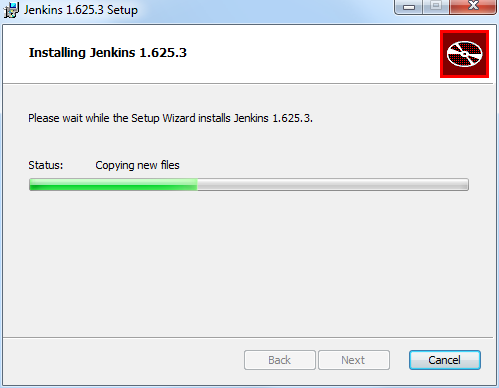
本文简要介绍了基于Jenkins的持续集成(Continuous integration, CI)环境的搭建与配置。

# Jenkins安装与配置

## Jenkins的安装

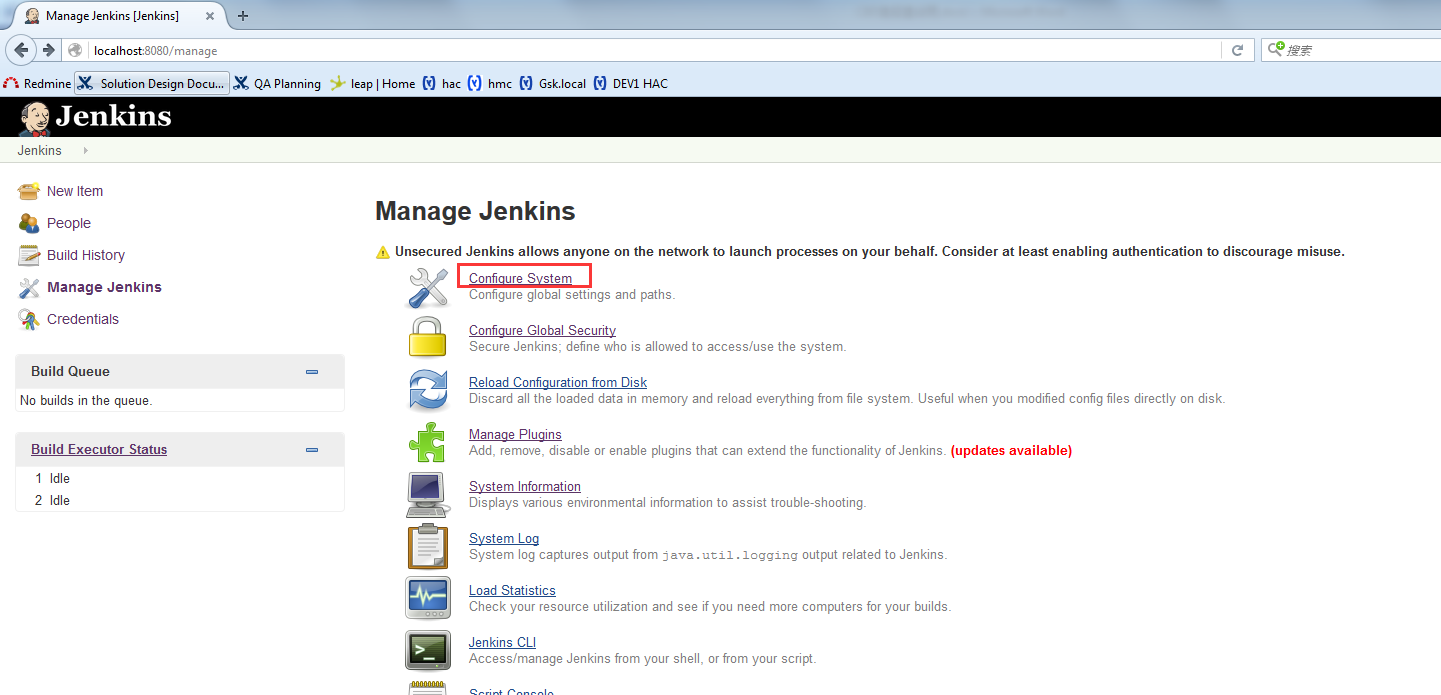
Jenkins的官方网址是<http://jenkins-ci.org/>。请在官方网站下载与操作系统相匹配的安装包。此处建议大家下载长期支持版本。

另外，Jenkins需要Java环境。请在安装前正确安装JDK及配置JAVA\_HOME以及Path环境变量。

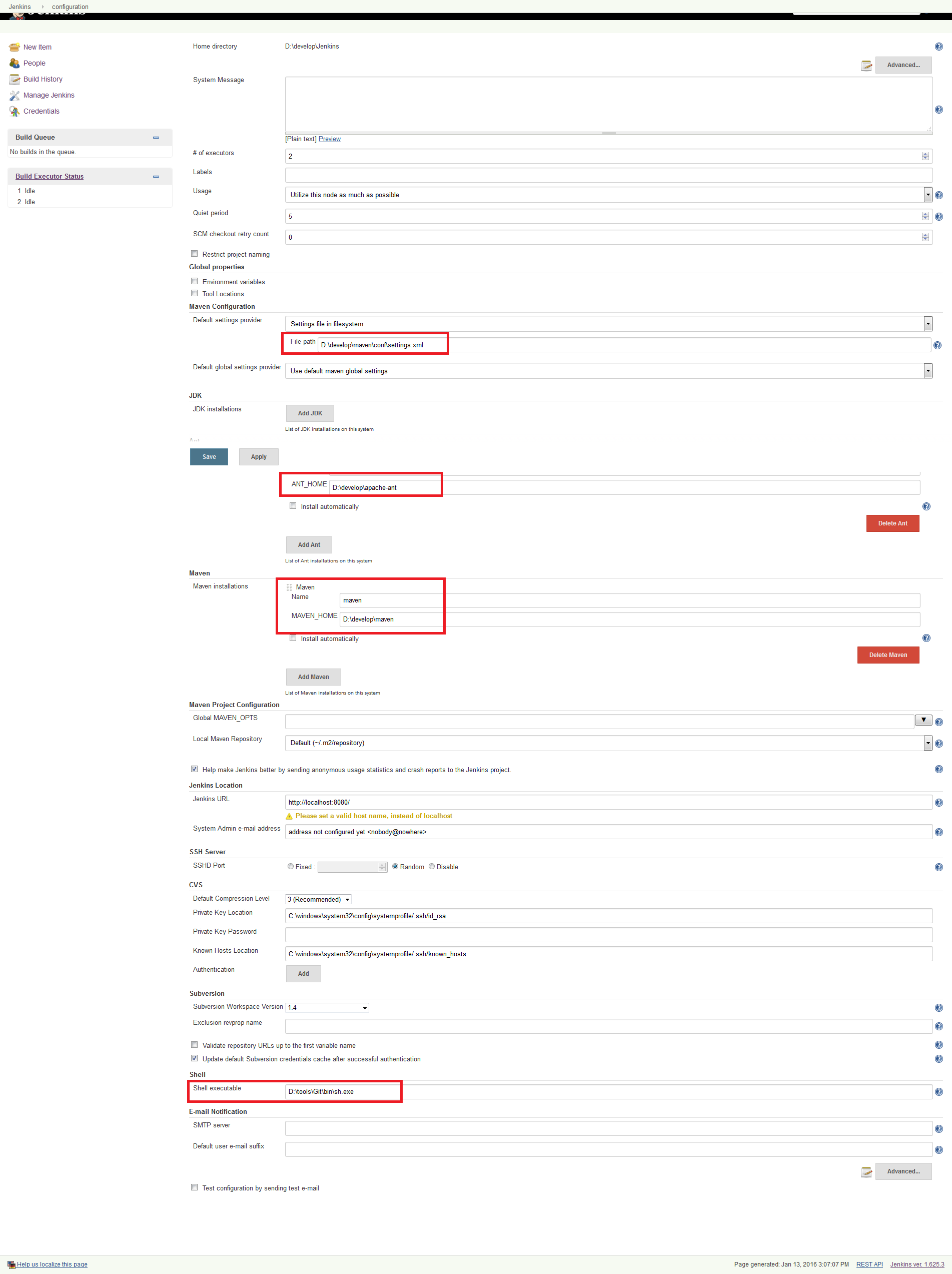


## Jenkins的配置

安装完成后，Jenkins会在默认浏览器打开。选择Manage Jenkins——Configure System配置Jenkins。



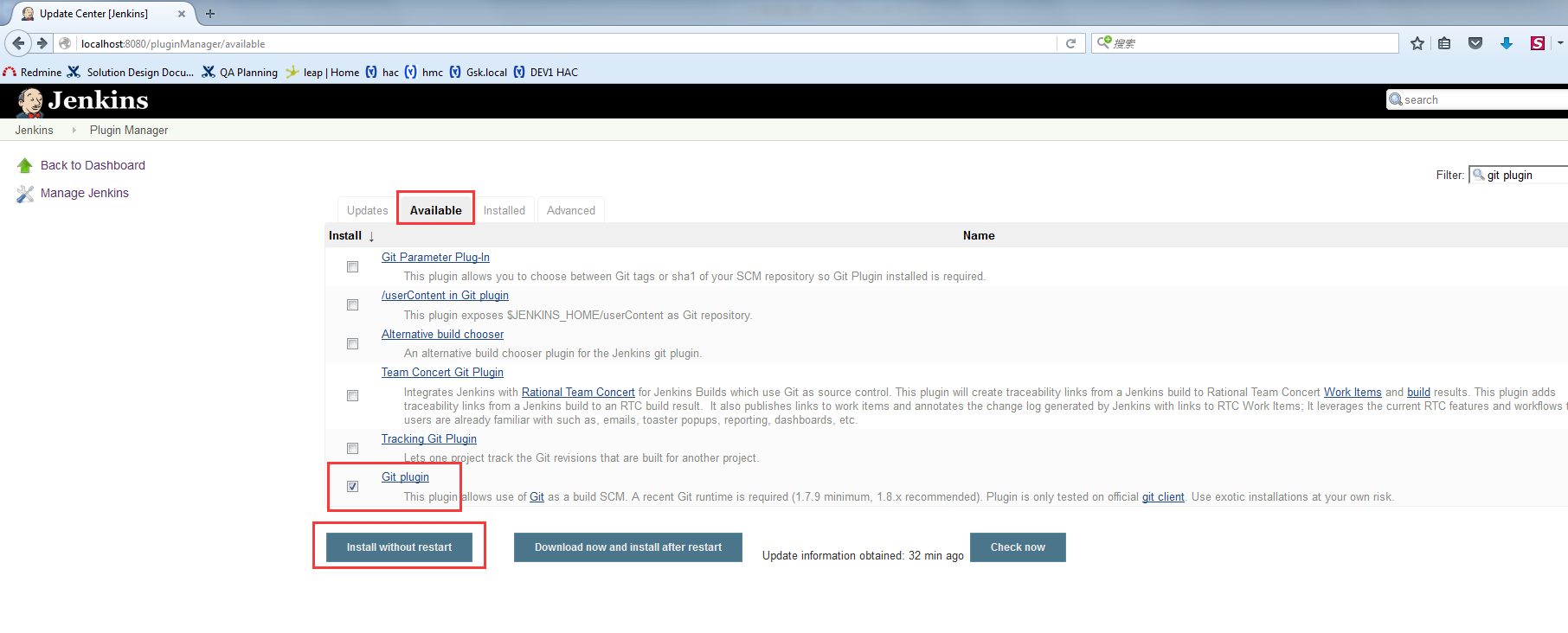
请正确配置Maven用户配置文件路径、ANT\_HOME、MAVEN\_HOME以及Shell executable。Shell executable可以配置为Window系统的cmd.exe或者shell。



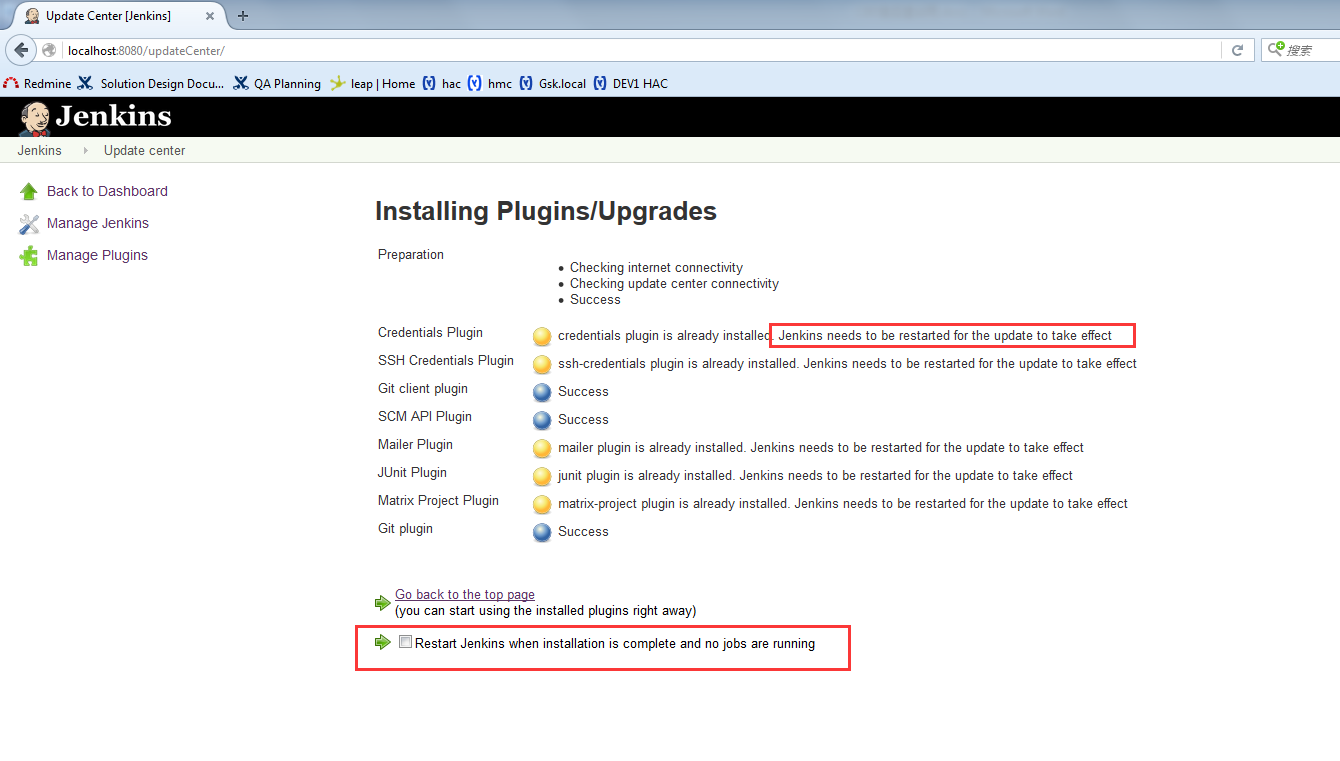
## 相关插件的安装

选择Manage Jenkins——Manage Plugins管理Jenkins的插件。需要安装Git Plugin (用于Git方式获取代码)，Selenium Plugin(用于Selenium Grid的Hub)，Cucumber Reports Plugin和Cucumber json test reporting(用于Cucumber的报告处理)。此外，可以安装PMD及Findbugs插件用于静态代码代码检查报告的显示。

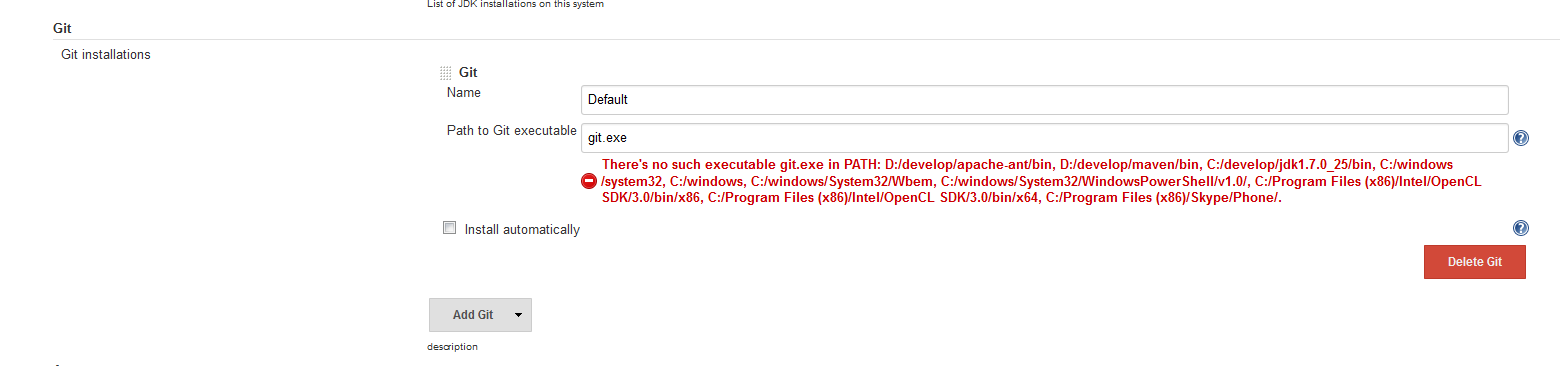
在Available Tab，显示了所有可安装的Plugin。在Filter键入插件的关键字，可以快速的筛选插件。在选择完成后点击Install without restart按钮用于完成插件的安装。

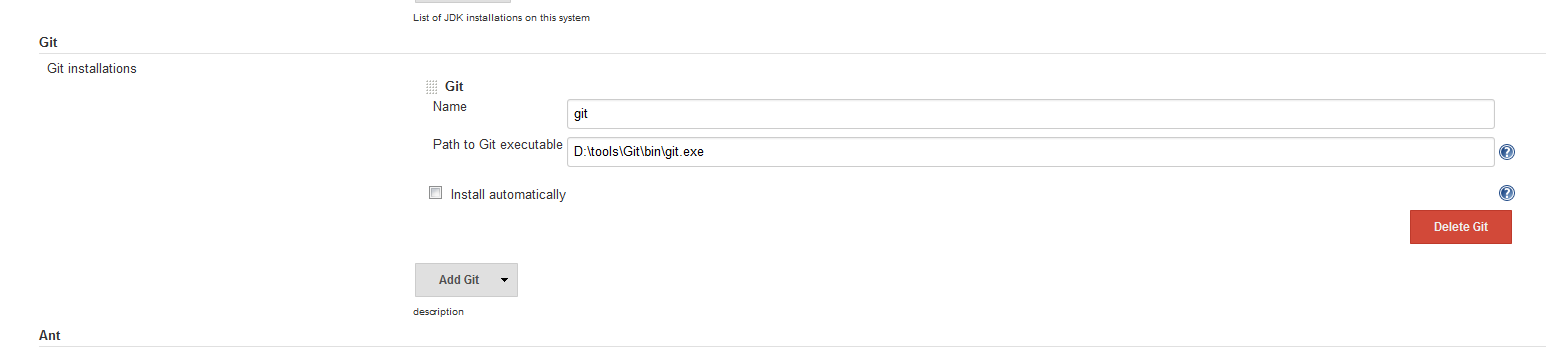


Jenkins会自动安装选择插件所依赖的全部插件。安装过程中出现需要restart Jenkins时请勾选下方的Restart Jenkins复选框。



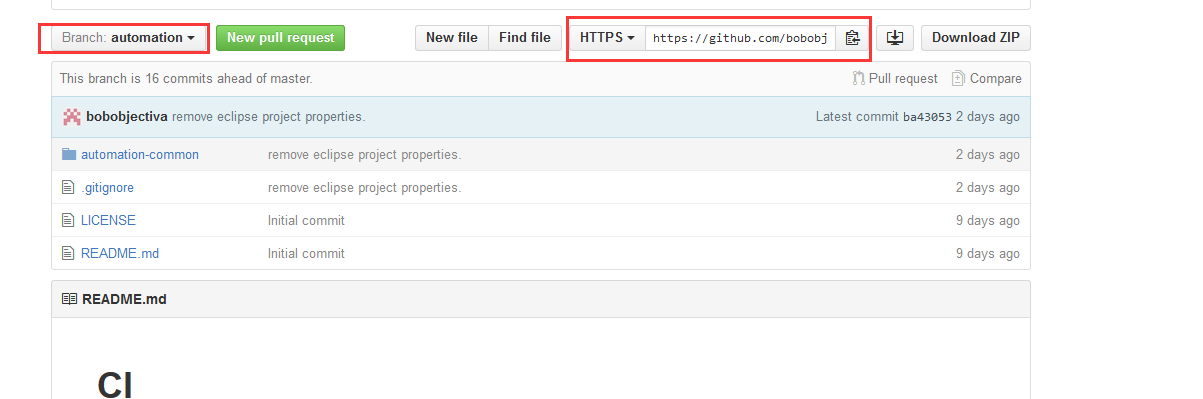
请注意，当Git插件安装完成后，需要重新配置Jenkins的Git信息。打开Configure System可以看到Jenkins的错误信息提示。配置正确路径后，保存最新配置。



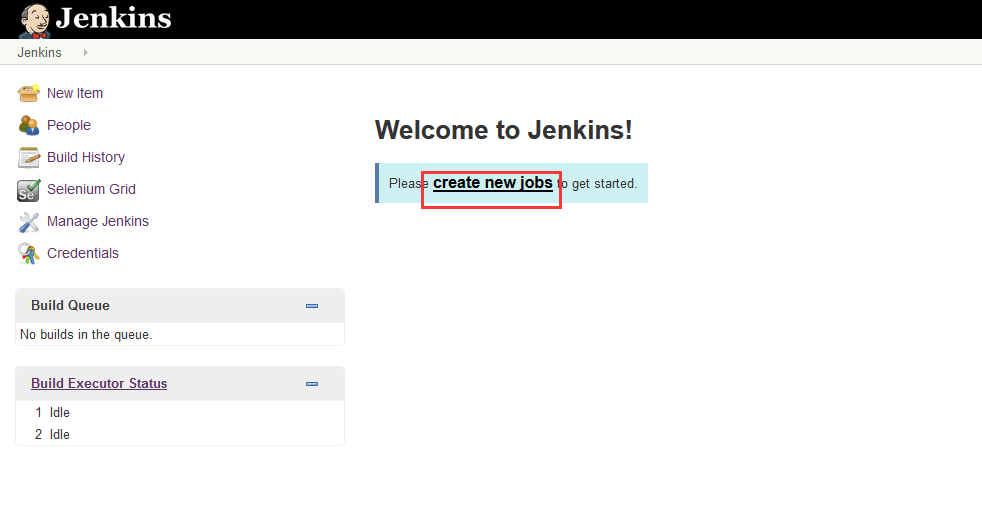


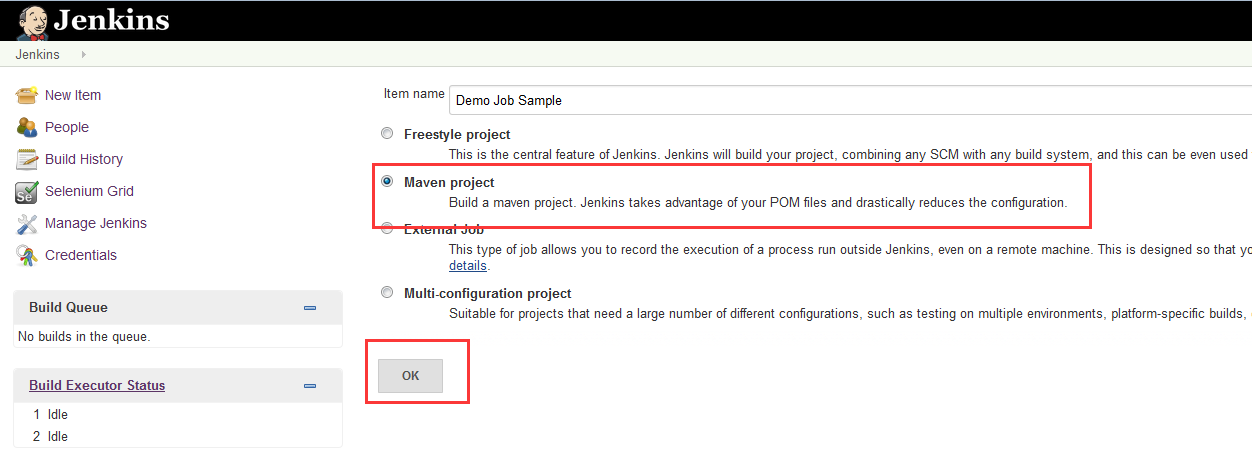
# Jenkins Job配置描述

本文演示代码托管在GItHub，Jenkins Job的配置将以Git获取代码为例说明。打开GitHub，请留意演示代码分支及HTTPS地址。

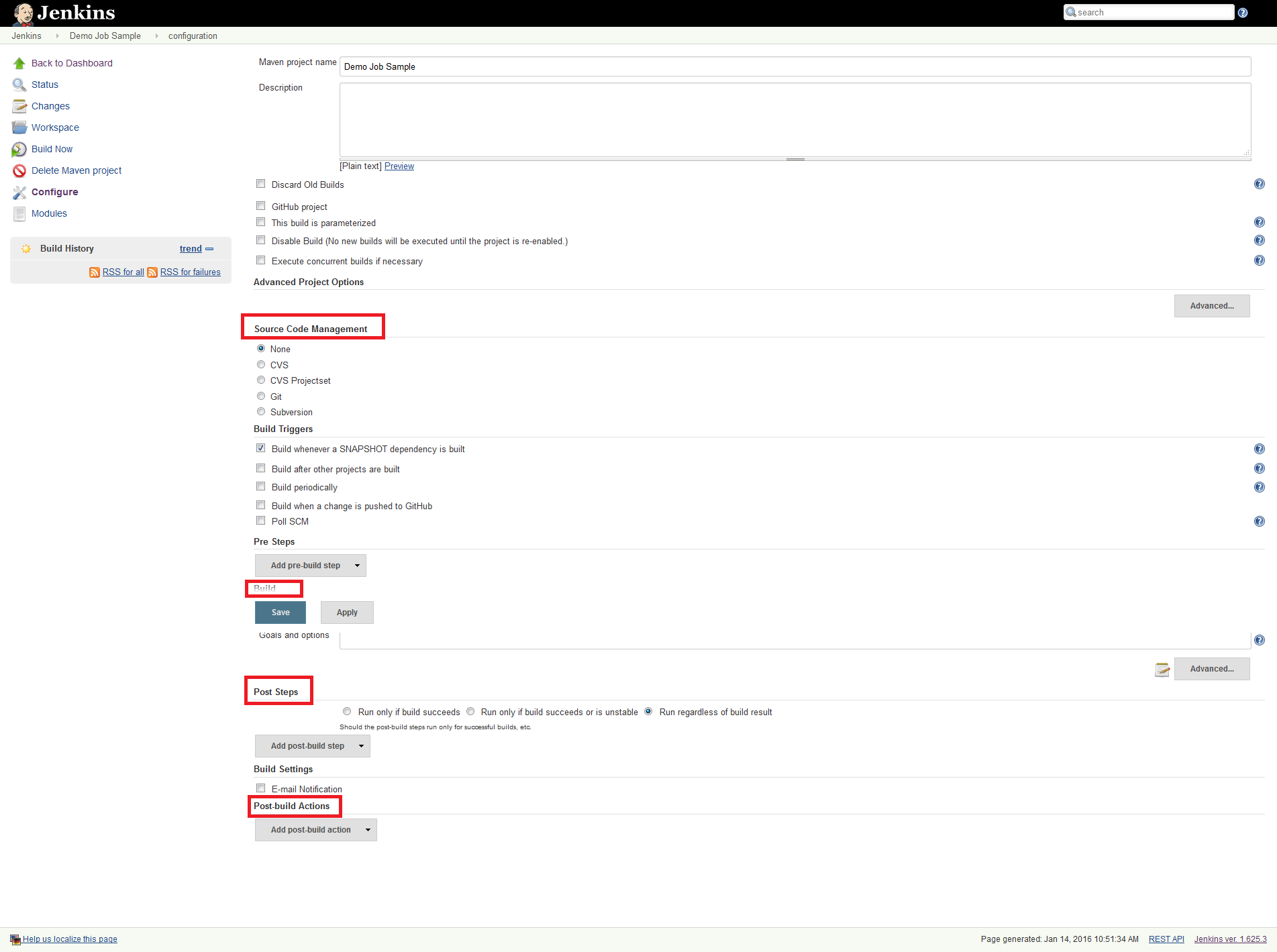


打开Jenkins，选择新建Jenkins Job。输入Job的名称为Demo Job Sample，由于演示代码为Maven管理，选择类型为Maven Project。对于非Maven项目，可以选择Freestyle Project。



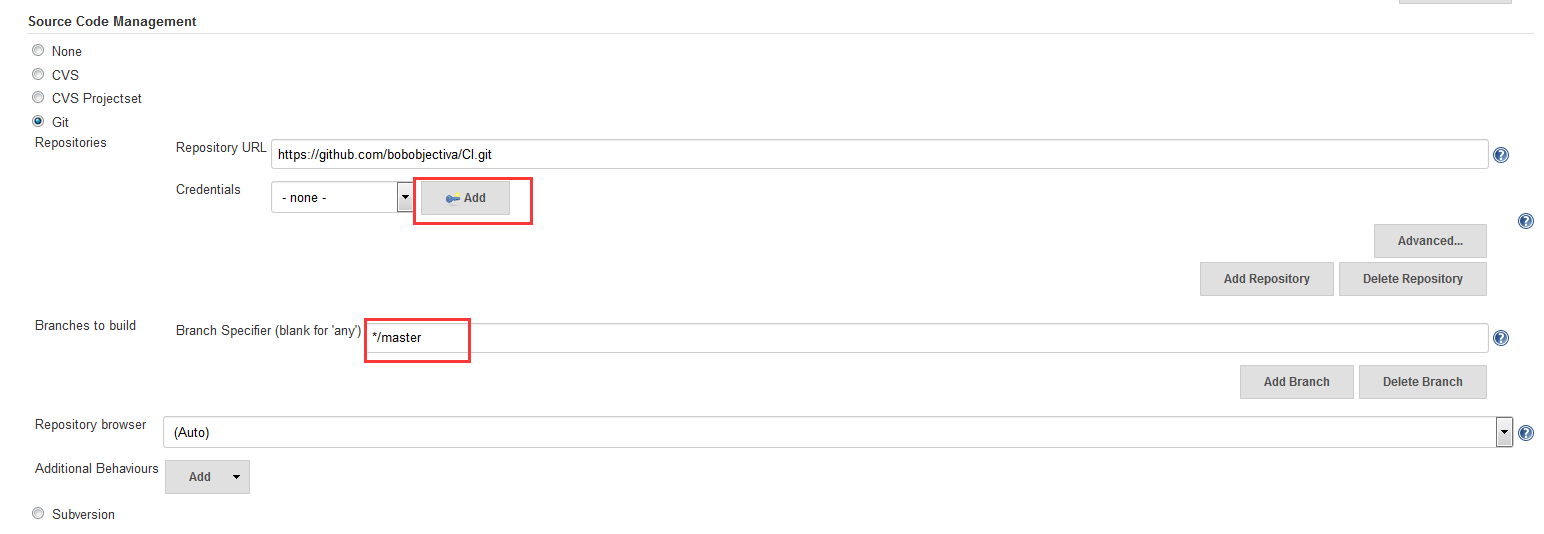


Jenkins Job的配置界面如下， 后续将分别介绍Source Code Management，Build，Post Steps以及Post-build Actions四个部分。

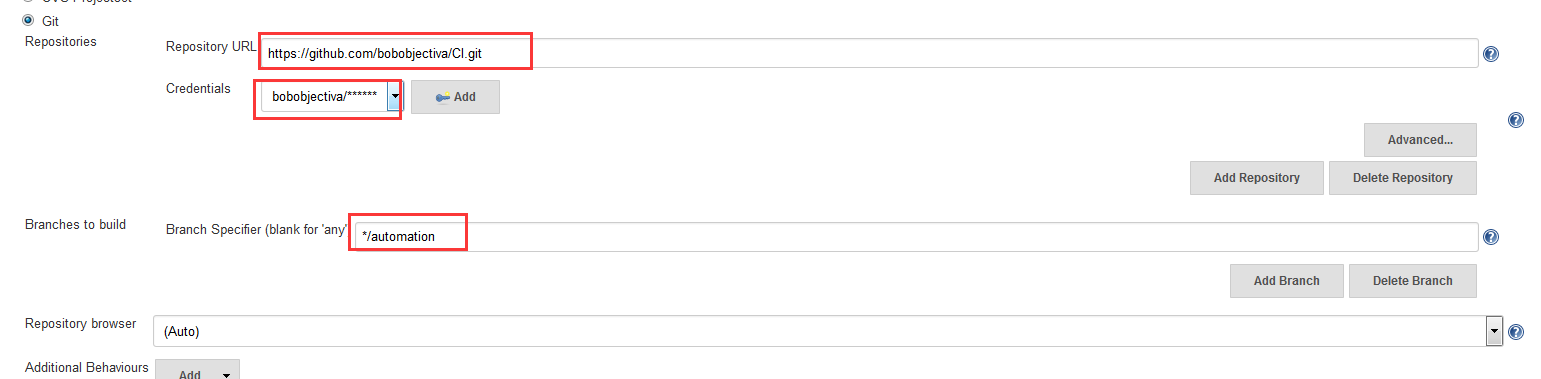


## Source Code Management

Jenkins支持主流方式的代码管理工具，如SVN，CVS，Git等，GIt支持需要通过安装Git Plugin实现。选择Git后，复制GitHub演示代码HTTPS地址，并输入验证方式。Jenkins支持多种验证方式，由于演示代码地址为HTTPS，所以此处验证方式为账号及密码。最后填写演示代码所在分支名称。







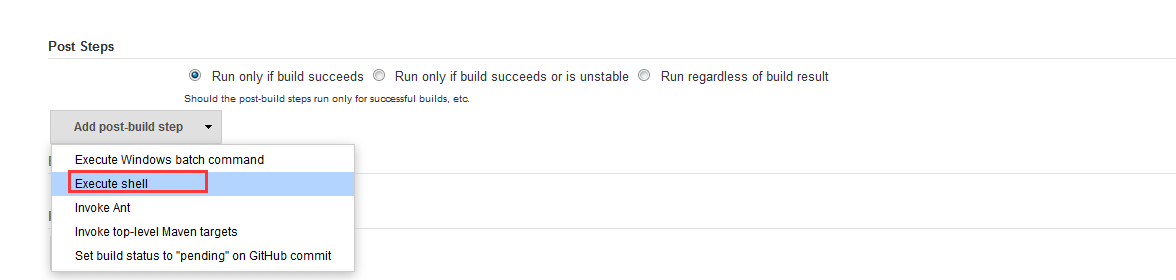
## Build

Build需要配置项目POM.xml文件路径，以及Maven执行的命令。



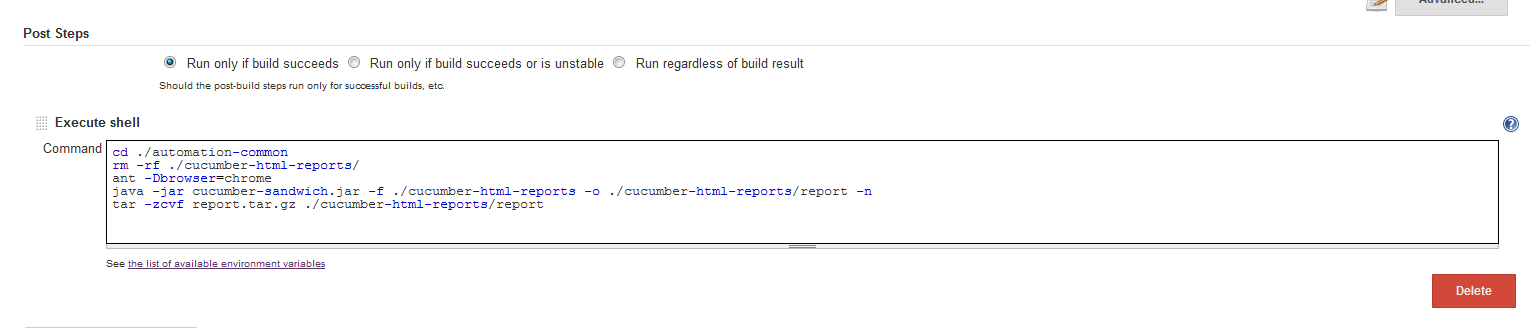
## Post Steps

首先选择Build成功后执行， 并选择执行类型为Shell。



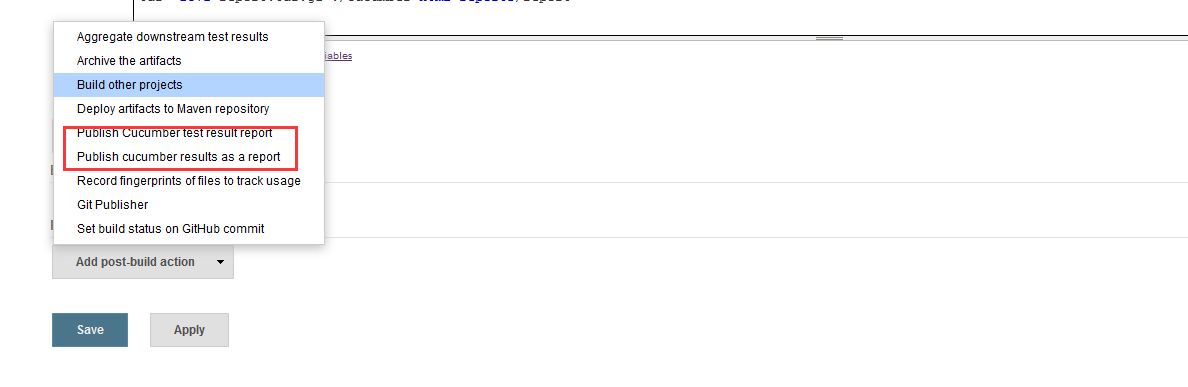
输入相应的Shell命令。从第一行开始分别执行以下操作：

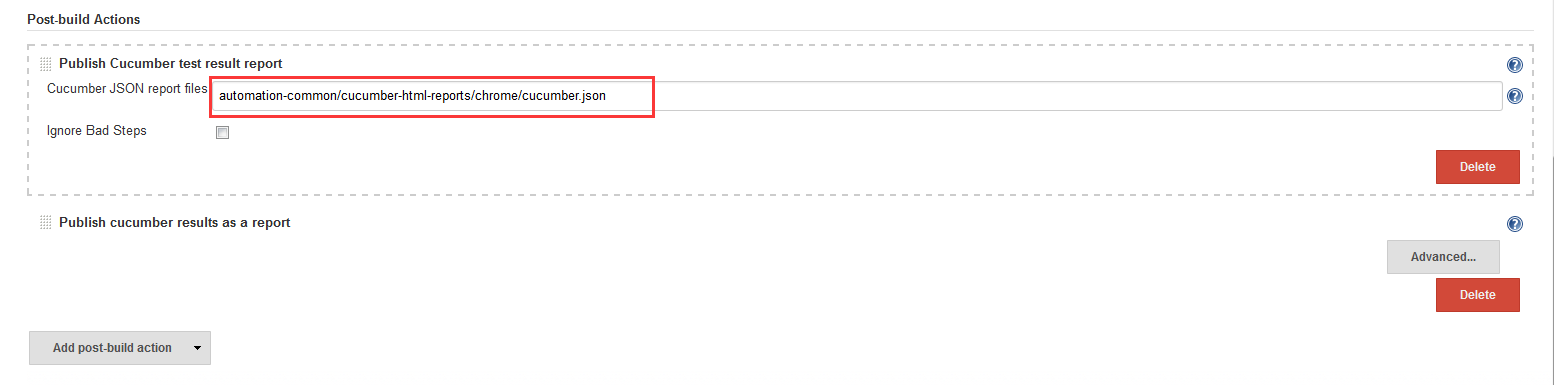
* 进入演示代码根目录
* 删除历史Cucumber报告
* Ant执行演示项目根目录Build.Xml执行的默认target，且指定浏览器类型为Chrome
* 调用Cucumber-sandwich格式化Cucumber报告
* 将cucumber-html-report目录的报告打包到report.tar.gz



## Post-build Actions

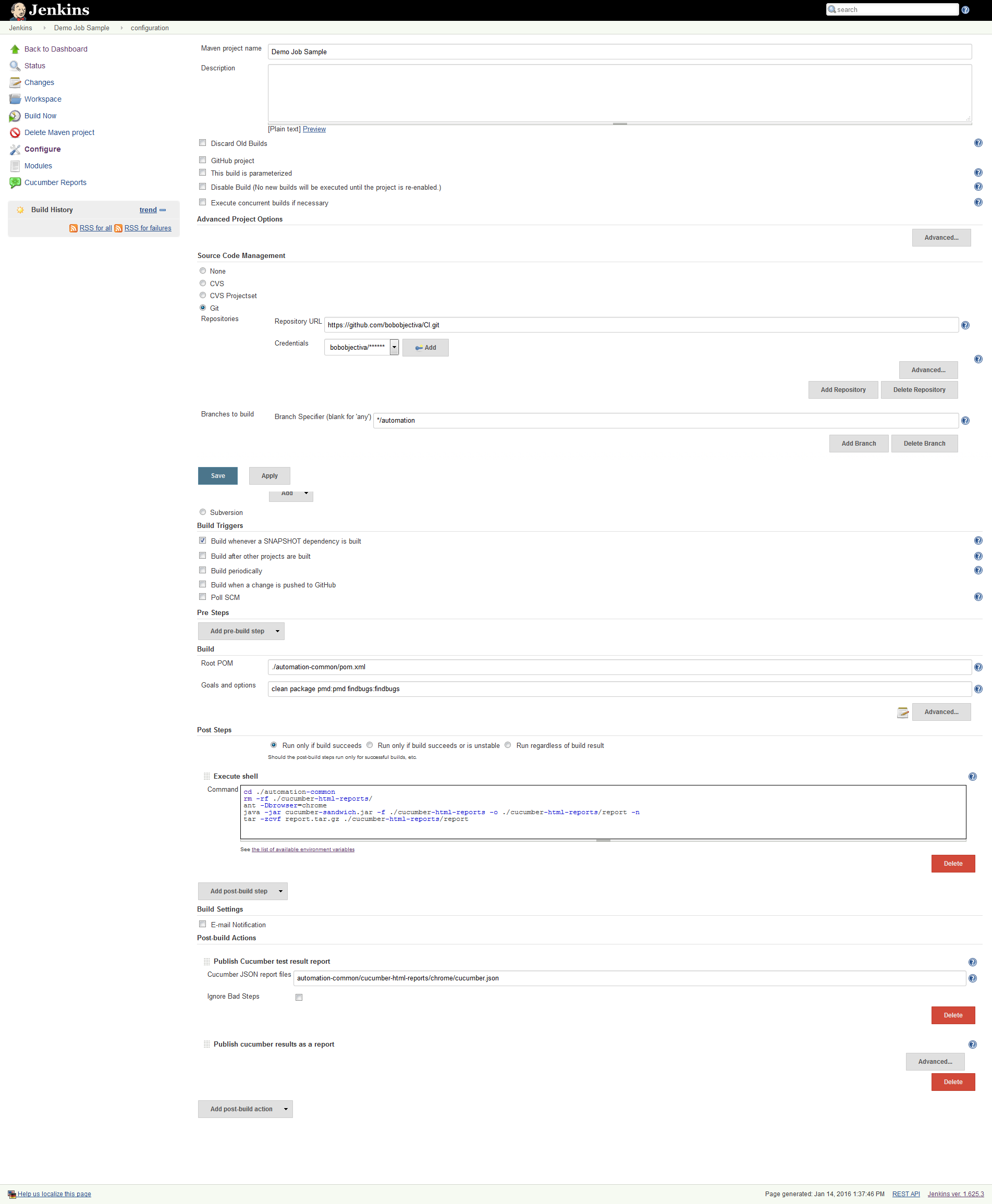
最后，配置Cucumber报告插件。对于Publish Cucumber test result report插件需要配置Json格式的报告全路径。

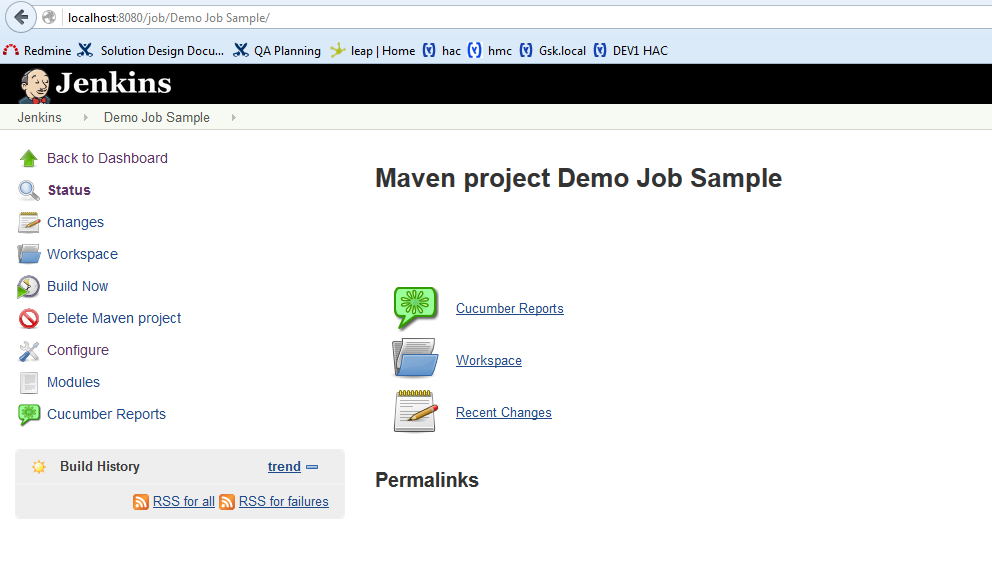




## Demo Job

配置完成的Job展示如下，点击Save按钮完成配置。

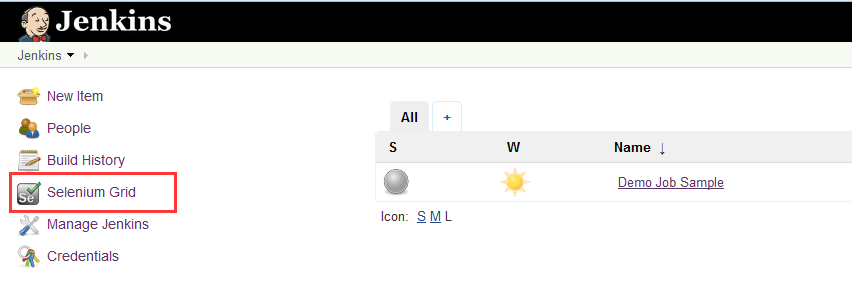


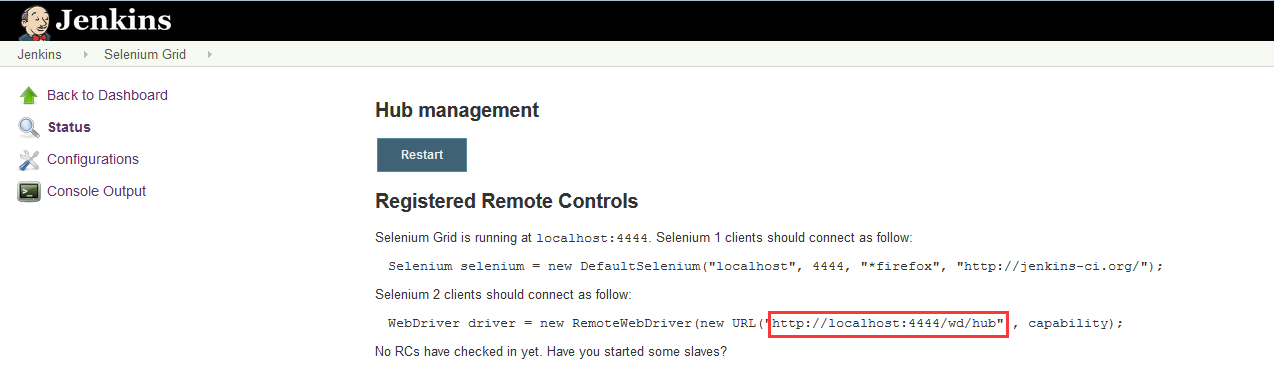


# Selenium Grid节点

## Selenium Grid节点配置

配置完成Job后，还需要启动Selenium Grid节点作为自动化测试执行节点。当安装Selenium Plugin后，可以在Jenkins首页看到Selenium Grid的选项。选择Selenium Grid，可以看到Selenium Hub的相关信息。





然后需要在相应服务器启动Selenium Server服务作为Selenium Grid的节点。使用selenium-server-standalone启动Selenium Grid的节点。启动命令如下：

java -jar selenium-server-standalone-2.48.2.jar

-Dwebdriver.chrome.driver=C:\workspace\dtautomation\automation-common\resources\chromedriver.exe

-hub <http://localhost:4444/grid/register>

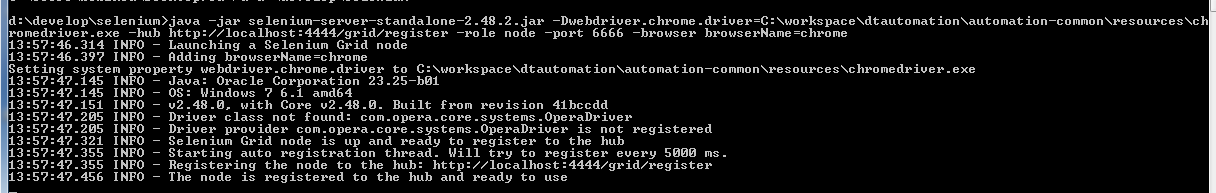
-role node -port 6666 -browser browserName=chrome

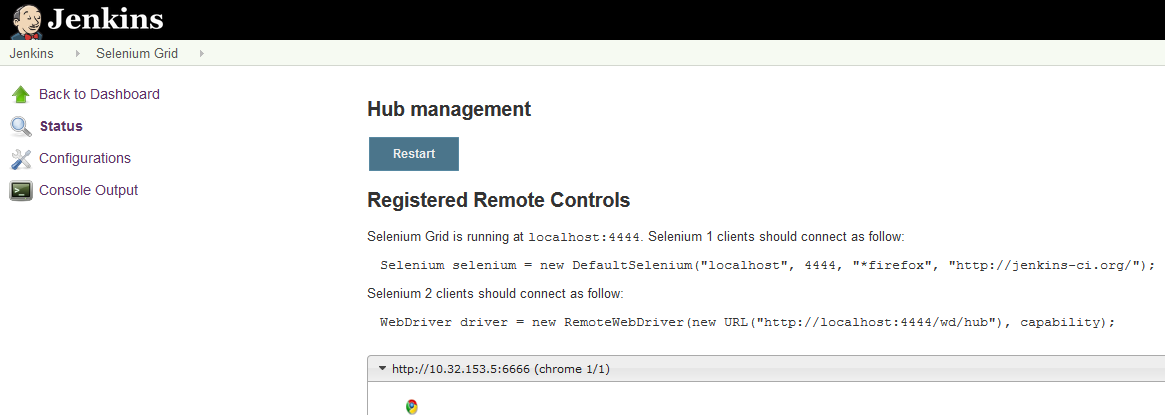
这里总共有五个参数，描述如下：

|  |  |
| --- | --- |
| **参数** | **描述** |
| webdriver.chrome.driver | 指定浏览器的可执行文件地址，此配置以Chrome为例 |
| hub | Selenium Grid Hub地址，这里使用Jenkins Selenium插件地址 |
| role | Selenium服务的角色为Node |
| port | 启动服务的端口号 |
| browser | 节点运行的浏览器名称 |

## Selenium Grid节点启动

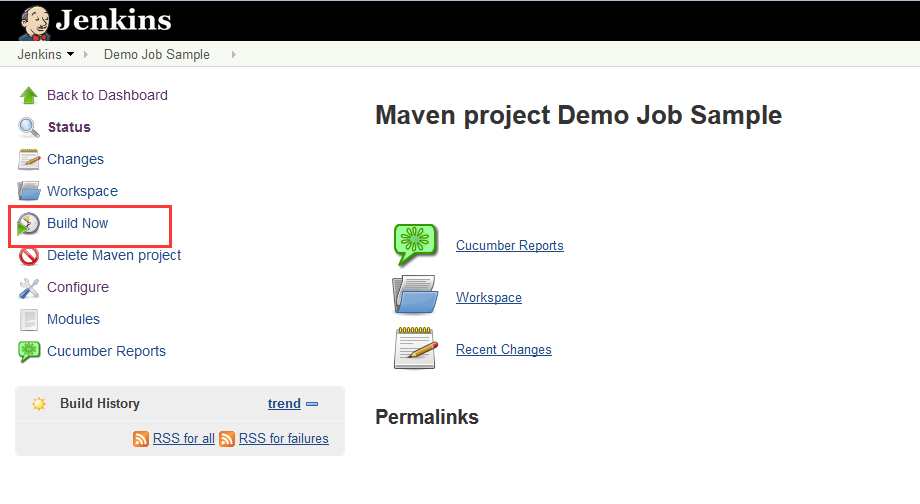
服务启动后，可以在Jenkins的Selenium Grid看到相应Node的信息。

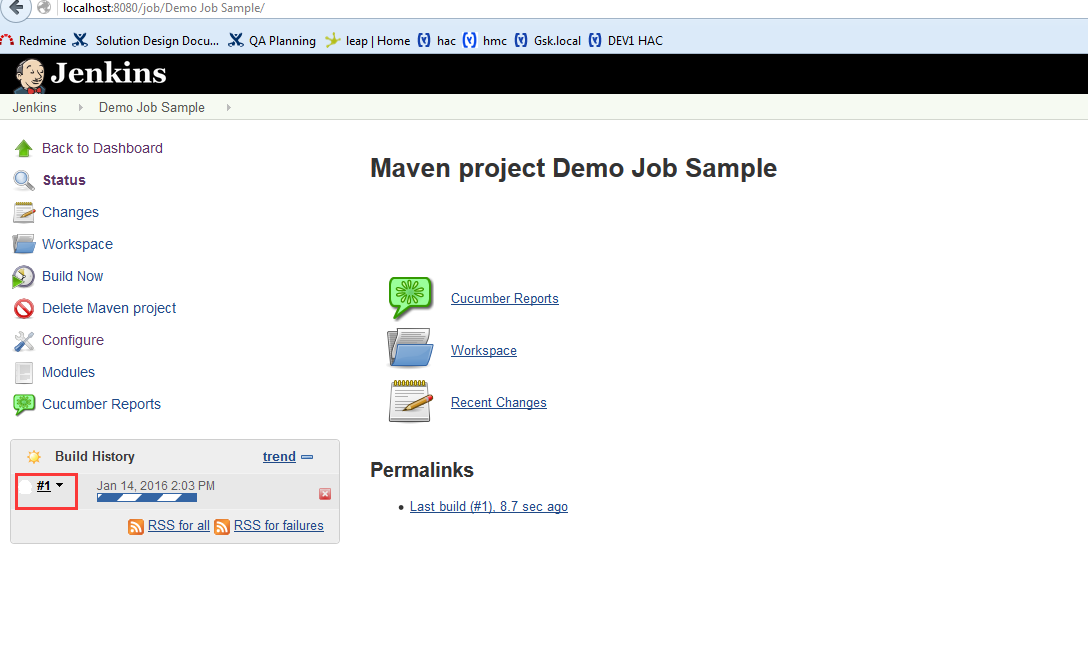




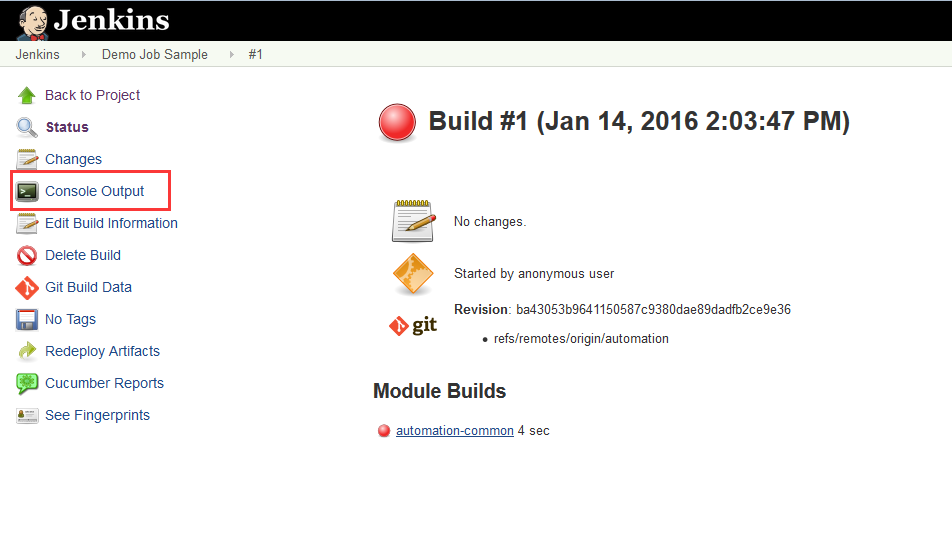
# Jenkins Job的执行

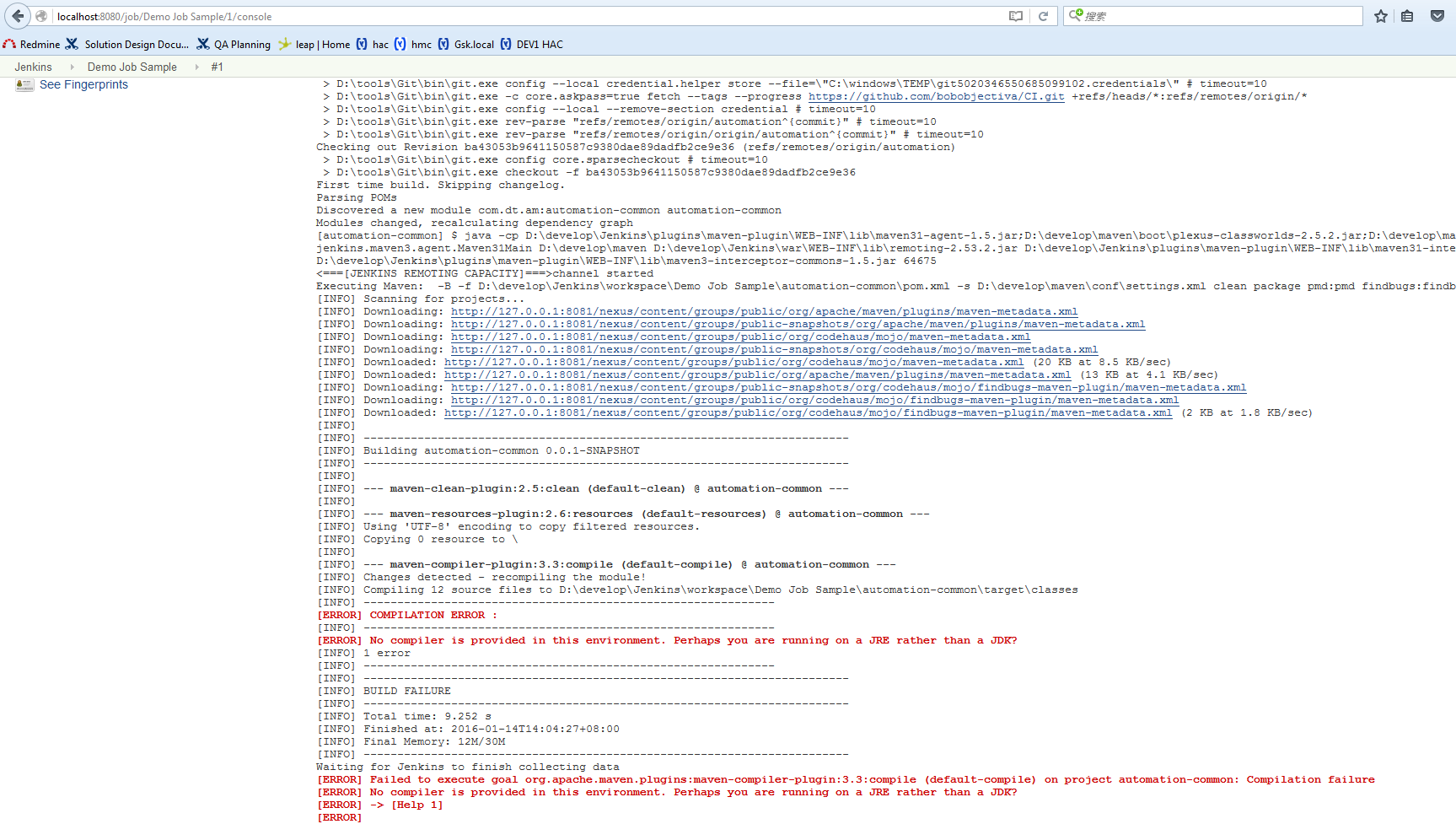
完成上述步骤后，可以启动Jenkins Job查看执行情况。 选择Demo Job Sample，点击Build Now。这时在Build History部分可以看到执行情况。



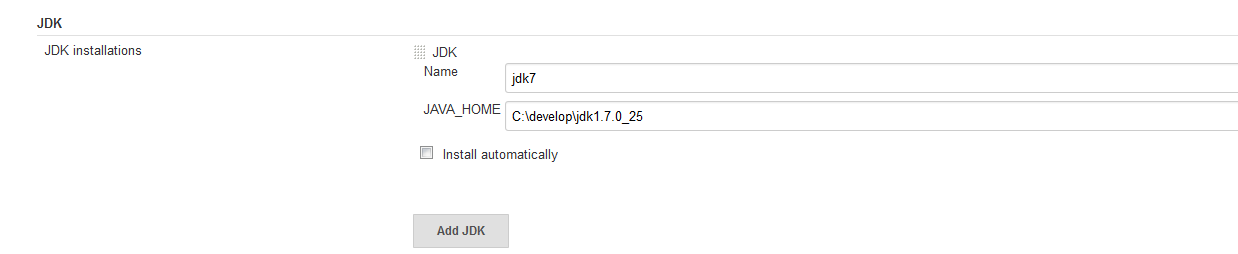


选择Console Output，可以查看到当前任务执行的详细信息。通过任务输入信息，可以定位由于JDK相关设置问题引发了这次执行异常。

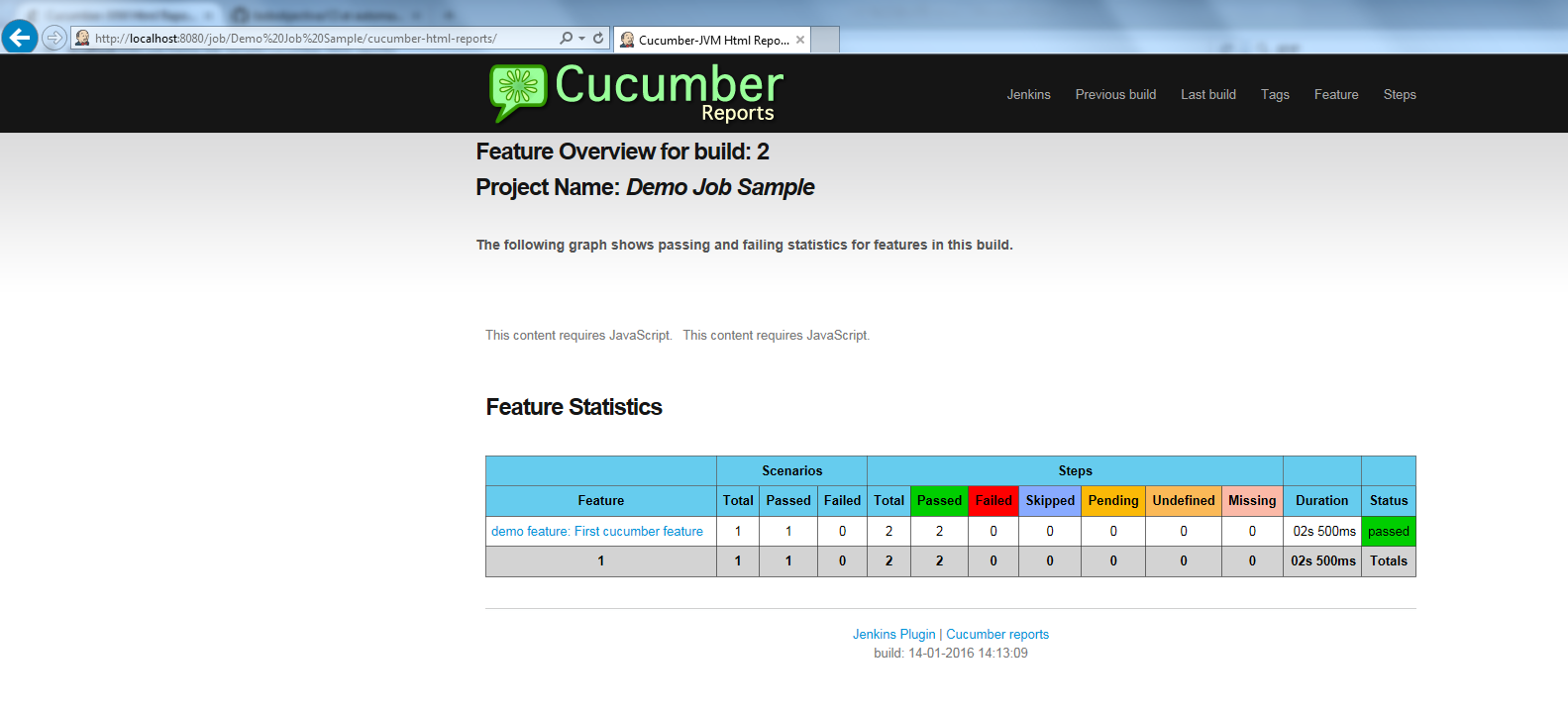




选择Manage Jenkins——Configure Jenkins修改JDK设置。再次执行Job，通过查看任务执行信息，可以看到Cucumber的自动化测试Case已经正确执行。并且Cucumber的执行结果可以通过点击Cucumber Reports能够查看(由于浏览器安全限制，报告的样式不能正确加载，需要后续服务器验证)。







# PMD+FindBugs

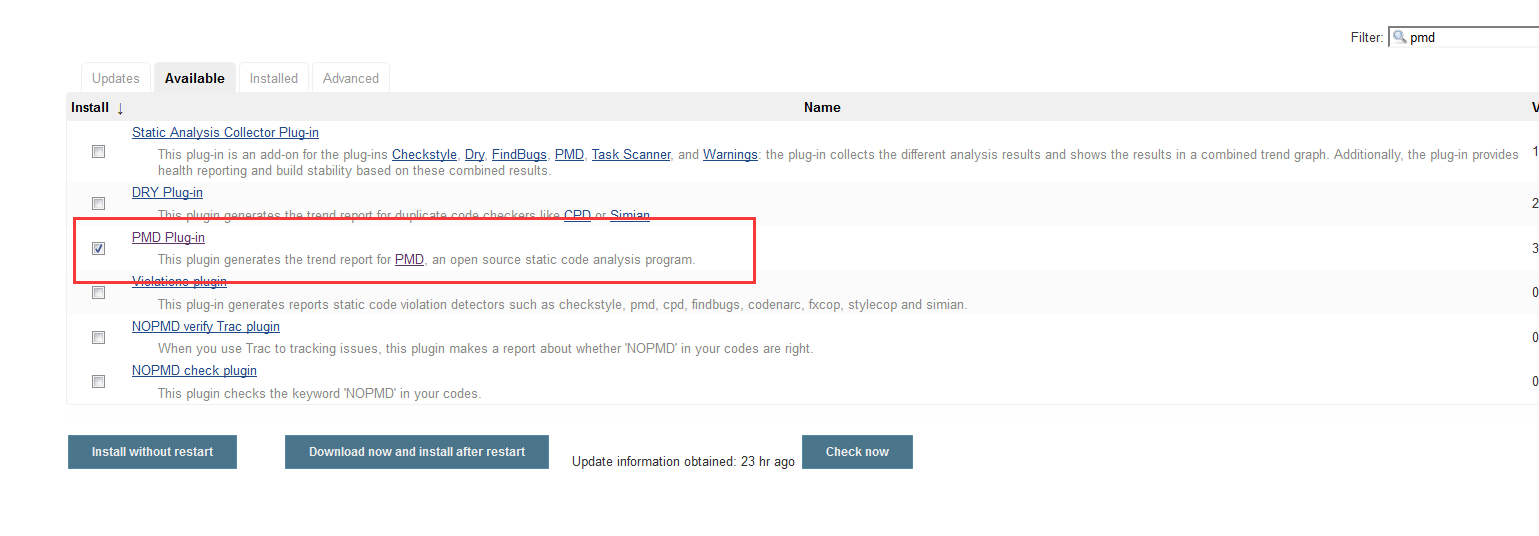
## 演示代码添加PMD及Findbugs插件

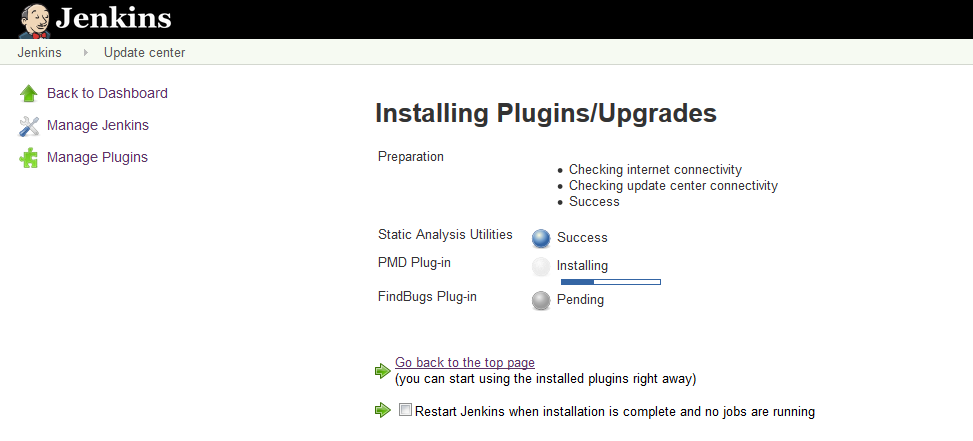
演示代码由Maven，为了支持静态代码检查，需要在演示代码的POM文件添加静态代码检查插件。详情见演示代码。

|  |
| --- |
| <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-pmd-plugin</artifactId>  <version>3.6</version>  <configuration>  <sourceEncoding>UTF-8</sourceEncoding>  <rulesets>  <ruleset>D:\ruleset.xml</ruleset>  </rulesets>  </configuration>  </plugin>  <plugin>  <groupId>org.codehaus.mojo</groupId>  <artifactId>findbugs-maven-plugin</artifactId>  <version>3.0.1</version>  </plugin> |

## 安装Pmd及FindBugs的Jenkins插件

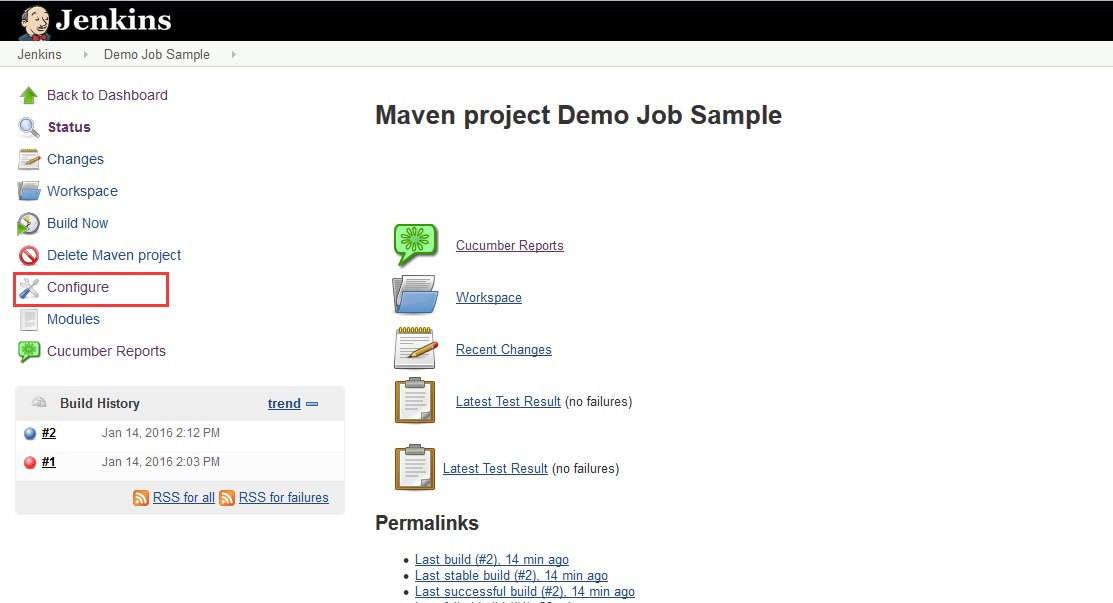
Jenkins插件安装方式同1.3，这里不再详细介绍。

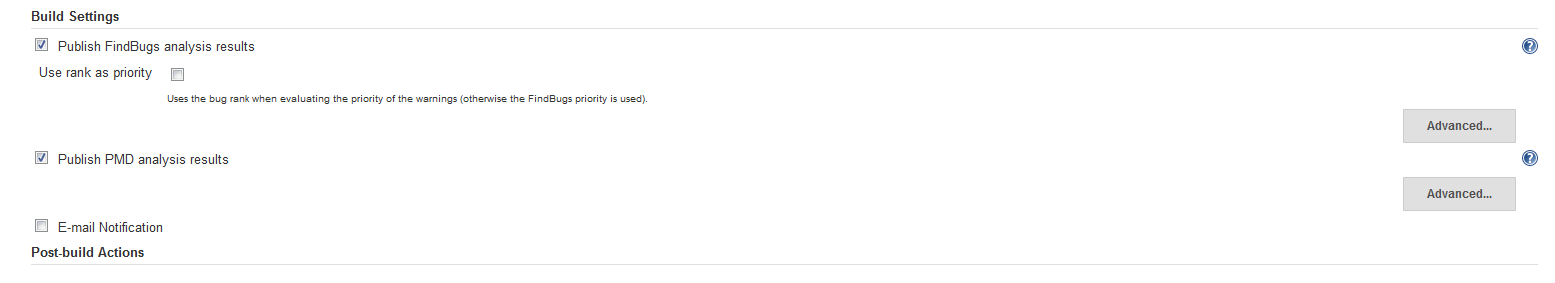




## 更新Demo Job配置

选择Configure更新Job的配置。在Build Settings勾选Publish FindBugs analysis results和Publish PMD analysis results，更多配置可以点击Advanced…按钮设置。保存配置，再次执行任务。





## 静态代码检查结果的查看

执行完成后，左侧选项中可以查看静态代码检查的详细结果。右侧显示了静态代码检查结果的趋势。选择PMD Warnings，可以查看详细的PMD检查结果。

