

持续集成篇

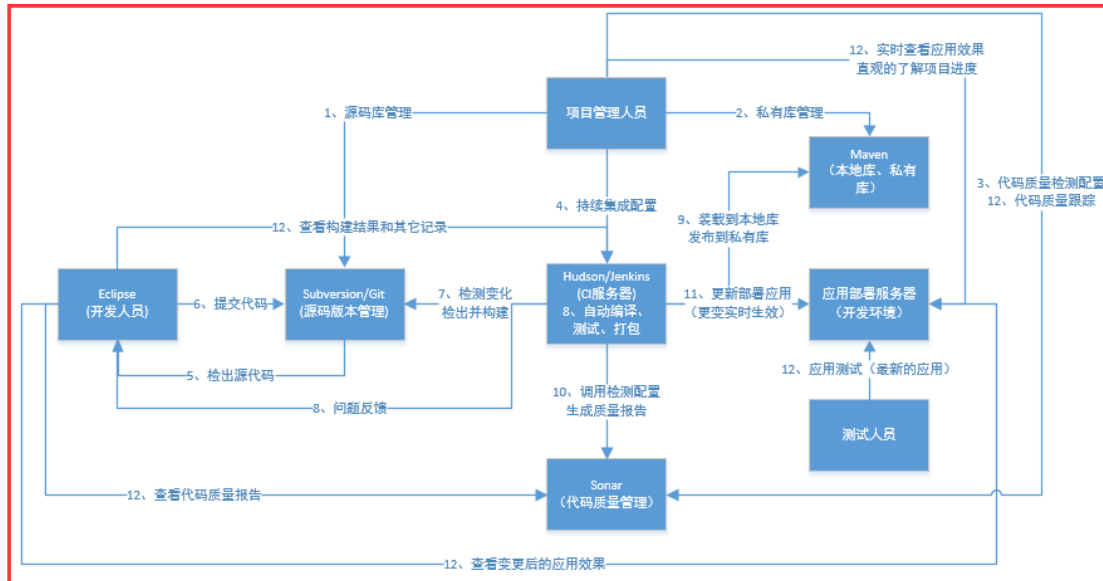
Hudson 持续集成服务器的安装配置与使用

IP: 192.168.4.221 8G 内存 (Hudson 多个工程在同时构建的情况下比较耗内存)

环境: CentOS 6.6、JDK7

Hudson 不需要用到数据库

Hudson 只是一个持续集成服务器(持续集成工具), 要想搭建一套完整的持续集成管理平台, 还需要用到前面课程中所讲到的 SVN、Maven、Sonar 等工具, 按需求整合则可。



1、安装 JDK 并配置环境变量 (略)

`JAVA_HOME=/usr/local/java/jdk1.7.0_72`

2、Maven 本地仓库的安装 (使用 Maven 作为项目构建与管理工具):

(1) 下载 maven-3.0.5

(注意: 建议不要下载 3.1 或更高版本的 Maven, 因为与 Hudson 进行集成时会有问题, 之前有遇到过):

```
# wget http://mirrors.hust.edu.cn/apache/maven/maven-3/3.0.5/binaries/apache-maven-3.0.5-bin.tar.gz
```

(2) 解压:

```
# tar -zxvf apache-maven-3.0.5-bin.tar.gz
# mv apache-maven-3.0.5 maven-3.0.5
```

(3) 配置 Maven 环境变量:

```
# vi /etc/profile
## maven env
export MAVEN_HOME=/root/maven-3.0.5
export PATH=$PATH:$MAVEN_HOME/bin
```



```
# source /etc/profile
```

(4)Maven 本地库配置:



settings.xml

3、配置 HudsonHome, 在/root 目录下创建 HudsonHome 目录, 并配置到环境变量

```
# mkdir HudsonHome
```

切换到 root 用户, 在/etc/profile 中配置全局环境变量

```
# vi /etc/profile
```

```
## hudson env
```

```
export HUDSON_HOME=/root/HudsonHome
```

```
# source /etc/profile
```

4、下载最新版 Tomcat7, 当前最新版为 7.0.59:

```
# wget http://apache.fayea.com/tomcat/tomcat-7/v7.0.59/bin/apache-tomcat-7.0.59.tar.gz
```

5、解压安装 Tomcat:

```
# tar -zxvf apache-tomcat-7.0.59.tar.gz
```

```
# mv apache-tomcat-7.0.59 hudson-tomcat
```

移除/root/hudson-tomcat/webapps 目录下的所有文件:

```
# rm -rf /root/hudson-tomcat/webapps/*
```

将 Tomcat 容器的编码设为 UTF-8:

```
# vi /root/hudson-tomcat/conf/server.xml
```

```
<Connector port="8080" protocol="HTTP/1.1"
```

```
connectionTimeout="20000"
```

```
redirectPort="8443" URIEncoding="UTF-8" />
```

如果不把 Tomcat 容器的编码设为 UTF-8, 在以后配置 Hudson 是有下面的提示:

Hudson 自动



系统管理

⚠ Your container doesn't use UTF-8 to decode URLs. If you use non-ASCII characters as a job name etc, this will cause problems. See [Containers](#) and [Tomcat i18n](#) for more details.



[Configure Security](#)

Configure Authentication and Authorization Strategy for Hudson Security.



龙果学院微信公众号: ron-coo

设置 hudson-tomcat 的内存

```
# vi /root/hudson-tomcat/bin/catalina.sh
```

#!/bin/sh 下面增加:

```
JAVA_OPTS='-Xms512m -Xmx2048m'
```

6、下载最新版的 Hudson (这里是 3.2.2 版) 包:

```
# wget http://mirror.bit.edu.cn/eclipse/hudson/war/hudson-3.2.2.war
```

将 war 包拷贝到 hudson-tomcat/webapps 目录, 并重命名为 hudson.war

```
# cp /root/hudson-3.2.2.war /root/hudson-tomcat/webapps/hudson.war
```

7、防火墙开启 8080 端口, 用 root 用户修改/etc/sysconfig/iptables,

```
# vi /etc/sysconfig/iptables
```

增加:

```
## hudson-tomcat port:8080
```

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT
```

重启防火墙:

```
# service iptables restart
```

8、设置 hudson-tomcat 开机启动:

在虚拟主机中编辑/etc/rc.local 文件,

```
# vi /etc/rc.local
```

加入:

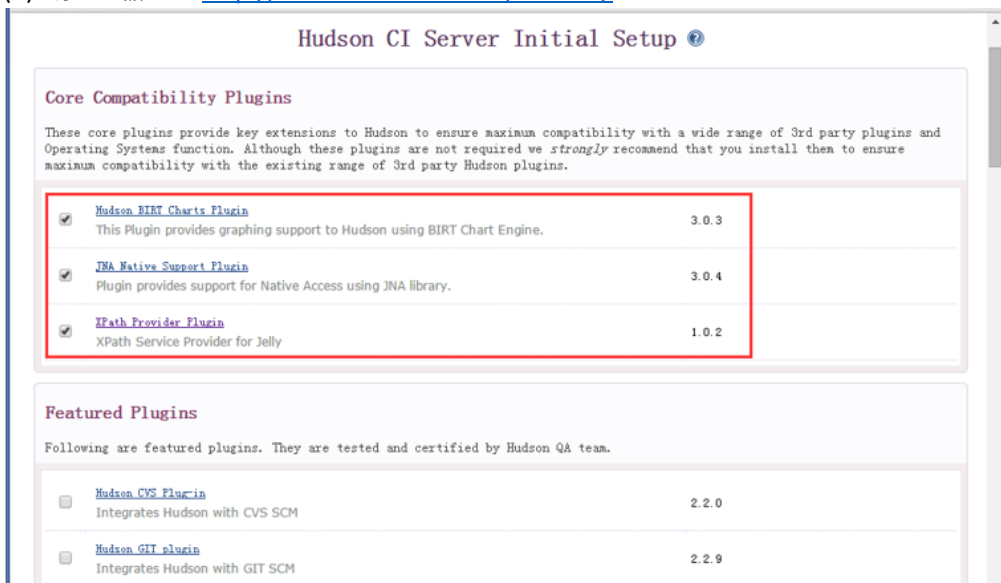
```
/root/hudson-tomcat/bin/startup.sh
```

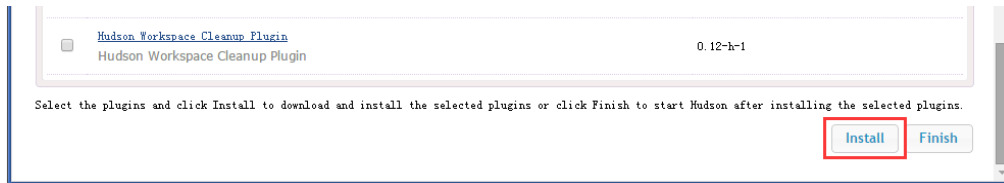
9、启动 hudson-tomcat

```
# /root/hudson-tomcat/bin/startup.sh
```

10、配置 Hudson:

(1)浏览器输入: <http://192.168.4.221:8080/hudson/>

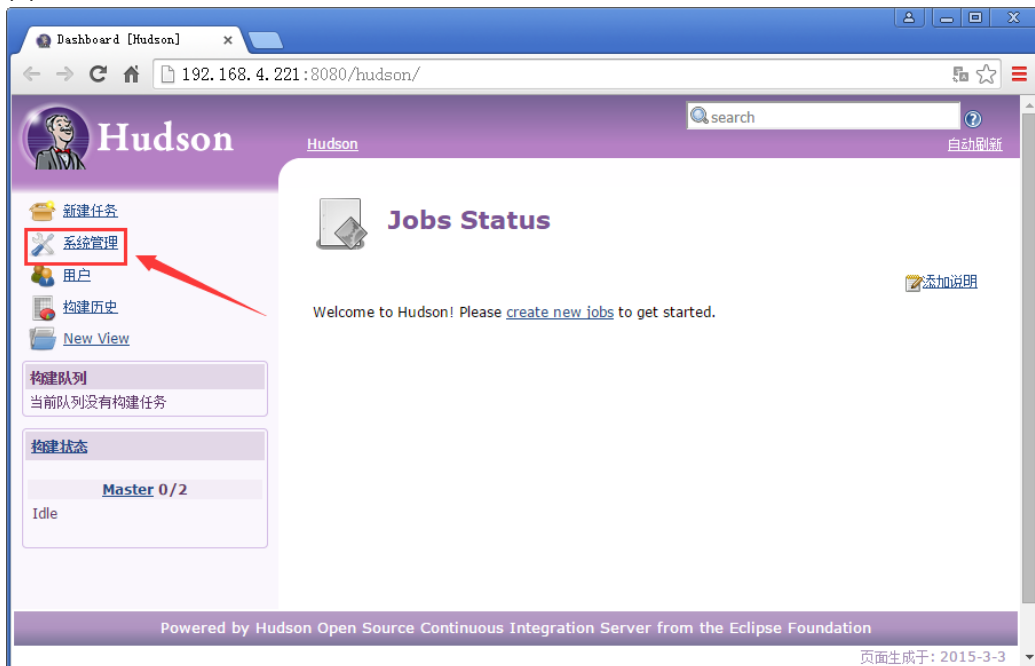




初始化安装需要安装 3 个默认勾选中的插件（如上图红色部分），其它插件可以等初始化安装完成之后再选择安装。

点击“Install”安装按钮后，需要等待一会时间才能安装完成。安装完成后按“Finish”按钮。安装的插件保存在 `/root/HudsonHome/plugins` 目录。

(2)初始化完成后就会进行 Hudson 的配置管理界面：

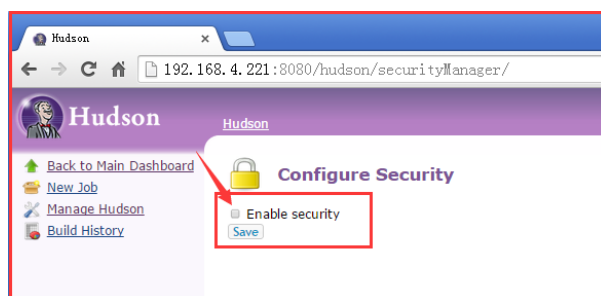


安全配置



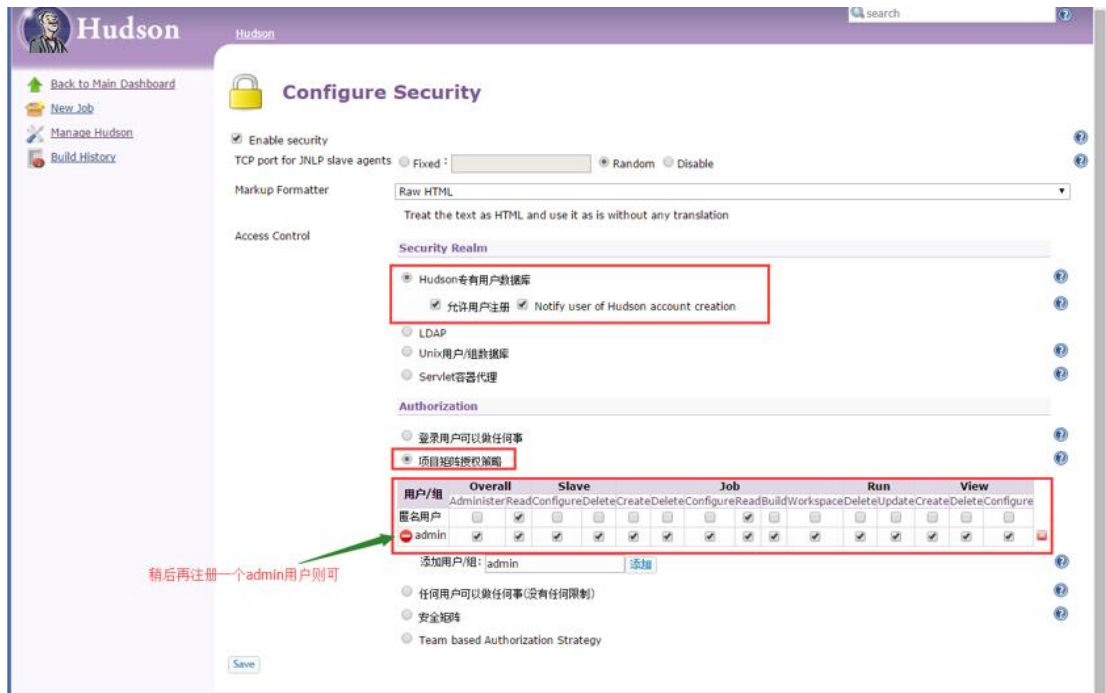


启用安全配置

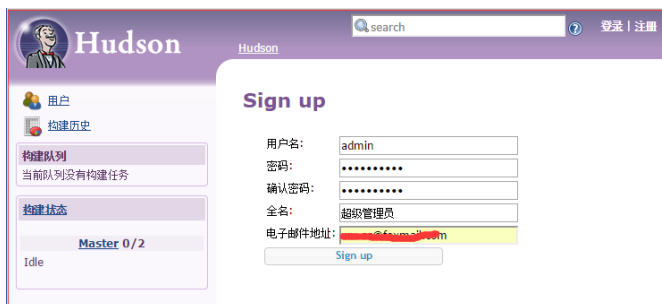
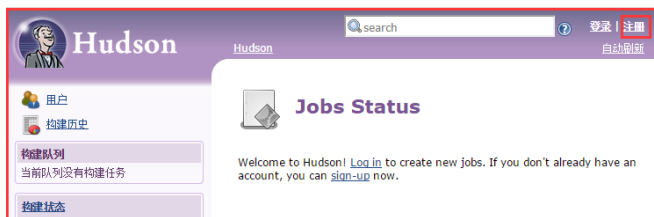


使用项目矩阵授权策略





注册一个超级管理员账号



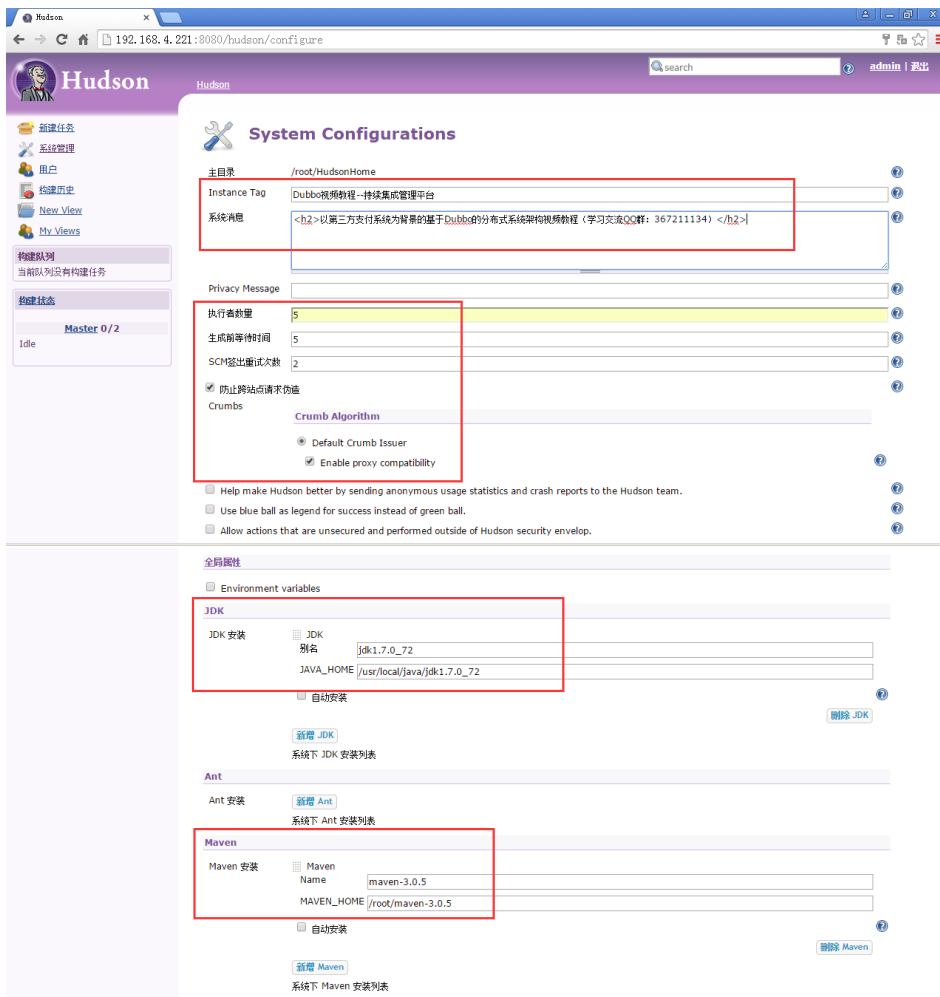
系统设置



龙果学院微信公众号: ron-coo



配置系统信息、JDK、Maven



Shell executable

邮件通知

SMTP服务器: smtp.qq.com

用户默认邮件后缀

系统管理员邮件地址: wu-sc@foxmail.com

Hudson URL: http://192.168.4.221:8080/hudson/

☒ 使用SMTP认证

用户名: 840765167@qq.com

密码:

☐ 使用SSL协议

SMTP端口: 25

Charset: UTF-8

Save 保存

Email was successfully sent

用系统管理员测试邮件配置

Powered by Hudson Open Source Continuous Integration Server from the Eclipse Foundation

保存后的效果

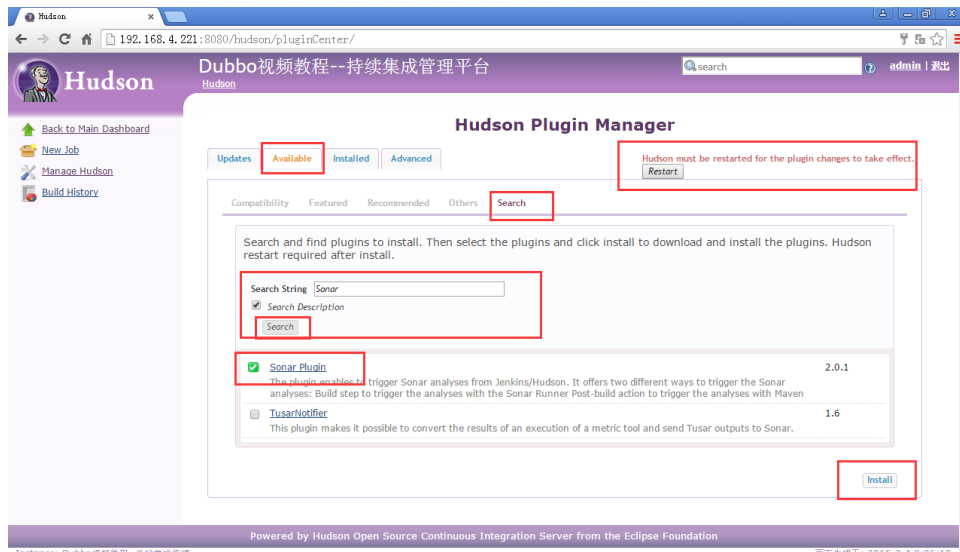


插件安装

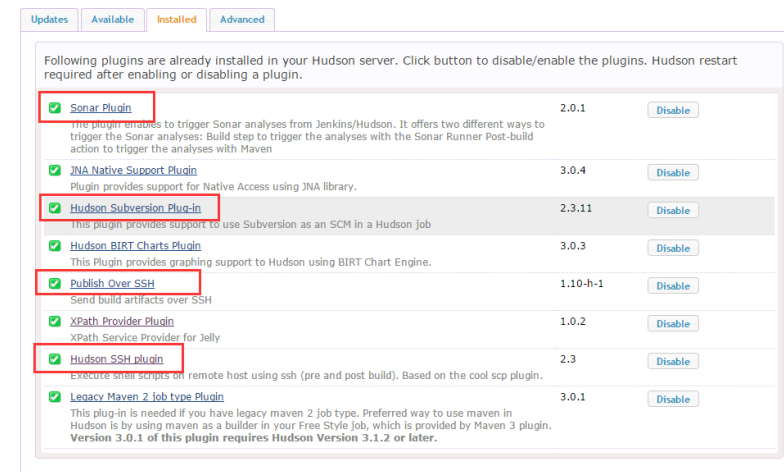


结合我们想要实现的持续集成功能，需要安装如下几个插件。如想集成更多功能，自行添加插件并配置则可。（注意：现在我们使用了 SonarQube 质量管理平台，则不再需要在 Hudson 中单独去安装 CheckStyle、Findbugs、PMD、Cobertura 等 Sonar 中已有的插件）逐个搜索你想要安装的插件并点击安装，安装完之后重启 Hudson。如下图所示：

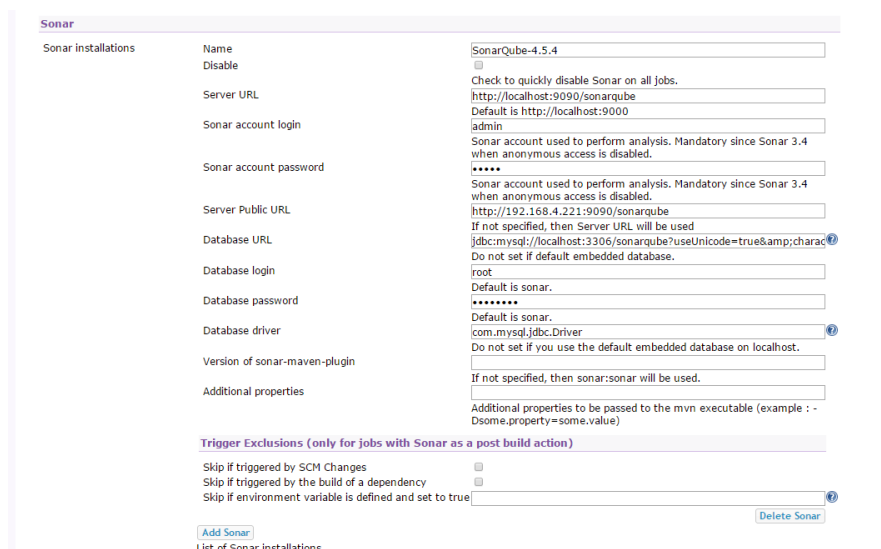




Hudson Plugin Manager



在 Hudson 中配置 SonarQube 链接



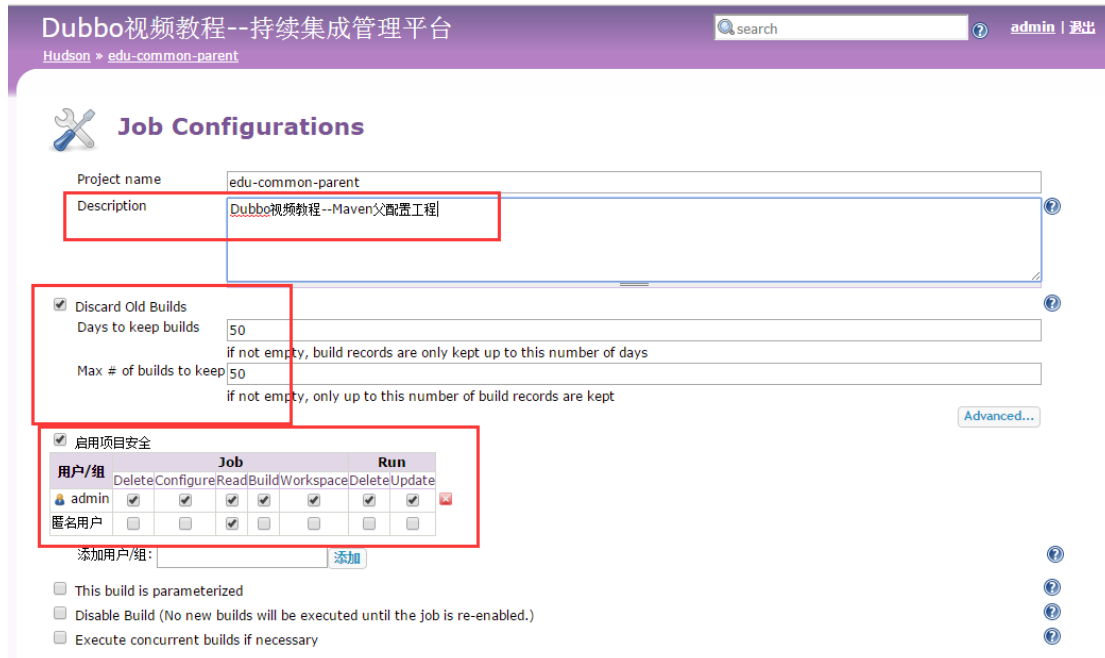
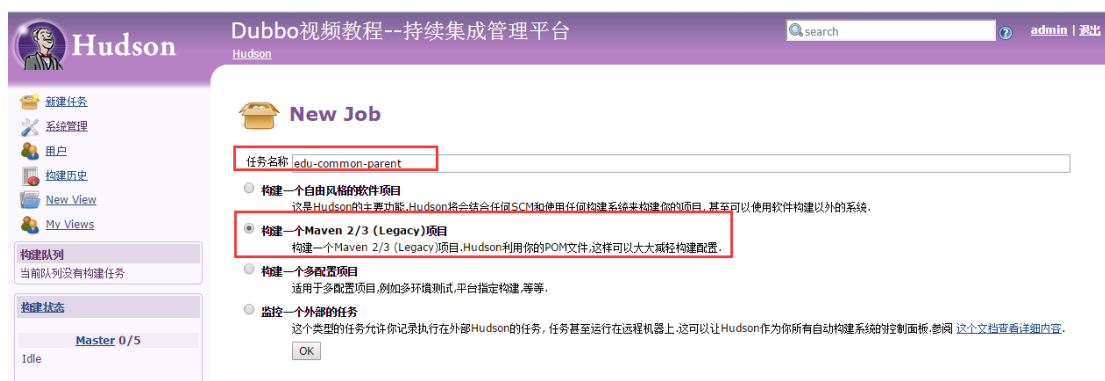
以上就是 Hudson 的基本安装和配置，更多其它配置和功能可自行扩展。



龙果学院微信公众号: ron-coo

Hudson 的使用（使用 Hudson 来自动化编译、分析、打包、发布、部署项目）

添加项目



Source Code Management

☐ None

☒ Subversion

Modules

Repository URL

Local module directory (optional)

Repository depth option

Ignore externals option

http://192.168.4.221/svn/wusc_edu/edu-common-parent

.

infinity

☐

Add more locations...

Check-out Strategy

Use 'svn update' as much as possible

Use 'svn update' whenever possible, making the build faster. But this causes the artifacts from the previous build to remain when a new build starts.

Repository browser

(Auto)

Advanced...

Build Triggers

☐ Build whenever a SNAPSHOT dependency is built

☐ Build after other jobs are built

☐ Trigger builds remotely (e.g., from scripts)

☐ Build periodically

☒ Poll SCM

Schedule

Build

Root POM

Goals and options

pom.xml

clean install deploy

Advanced...

Build Settings

☒ E-mail Notification

Recipients

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

☒ Send e-mail for every unstable build

☐ Send separate e-mails to individuals who broke the build

Build Environment

☐ Send files or execute commands over SSH before the build starts

☐ Send files or execute commands over SSH after the build runs

☐ Execute shell script on remote host using ssh

Post-build Actions

☐ Aggregate downstream test results

☐ Archive the artifacts

☐ Build other jobs

☐ Deploy artifacts to Maven repository

☐ Sonar

☐ Send build artifacts over SSH

Save

Hudson

返回

状态

变更集

立即构建

删除Project

设置

Modules

Subversion Polling Log

Build History (趋势图)

全部

失败

Dubbo视频教程--持续集成管理平台

Hudson > edu-common-parent

search

admin | 退出

自动刷新

Project edu-common-parent

Dubbo视频教程--Maven父配置工程

编辑描述

Disable Project

Workspace

Recent Changes

永久连接

