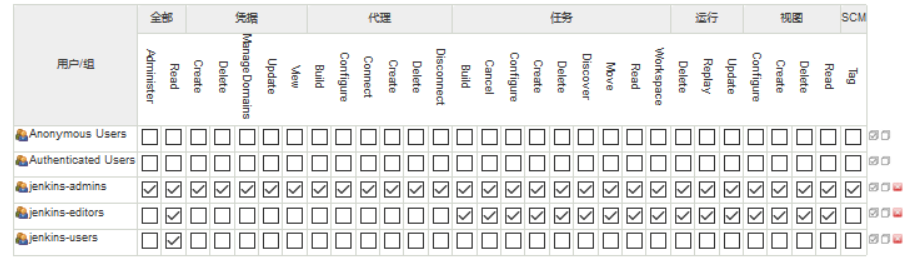
保存以后，返回系统管理界面就可以看到多处一个Manage and Assign Roles



* ****Manage Roles****: 角色分为Global和Project，可创建角色分组和添加项目。
* ****Assign Roles****: 增加具体的用户，分配到角色组，指定项目权限。







<https://www.xxlaila.cn/2019/10/14/jenkins%E9%85%8D%E7%BD%AEldap/>

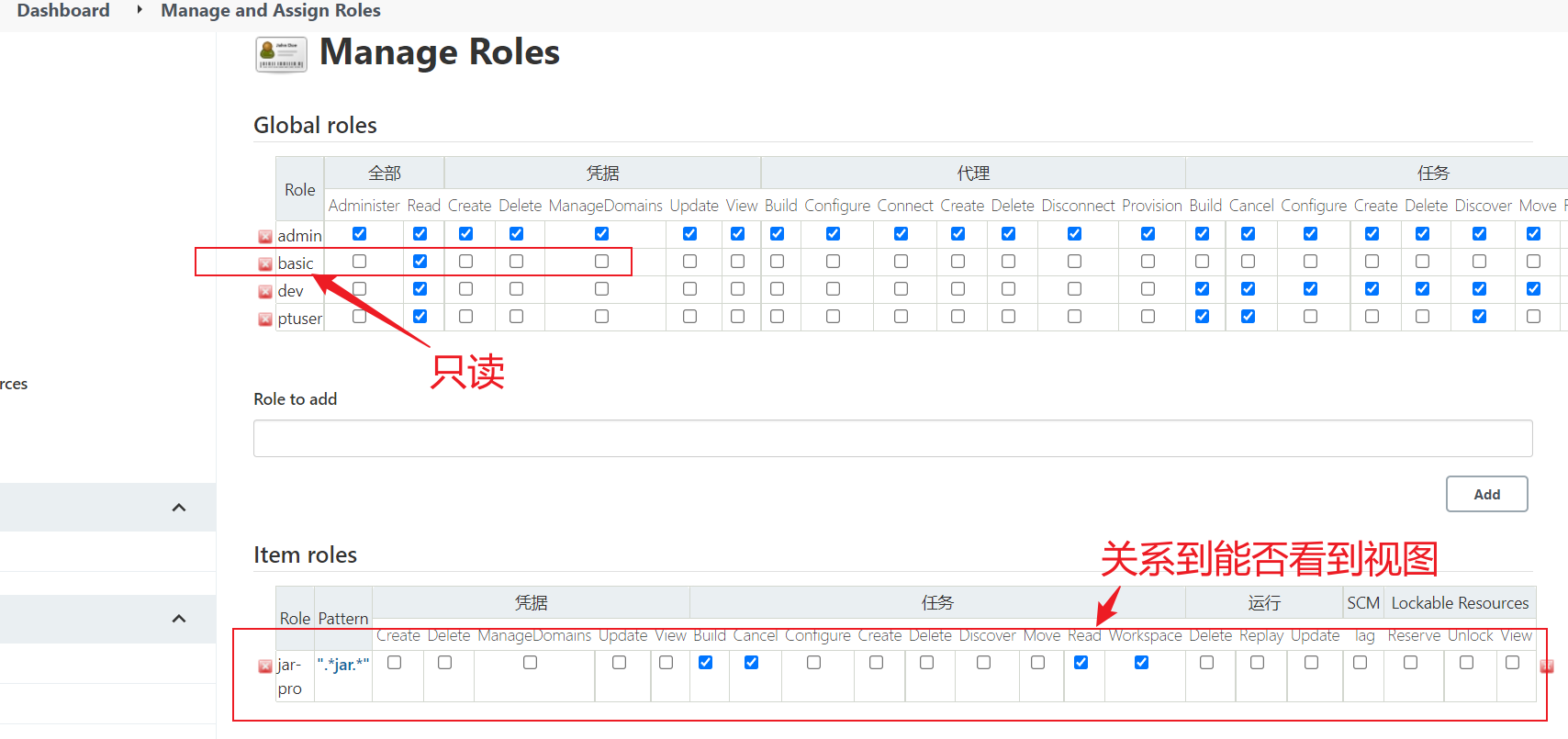
<https://www.cnblogs.com/zhaojiedi1992/p/zhaojiedi_liunx_52_ldap_for_jenkins.html>

<https://blog.csdn.net/wangshui898/article/details/108527378>

<https://blog.csdn.net/pcn01/article/details/105442954>

**Global roles 与 Item roles 配合使用**

其中overall中的read要勾选，否则用户登陆后什么也看不到

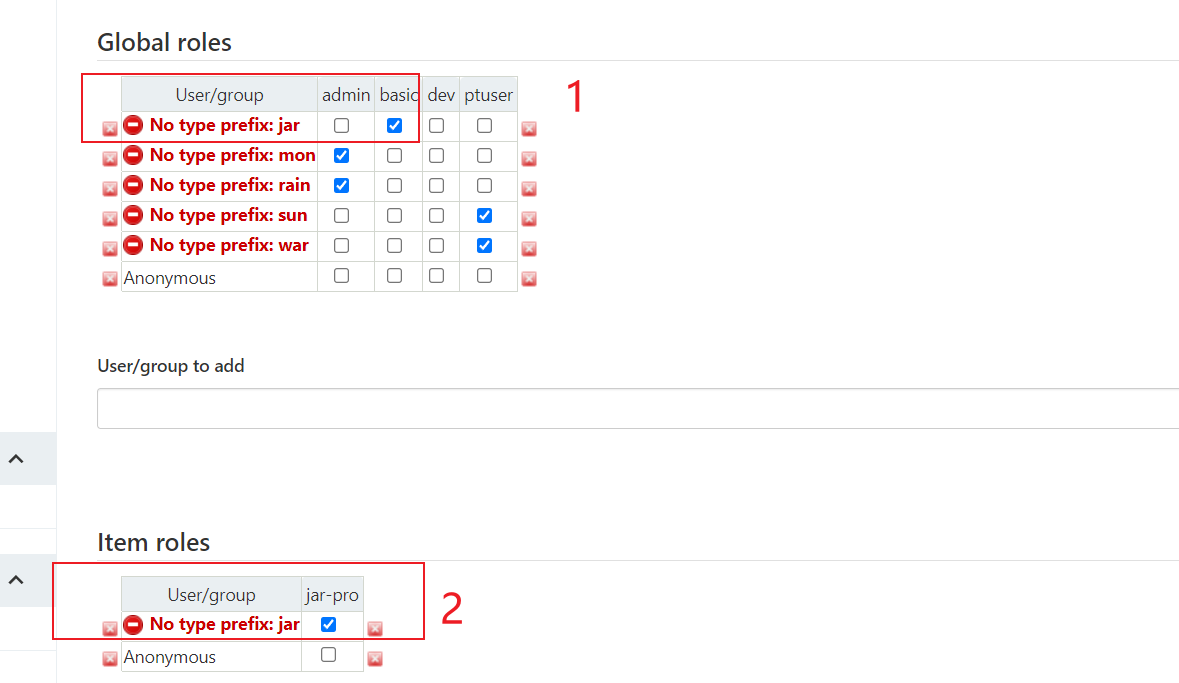


**匹配包含jar的视图任务**

**输入** **.\*jar.\***



接下来权限分配给具体用户，本例是jar用户



登录查看验证，jar用户只有以下两个视图任务的权限



<https://www.cnblogs.com/g2thend/p/12403282.html>

**2.7Jenkins显示语言切换为中文(**

2.8 [jenkins集成ldap后管理员密码登录不了](https://www.cnblogs.com/amoyzhu/p/6693097.html)



|  |
| --- |
| 1.密码管理员密码，如何修改  进入/var/lib/jenkins目录下修改config.xml文件：  以下密码是admin  <hudson.security.HudsonPrivateSecurityRealm\_-Details> <passwordHash>#jbcrypt:$2a$10$oZgKQuNVqiMglf2lqPm2WOfXqZoq5gWt4NvrT1iugjCNEtVDsUP1G</passwordHash> </hudson.security.HudsonPrivateSecurityRealm\_-Details>  2.配置了LDAP导致管理员密码登录不了。  进入/var/lib/jenkins目录下修改config.xml文件  <securityRealm class="hudson.security.LDAPSecurityRealm" plugin="ldap@1.26">  <disableMailAddressResolver>false</disableMailAddressResolver>  <configurations>  <jenkins.security.plugins.ldap.LDAPConfiguration>  <server>192.168.91.11:389</server>  <rootDN>dc=sun,dc=com</rootDN>  <inhibitInferRootDN>false</inhibitInferRootDN>  <userSearchBase></userSearchBase>  <userSearch>uid={0}</userSearch>  <groupMembershipStrategy class="jenkins.security.plugins.ldap.FromGroupSearchLDAPGroupMembershipStrategy">  <filter></filter>  </groupMembershipStrategy>  <managerDN>cn=sunadmin,dc=sun,dc=com</managerDN>  <managerPasswordSecret>{AQAAABAAAAAQM3jlS1iwFmvJnGww15uapycyx7FNch6LpftkzGb6U8U=}</managerPasswordSecret>  <displayNameAttributeName>displayname</displayNameAttributeName>  <mailAddressAttributeName>mail</mailAddressAttributeName>  <ignoreIfUnavailable>false</ignoreIfUnavailable>  </jenkins.security.plugins.ldap.LDAPConfiguration>  </configurations>  <userIdStrategy class="jenkins.model.IdStrategy$CaseInsensitive"/>  <groupIdStrategy class="jenkins.model.IdStrategy$CaseInsensitive"/>  <disableRolePrefixing>true</disableRolePrefixing>  </securityRealm>  改成：  <securityRealm class="hudson.security.HudsonPrivateSecurityRealm">  <disableSignup>true</disableSignup>  <enableCaptcha>false</enableCaptcha>  </securityRealm> |

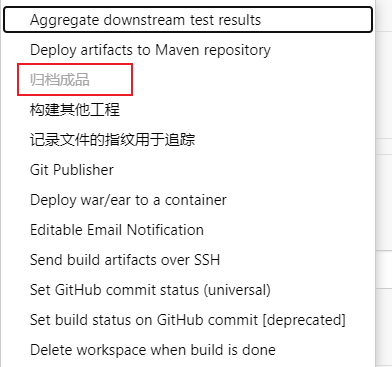
<https://blog.csdn.net/weixin_34110749/article/details/92949695>

2.9 [jenkins构建后操作archive the artfacts-归档成品的用法](https://www.cnblogs.com/kaerxifa/p/11142470.html)

**注意了，archive the artifacts的相对路径是：WORKSPACE中当前job的工作目录**

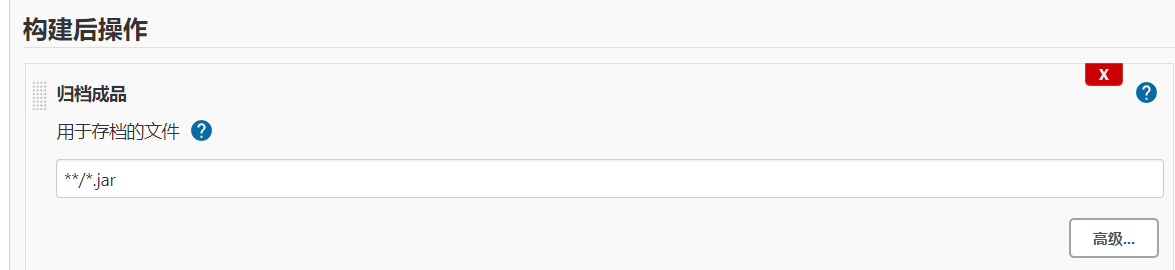
**它会去job所在的目录去匹配文件**

打开项目配置-> 构建后操作 -> Archive the artifacts（中文名：归档成品）



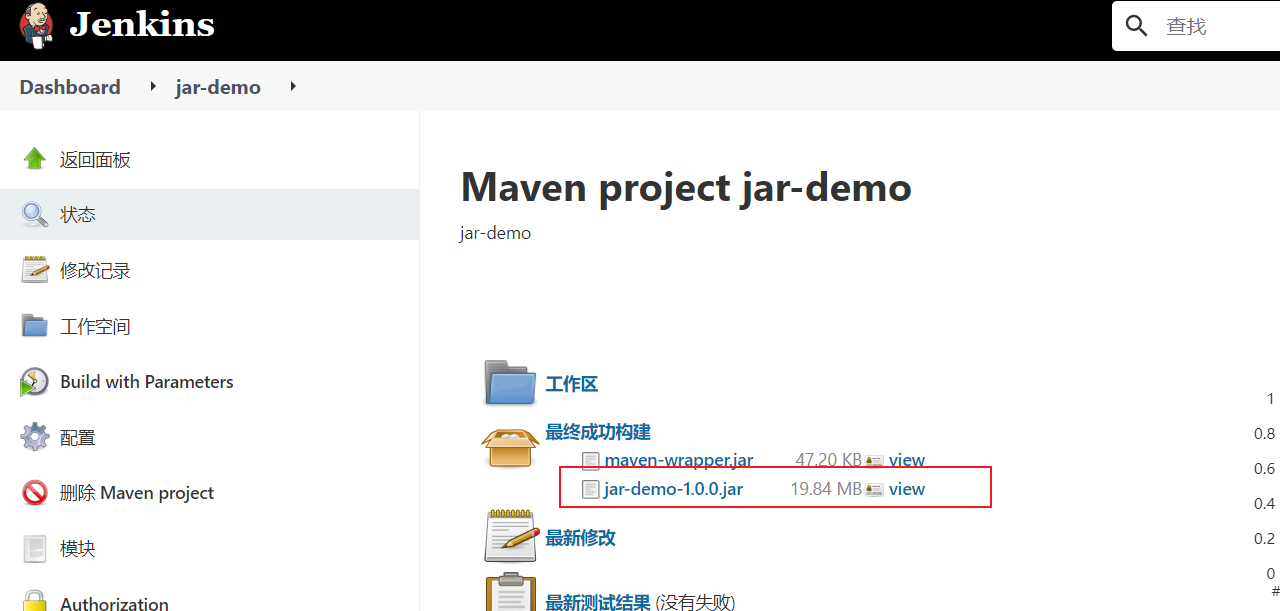
在存档文件中输入需要存档的文件，多个文件以 “,“(逗号) 分割，**存档文件默认路径为 WORKSPACE中当前job的工作目录 ,存档文件的相对路径既是 WORKSPACE 并且是以正则表达式路径、文件名去匹配需要存档的文件**

构建的 apk 存放在 WORKSPACE目录 (workspace\jar-demo)，存档文件写为 \*\*/\*.jar



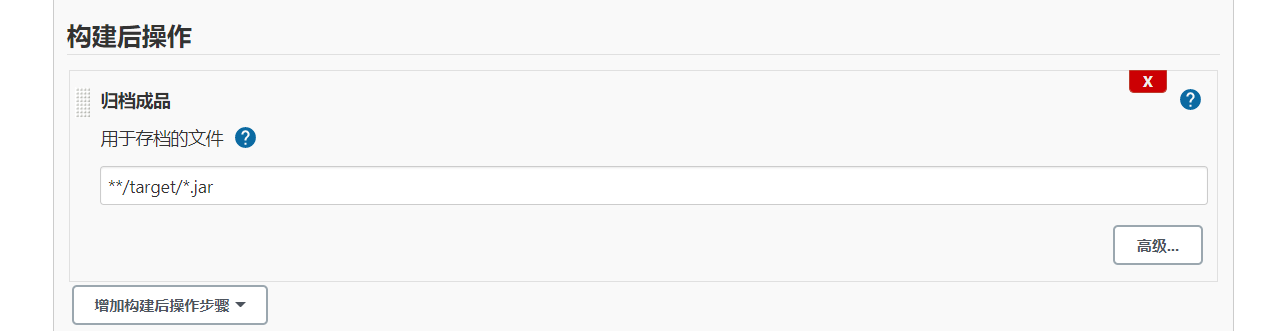
配置完成执行构建

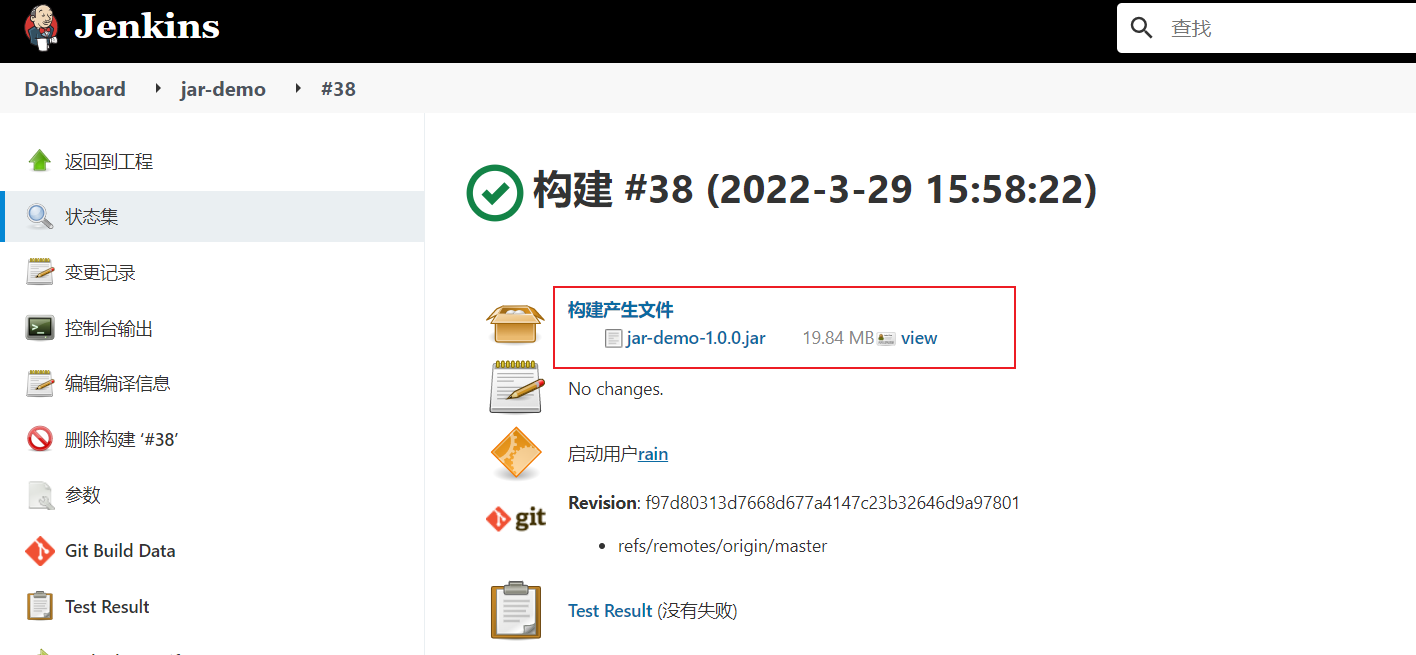
在工作区下面显示最后一次成功构建的结果



<https://www.cnblogs.com/kaerxifa/p/11142470.html>

**或者：范围缩小**





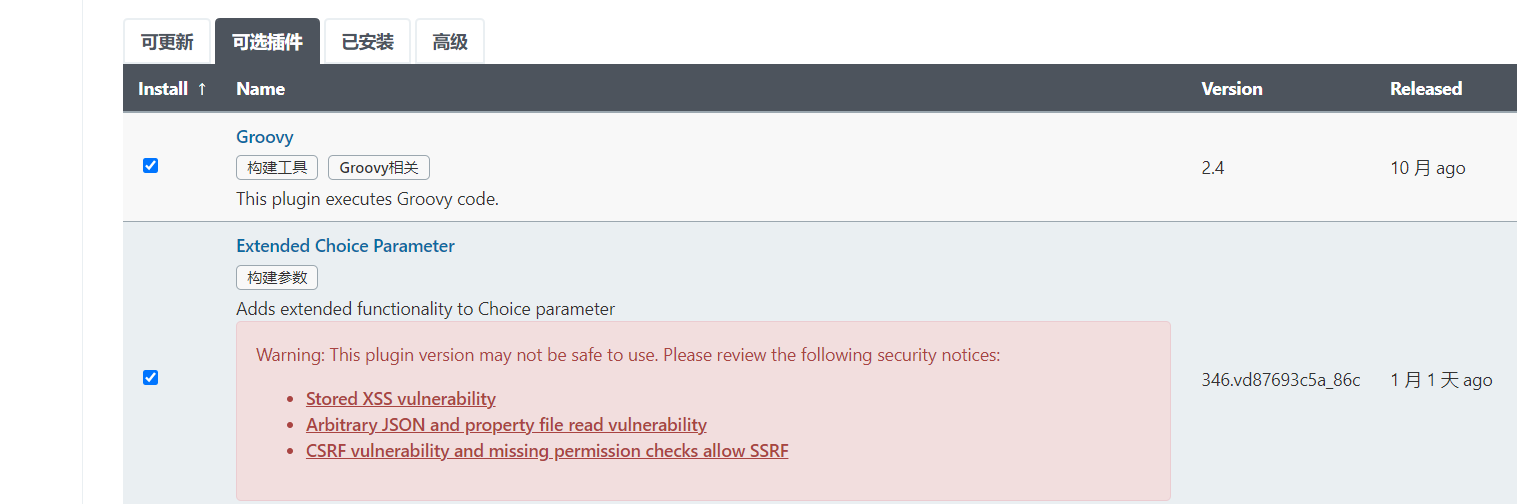
<https://blog.51cto.com/215687833/4776794>

2.10 ThinBackup插件备份

<http://www.mydlq.club/article/60/>

[https://www.cnblogs.com/operationhome/p/12240233.html](http://www.mydlq.club/article/60/)

[2.11 Groovy插件](http://www.mydlq.club/article/60/)

[](http://www.mydlq.club/article/60/)

[https://blog.csdn.net/qq\_36294284/article/details/103597156](http://www.mydlq.club/article/60/)

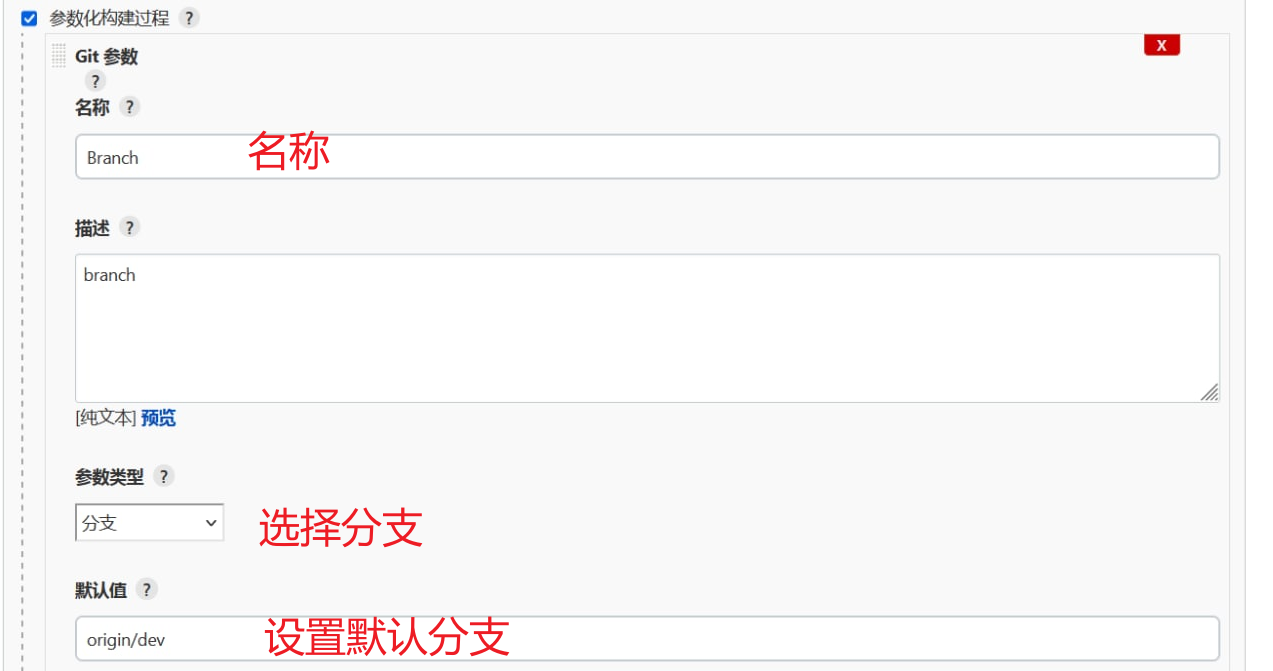
[return['sh /tmp/1.sh'.execute().text]](http://www.mydlq.club/article/60/)

[2.12 jenkins构建时支持git选择分支](http://www.mydlq.club/article/60/)

[1.安装](http://www.mydlq.club/article/60/)[Git Parameter](https://wiki.jenkins-ci.org/display/JENKINS/Git+Parameter+Plugin" \t "https://blog.csdn.net/u012076316/article/details/_blank)[插件](http://www.mydlq.club/article/60/)

[2.勾选“ 参数化构建过程 ”，然后添加](http://www.mydlq.club/article/60/)[Git Parameter](https://wiki.jenkins-ci.org/display/JENKINS/Git+Parameter+Plugin" \t "https://blog.csdn.net/u012076316/article/details/_blank)[,配置如下](http://www.mydlq.club/article/60/)

[](http://www.mydlq.club/article/60/)

[](http://www.mydlq.club/article/60/)

[](http://www.mydlq.club/article/60/)