Linux(CentOS)下安装jenkins

# 准备工作

## JDK的安装

#卸载系统默认的老版本jdk

rpm -qa | grep jdk

rpm -e java-1.5.0-gcj-1.5.0.0-29.1.el6.x86\_64

#安装jdk

rpm -ivh /usr/local/src/jdk-7u79-linux-x64.rpm

## Git的安装

yum install git –y

## Maven的安装

|  |
| --- |
| #解压maven  tar -zxvf /usr/local/src/apache-maven-3.3.3-bin.tar.gz -C /usr/local  mv /usr/local/apache-maven-3.3.3/ /usr/local/maven  #配置环境变量  cat >> /etc/profile <<EOF  export MAVEN\_HOME=/usr/local/maven  export PATH=\$PATH:\$MAVEN\_HOME/bin  EOF  source /etc/profile  #测试maven是否安装成功  mvn -version |

## Tomcat的安装

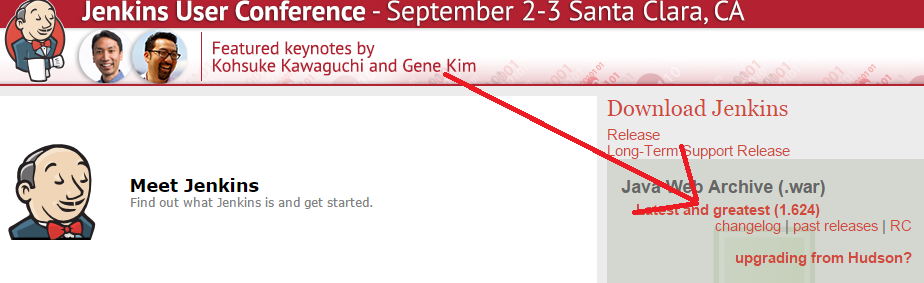
tar -zxvf apache-tomcat-7.0.62.tar.gz -C /usr/local

mv /usr/local/apache-tomcat-7.0.62/ /usr/local/tomcat7

## Jenkins下载和启动

第一步：下载jenkins

http://jenkins-ci.org/



第二步：将war包放入tomcat中，并启动tomcat。



此时jenkins正在初始化配置目录，默认的路径在当前用户目录下的.jenkins下。Jenkins默认是把数据读到内存中。如果你要更换之前的配置文件，你需要重启tomcat或者点击jenkins读取设置。/root/.jenkins主目录存放了jenkins所有数据文件。

## 安装jenkins插件

自动化部署所需全部插件：

Email Extension Plugin(邮件通知)、

GIT plugin(找Github Authentication plugin)、

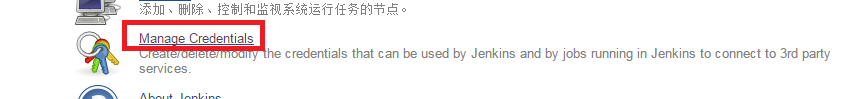
Publish Over SSH(远程Shell)、

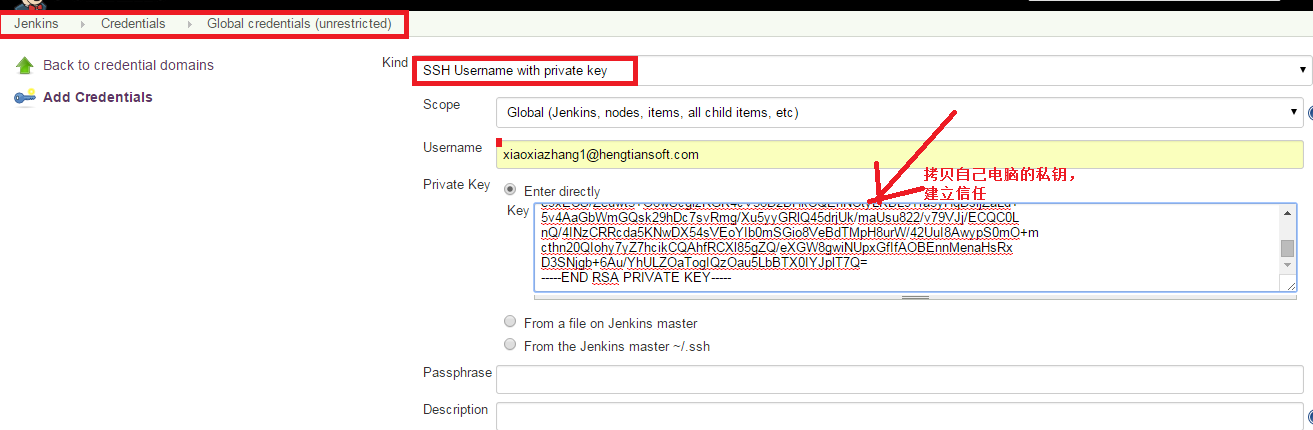
Deploy to container Plugin（远程发布插件）

安装方法：首页->系统管理->管理插件->可选插件->过滤(搜索插件名)->勾选->点击最下面直接安装即可(需要等待一段时间,详情可以看catalina.out日志变化)

# 持续集成部署

## Credentials管理



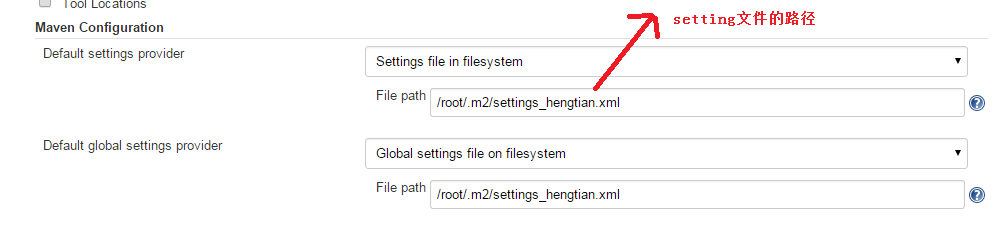


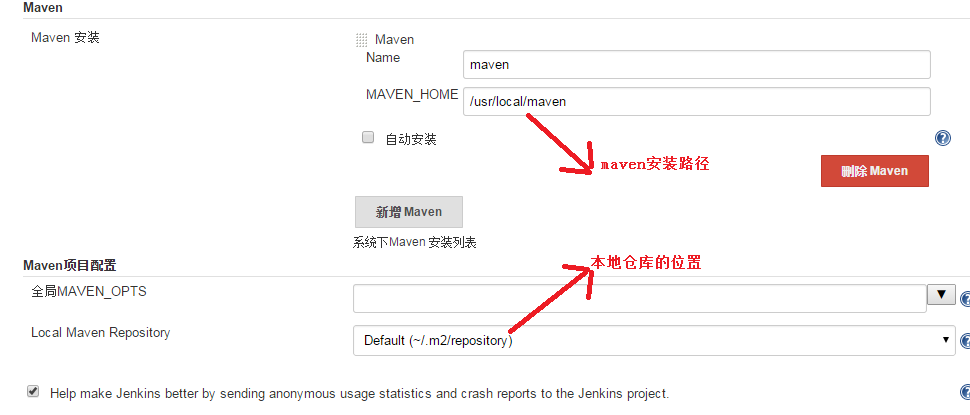
## 配置jenkins

### 全局属性



### Maven Configuration





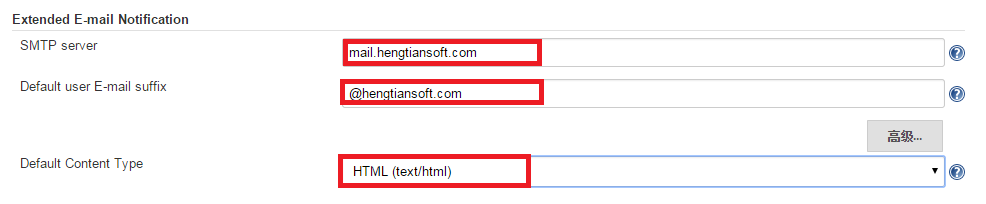
### Jdk配置



### Git的配置



### 邮件配置



Default Subject 代码：

|  |
| --- |
| $PROJECT\_NAME - Build # $BUILD\_NUMBER - $BUILD\_STATUS! |
| Default Recipients：自定义默认电子邮件收件人列表 |

Default Content：

|  |
| --- |
| (本邮件是程序自动下发的，请勿回复，<span style="color:red">请相关人员fix it,重新提交到git 构建</span>)<br/><hr/>  项目名称：$PROJECT\_NAME<br/><hr/>  构建编号：$BUILD\_NUMBER<br/><hr/>    GIT版本号：${GIT\_REVISION}<br/><hr/>    构建状态：$BUILD\_STATUS<br/><hr/>    触发原因：${CAUSE}<br/><hr/>    构建日志地址：<a href="${BUILD\_URL}console">${BUILD\_URL}console</a><br/><hr/>    构建地址：<a href="$BUILD\_URL">$BUILD\_URL</a><br/><hr/>    变更集:${JELLY\_SCRIPT,template="html"}<br/><hr/> |

### 配置 Publish over SSH

远程执行shell脚本 采用公钥私钥连接 其中Key里贴的是私钥 远程被管理的主机里贴的是公钥,这2台主机就是相互信任，这样scp等操作就不需要输入用户名和密码。

公共配置：

* Passphrase：密码（key的密码，如果你设置了）
* Path to key：key文件（私钥）的路径
* Key：将私钥复制到这个框中
* Disable exec：禁止运行命令

私有配置：

* SSH Server Name：标识的名字（随便你取什么）
* Hostname：需要连接ssh的主机名或ip地址（建议ip）
* Username：用户名
* Remote Directory：远程目录
* Use password authentication, or use a different key：可以替换公共配置（选中展开的就是公共配置的东西，这样做扩展性很好）

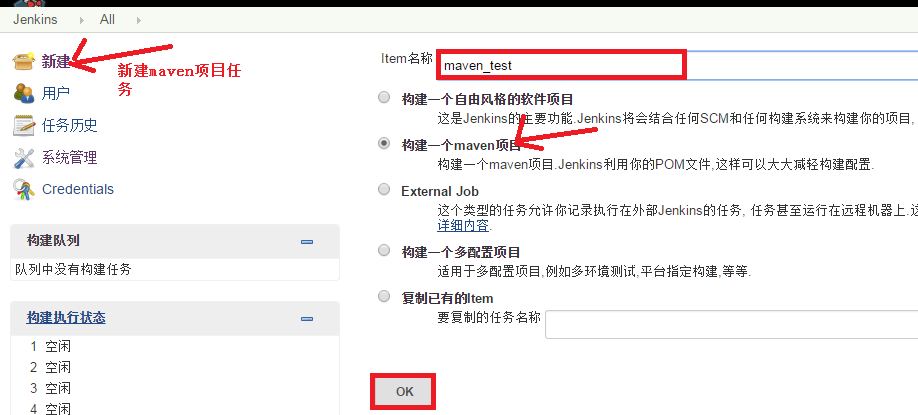
私有配置的高级：

* Port：端口（默认22）
* Timeout (ms)：超时时间（毫秒）默认即可
* Disable exec：禁止运行命令
* Test Configuration：测试连接

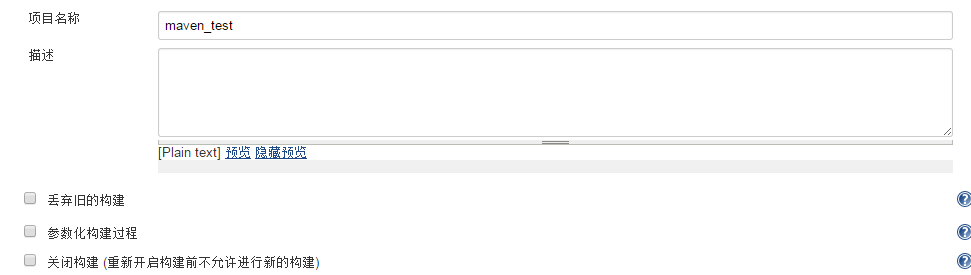
ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa #生成私钥

ssh-copy-id -i /root/.ssh/id\_rsa.pub [root@172.17.128.83#](mailto:root@172.17.128.83)传递公钥

## Job的配置

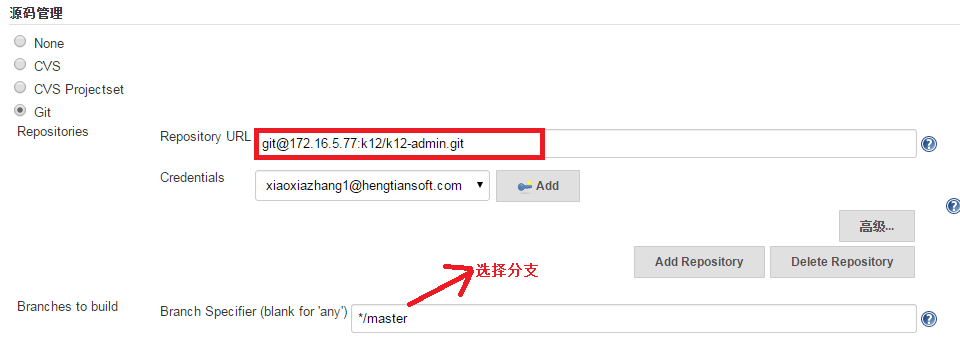


### Job基本信息



### 项目源码管理

源码管理的时候注意，要将本机的公钥拷贝到git服务器上，实现单向信任。



### 构建触发器



### 构建环境

其中在构建前和构建后可以指定执行的操作。常见的有Exceute shell和send files or execute commands over SSH



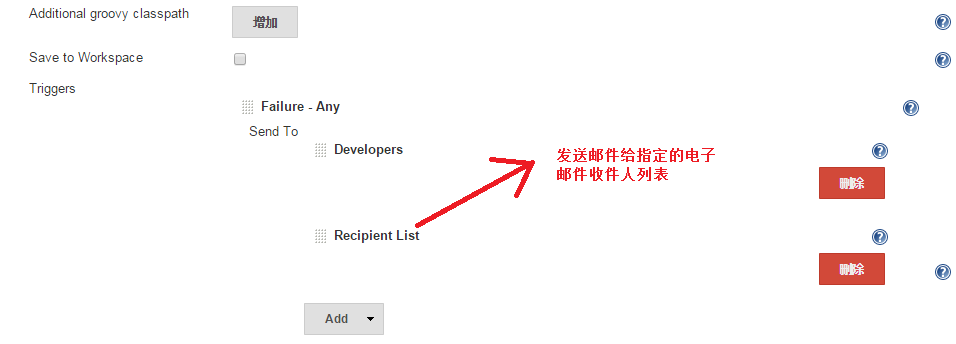
|  |
| --- |
| /usr/bin/mussh -m -h 172.16.129.247 -c "kill -9 $(ps -ef |grep java | grep -v "grep" | awk '{print $2}') ; rm -rf /usr/local/tomcat7/webapps/k12\*"  scp /root/.jenkins/jobs/test/workspace/target/k12-admin-1.0.0-SNAPSHOT.war root@172.16.129.247:/usr/local/tomcat7/webapps/k12-admin.war  /usr/bin/mussh -m -h 172.16.129.247 -c "/usr/local/tomcat7/bin/startup.sh" |

### 构建设置

#构建后操作，主要有Deploy war/ear to a container(用于部署tomcat)，#Send Build artifacts over SSH(用于上传部署文件)，

#Editable Email Notification(用于详细email 配置)

1. Editable Email Notification

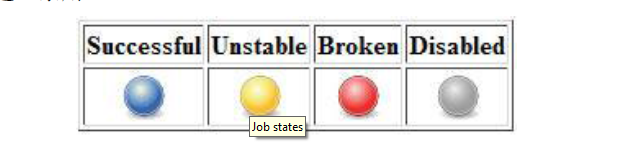


1. Deploy war/ear to a container



注意：这种部署方式导致tomcat加载时卡死现象，需要手动重启。

## 监控



Successful：完成构建，且认为是稳定的。

Unstable：完成构建，但是不稳定。

Failed：构建失败

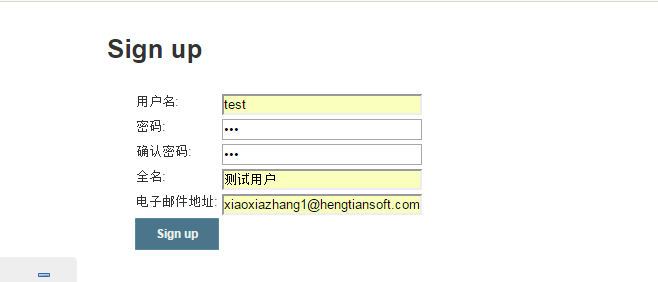
Disabled：构建已禁用

# Jenkins用户权限管理

## 配置用户注册

在已运行的Jenkins主页中，点击左侧的系统管理—>Configure Global Security进入如下界面：





## 配置用户权限

点击左侧的系统管理—>Configure Global Security 选择授权策略中的安全矩阵或者项目矩阵授权策略。说明：安全矩阵和项目矩阵授权策略的配置是一模一样的，唯一的区别是项目矩阵授权策略支持在Job的配置页面再次配置授权策略。各种权限如下(在配置页面将鼠标放到该权限上即可查看帮助)：



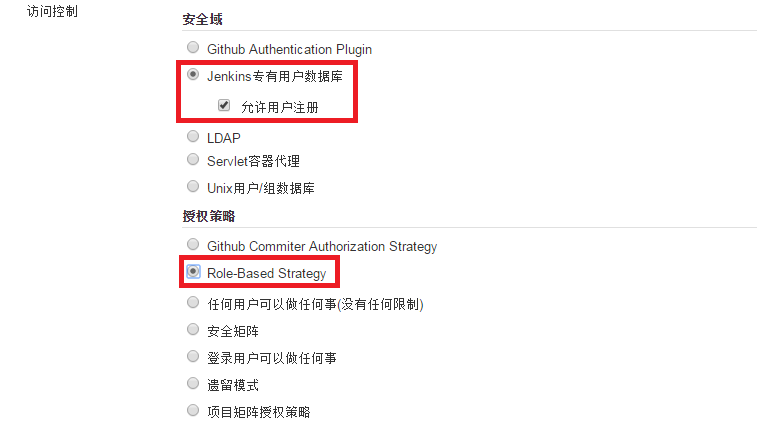
最大的权限是Overall的Administer，拥有该权限可以干任何事情。

最基本的权限是Overall的Read，用户必须赋予阅读的权限，不然什么都看不到。

# 基于角色的项目权限管理

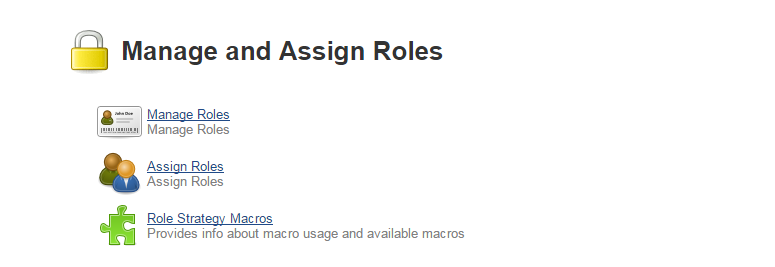
## 安装并配置插件

第一步：安装Role-based Authorization Strategy

安装插件后，进入系统设置页面，配置如下：  


## 配置权限

在系统管理页面点击Manage and Assign Roles进入角色管理页面：



### 角色管理

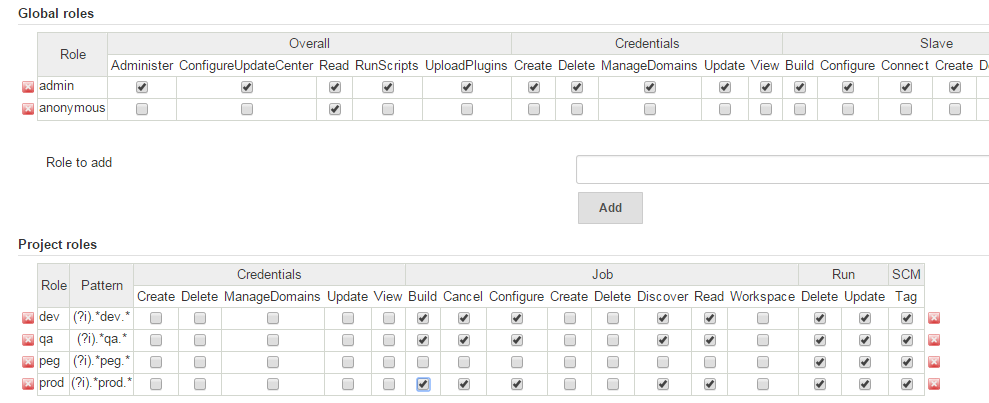
项目角色与全局角色的区别就是，项目角色只能管理项目，没有管理jenkins的权限配置。添加项目角色时，需要制定匹配项目的模式，如上图中的Pattern，官方文档介绍该选项支持正则表达式，如：

“Roger-.”：表示所有以Roger-开头的项目

“(?i)roger-.\*”：表示以roger-开头的项目并且不区分大小写， “ABC|ABC.\*”：表示如以ABC开头的项目可以配置，

“abc|bcd|efg”:表示直接匹配多个项目。

(?i).\*admin.\*：表示包含admin的项目



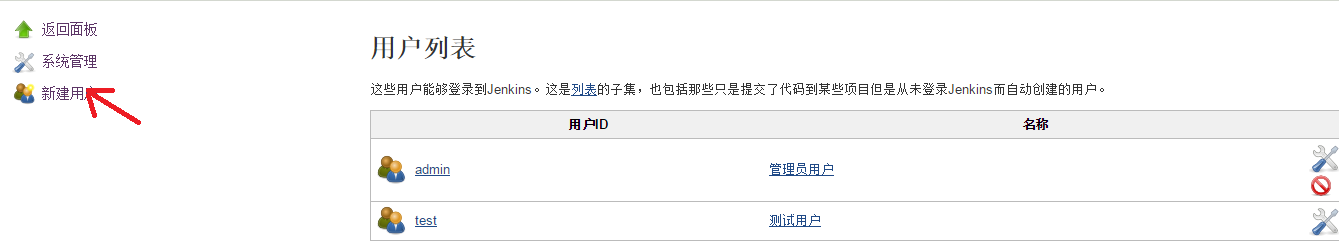
全局角色：admin、anonymous

项目角色：dev,qa,peg,prod

### 创建用户

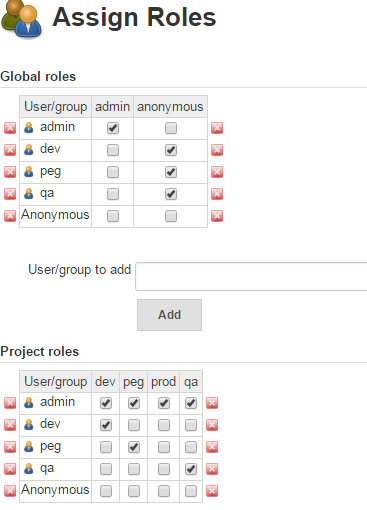
在分配角色之前需要先创建用户。





### 分配角色

选择Assign Roles可以为用户分配所属角色，可以分配全局角色和项目角色。



将不同的用户分别分配给不同的角色，这样用户就可以具有角色所拥有的权限。

# Jenkins其他技巧

## 迁移和备份

因为Jenkins的所有的数据都是以文件的形式存放在JENKINS\_HOME目录中。所以不管是迁移还是备份，只需要操作JENKINS\_HOME就行了。

迁移：建议将JENKINS\_HOME打包后在拷贝，windows上可以用zip，rar等，Linux上有zip，tar等。然后将打包的文件解压到新的JENKINS\_HOME目录就行了。

备份：可以将jenkins纳入版本控制。如果是临时备份，整个压缩文件就行了。

## 更改Jenkins的主目录

### 使用你Web容器的管理工具

设置JENKINS\_HOME环境参数.

打开tomcat的bin目录，编辑catalina.sh文件。

在# OS specific support. $var \_must\_ be set to either true or false.上面添加：export JENKINS\_HOME=""

在引号中填入你的路径。

### 在启动Web容器之前设置JENKINS\_HOME环境变量

编辑profile文件：vi /etc/profile

在最后加入：export JENKINS\_HOME=xxxx

保存，退出后执行：source /etc/profile