**网站自动化测试**

**Jenkins部署与迁移**

|  |  |  |  |
| --- | --- | --- | --- |
| 文档状态 | [√] 初稿 | 当前版本 |  |
| [ ] 评审通过 |
| [ ] 修改 |
| [ ] 发布 |
| [ ] 作废 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **修订内容描述** | **作者** | **审核人** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**修 订 记 录**

**目 录**

[一、 概述 3](#_Toc490221462)

[二、 Jenkins部署 3](#_Toc490221463)

[1.1. 部署准备 3](#_Toc490221464)

[1.2. 插件安装 4](#_Toc490221465)

[1.3. 配置Jenkins Credentials 5](#_Toc490221466)

[1.3.1. Add domain 6](#_Toc490221467)

[1.3.2. Add Credentials 6](#_Toc490221468)

[1.4. 系统设置 7](#_Toc490221469)

[1.4.1. Maven项目配置 7](#_Toc490221470)

[1.4.2. Tool locations 7](#_Toc490221471)

[1.4.3. Jenkins location 8](#_Toc490221472)

[1.5. 管理节点 8](#_Toc490221473)

[1.5.1. 新建节点 8](#_Toc490221474)

[1.5.2. 节点登录并启动jenkins client服务 9](#_Toc490221475)

[1.6. Job创建 10](#_Toc490221476)

[1.6.1. 创建maven job 10](#_Toc490221477)

[1.6.2. 创建multi Job 13](#_Toc490221478)

[三、 Jenkins迁移 15](#_Toc490221479)

[2.1. 迁移准备（和部署准备一致） 15](#_Toc490221481)

[2.2. 数据迁移 15](#_Toc490221482)

# 概述

Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with a Java Runtime Environment (JRE) installed.

# Jenkins部署

## 部署环境：

* OS：Win7
* RAM：8G
* 显示器：1920 \* 1080
* 浏览器：Chrome v49~51
* Java：Jdk v1.7
* Maven：v3.39
* Tomcat：v8.5（仅master）
* Jenkins：v2.46（仅master）

## 部署准备

* + - Java已正确安装部署
    - 下载tomcat bin文件，放在指定目录，例如d:\tomcat8
    - 设置tomcat环境变量%CATALINA\_HOME%， 例如CATALINA\_HOME= d:\tomcat8

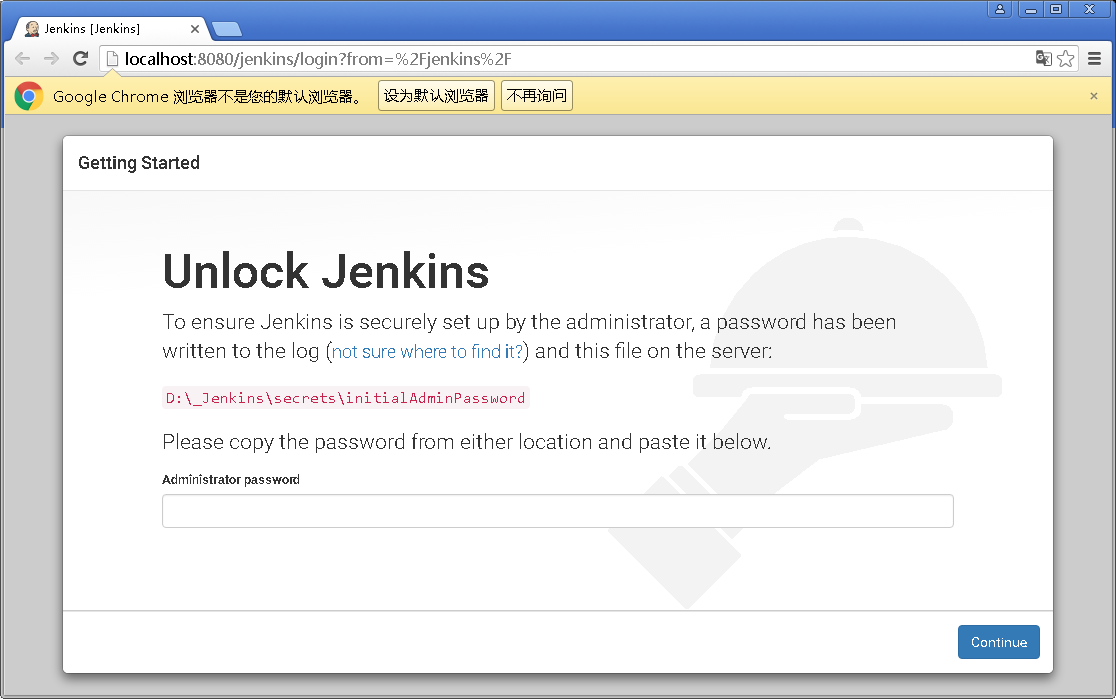


* + - 下载Jenkins.war，并放到%CATALINA\_HOME%\webapps
    - 设置jenkins环境变量%JENKINS\_HOME%，例如JENKINS\_HOME= D:\autoTest\_Jenkins



## 插件安装

* 在%CATALINA\_HOME%\bin目录下,双击startup.bat启动jenkins服务,注意jenkins初始密码路径
* Jenkins服务正常启动后,在浏览器中输入<http://localhost:8080/jenkins/>,打开jenkins

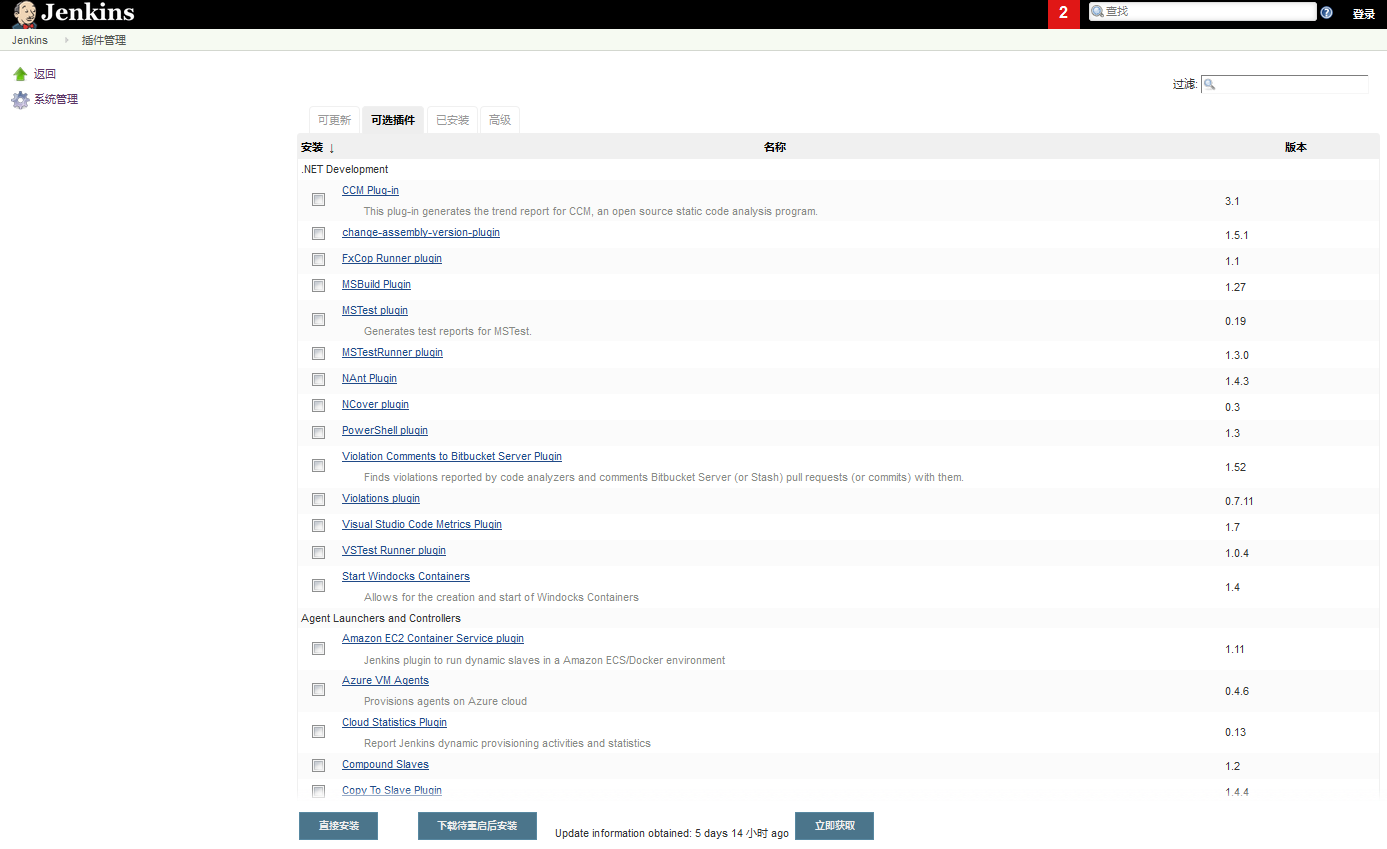


输入初始密码,点击continue按钮

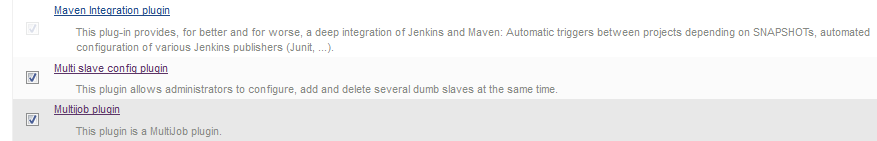
* 默认选择建议安装的所有插件（根据实际网络环境，可能需要首先配置上网代理），等待所有插件安装成功,进入jenkins主界面



* 默认插件安装成功后，修改admin账户的密码，其原始密码即为上面初始密码，改为自定义的密码以便于记
* 进入jenkins系统管理->插件管理->可选插件，等待所有可选插件加载完成，如下图



* 选择安装multiJob、maven插件



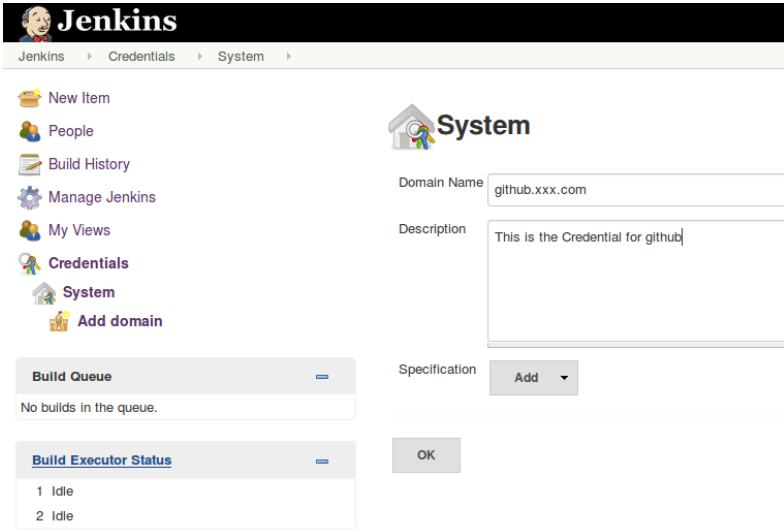
至此，项目需要的插件已完成安装

## 配置Jenkins Credentials

为配合job执行最新代码，在配置源码管理git服务时，除了需要git的SSH地之外，也需要Jenkins Credentials，现进行如下配置。

### Add domain

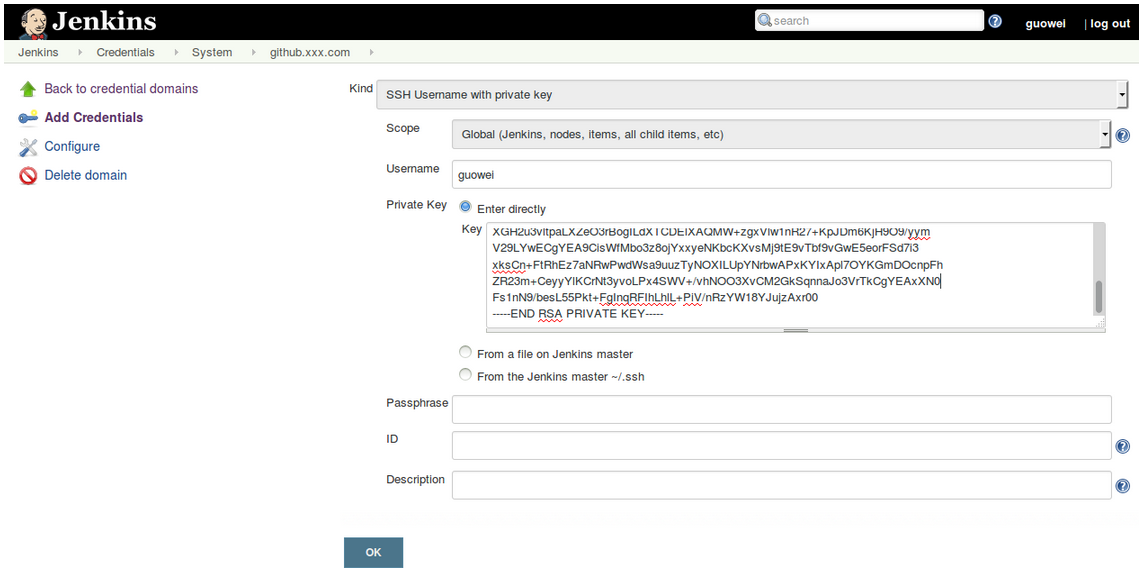
* 在jenkins界面，依次点击： Credentials -> System -> Add domain：   
  Domain Name: 填写你git服务器的地址，如 github.xxx.com   
  Description: 随便写一点描述，如 This is the Credential for github如图所示：



* 点击 ok 后，再点击 “adding some credentials?”

### Add Credentials

* 选择 SSH Username with private key：
* Username: jenkins，以便在创建 Job 的时候使用该 Credential   
  Private Key：选择Enter directly，copy git私钥到Key文本框，如下图：



* 点击 ok 后 Credential 创建成功。

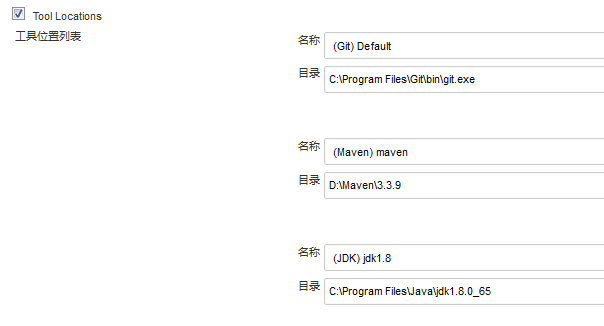
## 系统设置C:\Users\panyongjun\AppData\Roaming\feiq\RichOle\204883002.bmp

### Maven项目配置

C:\Users\panyongjun\AppData\Roaming\feiq\RichOle\901729755.bmp

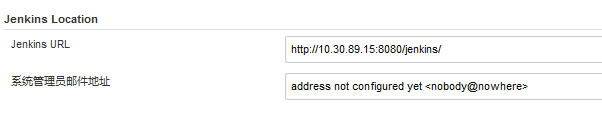
除maven\_opts之外，其他为默认

### Tool locations



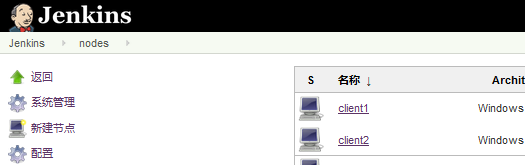
Git、Maven、JDK根据实际路径配置

### Jenkins location



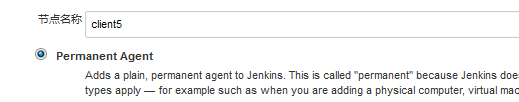
默认jenkins URL 为<http://localhost:8080/jenkins/>，根据实际IP修改

## 管理节点C:\Users\panyongjun\AppData\Roaming\feiq\RichOle\4102203585.bmp



### 新建节点

* 点击新建节点，如下



输入节点名，并选择permanent agent，点击ok按钮

* 进入节点信息设置，设置如下：

# of executors: 1

远程工作目录：c:\jenkins

标签：testClient

启动方法：launch agent via Java Web Start

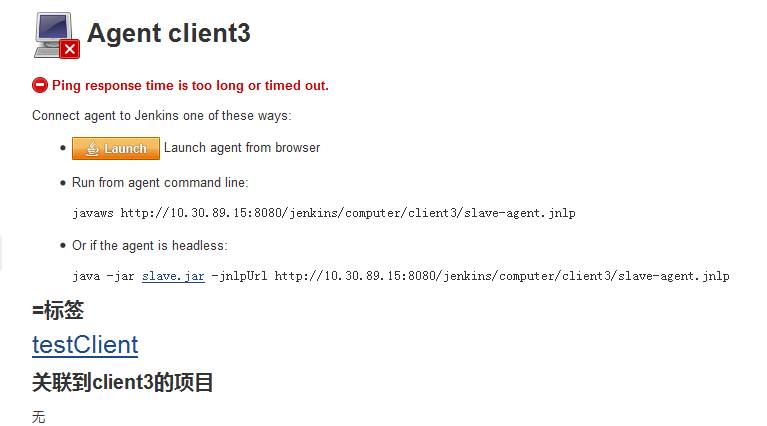
在node property中勾选tool locations，根据节点实际情况，配置git、maven、JDK安装路径，参考上面的系统设置



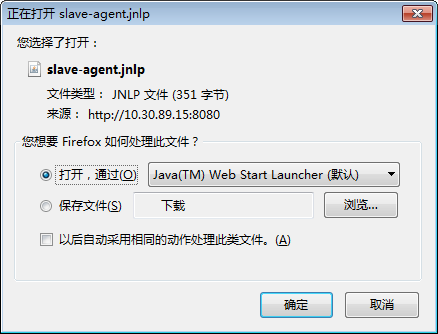
### 节点登录并启动jenkins client服务

* 在浏览器中输入

jenkins Url <http://localhost:8080/jenkins>，进入系统管理->管理节点，点击上面配置的节点，如下



* 点击launch，弹出如下弹框，点击确定，启动client jenkins服务



* 节点jenkins服务正常启动后如下：



## Job创建

在网站自动化项目中，需要创建一个multi job以及22个maven job，maven job为multi job的子job。各子job之间的关系并行，无依赖关系。

### 创建maven job

以新建“专线系统”为例：

* 在浏览器中输入

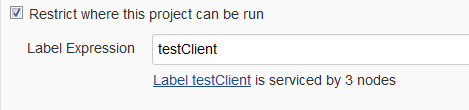
jenkins Url <http://localhost:8080/jenkins>，点新建按钮



* 输入有效的job名称，并选择 “构建一个maven项目”，点击ok按钮



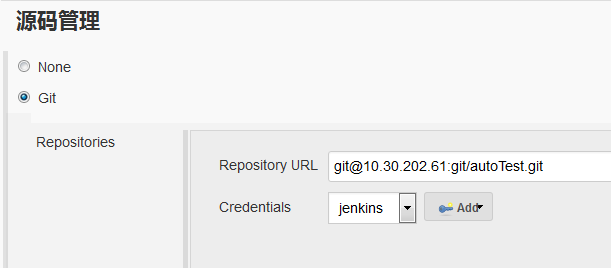
* 在general中勾选“丢弃旧的构建”，设置构建天数、构建最大个数、发布包保留天数、发布包最大保留个数，皆为3
* 在general中勾选“Restrict where this project can be run”，label expression中输入client标签，如下



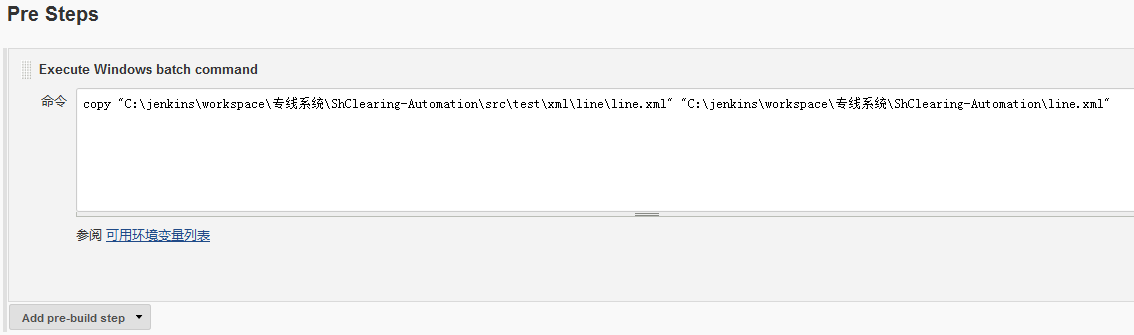
* 在源码管理中，勾选git，输入repository URL地址(git地址)，并选择已配置好的credentials，如下：

Repository [URL: git](URL:git) SSH地址

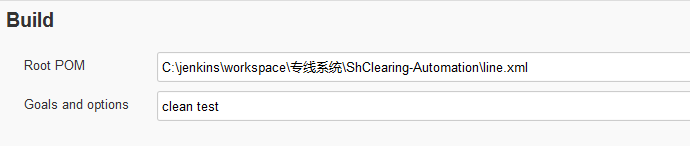
Credentials：即1.3中配置的credentials



* 在pre steps中，以防case变更，添加一条命令行执行pom文件的copy动作，copy到代码主目录中，如下：



* 在build中，设置root pom路径，如下：



Goals and options参数设置

* Package，打包
* Compile，编译
* Test：执行测试
* clean test：清空原编译并执行测试
* 在post steps中，即job执行完毕后，删除copy的pom文件，以防下次job执行时copy异常，如下：



其他为默认，所有maven job与专线系统类似

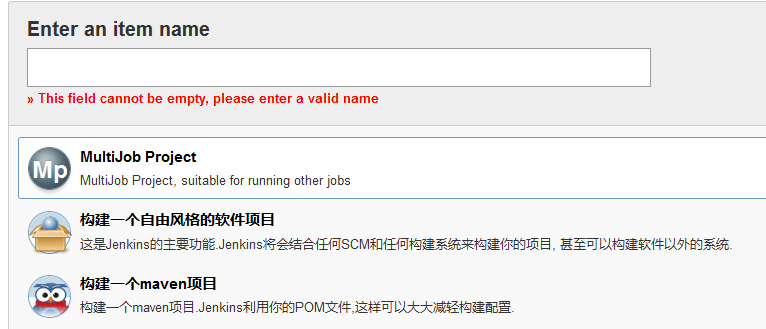
### 创建multi Job

* 在浏览器中输入

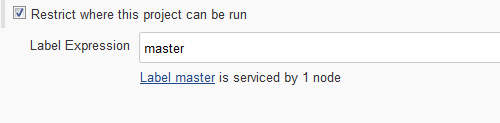
jenkins Url <http://localhost:8080/jenkins>，点新建按钮



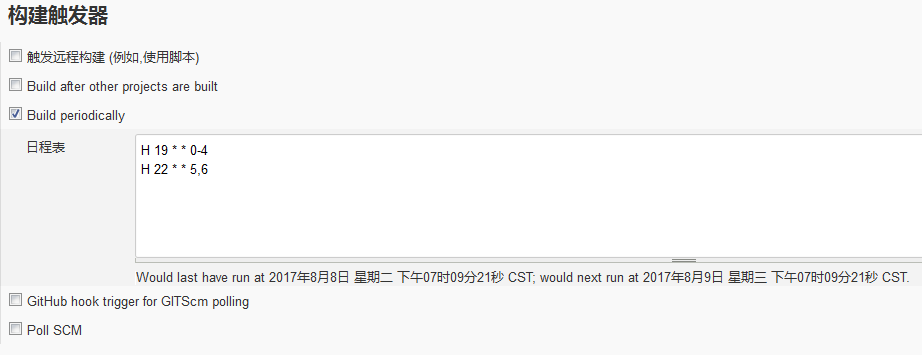
* 输入有效的job名称，并选择 “MultiJob Project”，点击ok按钮



* 在general中勾选“丢弃旧的构建”，设置构建天数、构建最大个数、发布包保留天数、发布包最大保留个数，皆为3
* 在general中勾选“Restrict where this project can be run”，label expression中输入master标签，如下



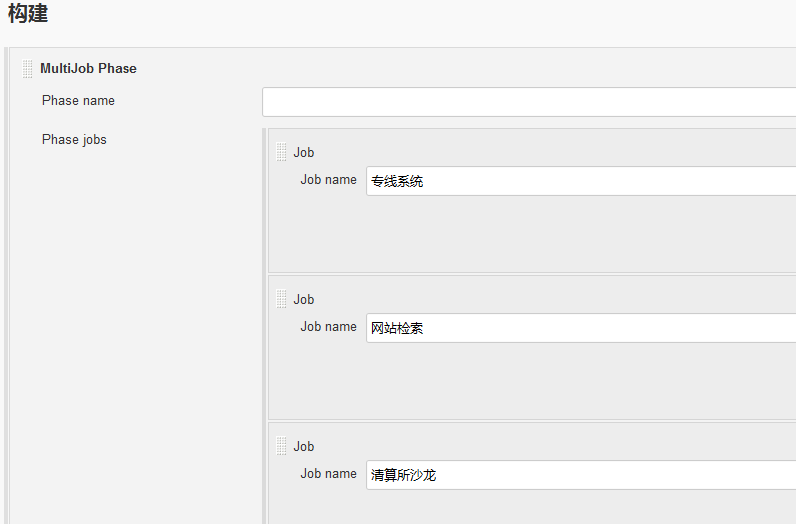
* 在构建触发器中，选择“build periodically”，构建定时job，如下：



0-4：周日到周四

5,6：周五、周六

* 在构建中，添加构建步骤“multijob phrase”，添加已建的maven job，如专线系统等。如下：



点击job下的高级按钮，对job执行时的一些属性进行设置。

* Kill the phrase on：Never (ignore the job result and continue the phase execution)
* Abort all other job：取消勾选

其他为默认

# Jenkins迁移



## 迁移准备（和部署准备一致）

## 数据迁移

* + - 在原有jenkins机器上，将%JENKINS\_HOME%目录下的所有文件做成一个压缩包
    - Copy压缩包到新机器的%JENKINS\_HOME%目录，并解压文件到本目录下
    - 在%CATALINA\_HOME%\bin目录下,双击startup.bat启动jenkins服务，
    - 在
    - 浏览器中输入

<http://localhost:8080/jenkins/>，打开

* + - jenkins
    - 参考jenkins部署中的1.4，重新部署节点
    - 构建一个job，若正常执行，数据迁移即全部完成