

文章出处: <http://velep.com/archives/867.html>

一，持续集成和 Jenkins 介绍



对于很多做嵌入式软件的朋友来说，可能从来都没有听说过持续集成这个词，也就无从接触过 **Jenkins** 这个工具软件。自去年转调部门后，除了解了设计模式外，还有幸接触到了 **Jenkins**。

那什么是持续集成？**Jenkins** 具体用来做什么，对软件开发有什么益处呢？总得来说，这两者主要是涉及一个软件质量的主体，特别是团队开发软件项目。下面就来介绍介绍下这两者。

一，持续集成的概念

持续集成，Continuous integration，简称 CI。

随着软件开发复杂度的不断提高，团队开发成员间如何更好地协同工作以确保软件开发的质量已经慢慢成为开发过程中不可回避的问题。尤其是近些年来，敏捷（**Agile**）在软件工程领域越来越红火，如何能再不断变化的需求中快速适应和保证软件的质量也显得尤其的重要。

持续集成正是针对这一类问题的一种软件开发实践。它倡导团队开发成员必须经常集成他们的工作，甚至每天都可能发生多次集成。而每次的集成都是通过自动化的构建来验证，包括自动编译、发布和测试，从而尽快地发现集成错误，让团队能够更快的开发内聚的软件。

以我经过的项目（假设为 A 项目）为例进行描述。

首先，解释下集成。我们所有项目的代码都是托管在 **SVN** 服务器上。每个项目都要有若干个单元测试，并有一个所谓集成测试。所谓集成测试就是把所有的单元测试跑一遍以及其它一些能自动完成的测试。只有在本地电脑上通过了集成测试的代码才能上传到 **SVN** 服务器上，保证上传的代码没有问题。所以，集成指集成测试。

再说持续。不言而喻，就是指长期的对项目代码进行集成测试。既然是长期，那肯定是自动执行的，否则，人工执行则没有保证，而且耗人力。对此，我们有一台服务器，它会定期的从 **SVN** 中检出代码，并编译，然后跑集成测试。每次集成测试结果都会记录在案。完成这方面工作的就是下面要介绍的 **Jenkins** 软件。当然，它的功能远不止这些。在我们的项目中，执行这个工作的周期是 1 天。也就是，服务器每 1 天都会准时地对 **SVN** 服务器上的最新代码自动进行一次集成测试。

二，持续集成的特点

- 它是一个自动化的周期性的集成测试过程，从检出代码、编译构建、运行测试、结果记录、测试统计等都是自动完成的，无需人工干预；
- 需要有专门的集成服务器来执行集成构建；
- 需要有代码托管工具支持；

三，持续集成的作用

- 保证团队开发人员提交代码的质量，减轻了软件发布时的压力；
- 持续集成中的任何一个环节都是自动完成的，无需太多的人工干预，有利于减少重复过程以节省时间、费用和工作量；

上面我们了解了持续集成的知识。既然有这么多的好处，那我们怎么样实现它呢？这就是接下来要介绍的名角：Jenkins 软件。



Jenkins

贰， Jenkins

一， Jenkins 介绍

Jenkins，原名 Hudson，2011 年改为现在的名字，它 是一个开源的实现持续集成的软件工具。官方网站：<http://jenkins-ci.org/>。

Jenkins 能实施监控集成中存在的错误，提供详细的日志文件和提醒功能，还能用图表的形式形象地展示项目构建的趋势和稳定性。

目前持续集成(CI)已成为当前许多软件开发团队在整个软件开发生命周期内侧重于保证代码质量的常见做法。它是一种实践，旨在缓和和稳固软件的构建过程。并且能够帮助您的开发团队应对如下挑战：

- 软件构建自动化：配置完成后，CI 系统会依照预先制定的时间表，或者针对某一特定事件，对目标软件进行构建。

- 构建可持续的自动化检查：CI 系统能持续地获取新增或修改后签入的源代码，也就是说，当软件开发团队需要周期性的检查新增或修改后的代码时，CI 系统会不断确认这些新代码是否破坏了原有软件的成功构建。这减少了开发者们在检查彼此相互依存的代码中变化情况需要花费的时间和精力(说直接一点也是钱啊，呵呵)。
- 构建可持续的自动化测试：构建检查的扩展部分，构建后执行预先制定的一套测试规则，完成后触发通知(Email,RSS 等等)给相关的当事人。
- 生成后后续过程的自动化：当自动化检查和测试成功完成，软件构建的周期中可能也需要一些额外的任务，诸如生成文档、打包软件、部署构件到一个运行环境或者软件仓库。这样，构件才能更迅速地提供给用户使用。

部署一个 CI 系统需要的最低要求是，一个可获取的源代码的仓库，一个包含构建脚本的项目。

二，Jenkins 特点

- 易安装：仅仅一个 `java -jar jenkins.war`，从官网下载该文件后，直接运行，无需额外的安装，更无需安装数据库；
- 易配置：提供友好的 GUI 配置界面；
- 变更支持：Jenkins 能从代码仓库（Subversion/CVS）中获取并产生代码更新列表并输出到编译输出信息中；
- 支持永久链接：用户是通过 web 来访问 Jenkins 的，而这些 web 页面的链接地址都是永久链接地址，因此，你可以在各种文档中直接使用该链接；
- 集成 E-Mail/RSS/IM：当完成一次集成时，可通过这些工具实时告诉你集成结果（据我所知，构建一次集成需要花费一定时间，有了这个功能，你就可以在等待结果过程中，干别的事情）；
- JUnit/TestNG 测试报告：也就是用以图表等形式提供详细的测试报表功能；
- 支持分布式构建：Jenkins 可以把集成构建等工作分发到多台计算机中完成；
- 文件指纹信息：Jenkins 会保存哪次集成构建产生了哪些 jars 文件，哪一次集成构建使用了哪个版本的 jars 文件等构建记录；
- 支持第三方插件：使得 Jenkins 变得越来越强大；

三，其它集成工具

其它比较著名的持续集成工具有：CruiseControl, TeamCity, Continuum 等。

四，嵌入式软件集成

据我了解，在嵌入式软件开发中，很少有人有用到持续集成工具。个人觉得最主要的原因是：嵌入式软件与硬件联系比较紧密，很多时候难以满足持续集成的条件——构建自动化测试。

但我们还是可以有所作为。在设计应用软件时，把逻辑业务与硬件相关功能区分开来，对逻辑业务部分编写单元测试，然后做集成测试。

当然，对于小型的嵌入式应用软件就没必要做这个集成工作了。

二，部署与安装



事实上之前是做过一些简单的文档总结的，但是太扣住“简单”这个主题了，因此基本上只能是一些辅助理解的东东，并不能算作教程类的。今天，就特别来把教程详细说明写出。

在教程以及文档方面，个人的一些心得就是，这是一个很有悖论的事情，教程肯定都是会了的人写出来的，而会了的人好像又不需要教程（注意是好像，因为遗忘之后又会需要），因此会了的人写的教程往往都是简之又简，不会的人往往看了会了的人写的教程，则蒙之又蒙。

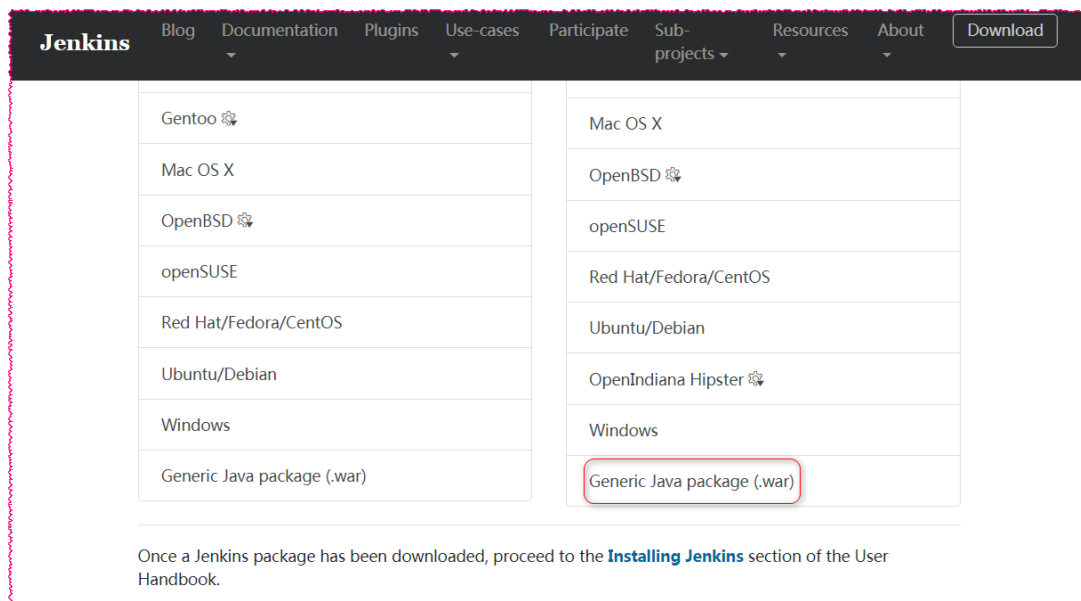
因此，当我愿意认真写文档的时候，常常喜欢把自己当成白痴一样的来把教程写给自己，可能会耗费多一些心力，也会有一些啰嗦繁多，但是可以真正让人受益，又有什么可与此相比呢。

Jenkins 官方网站: <http://jenkins-ci.org/>。



由于其安装包比较容易在官方进行下载，所以就不再繁琐的在百度云留存了。

下载也非常简单，点击大大的 [Download](#)

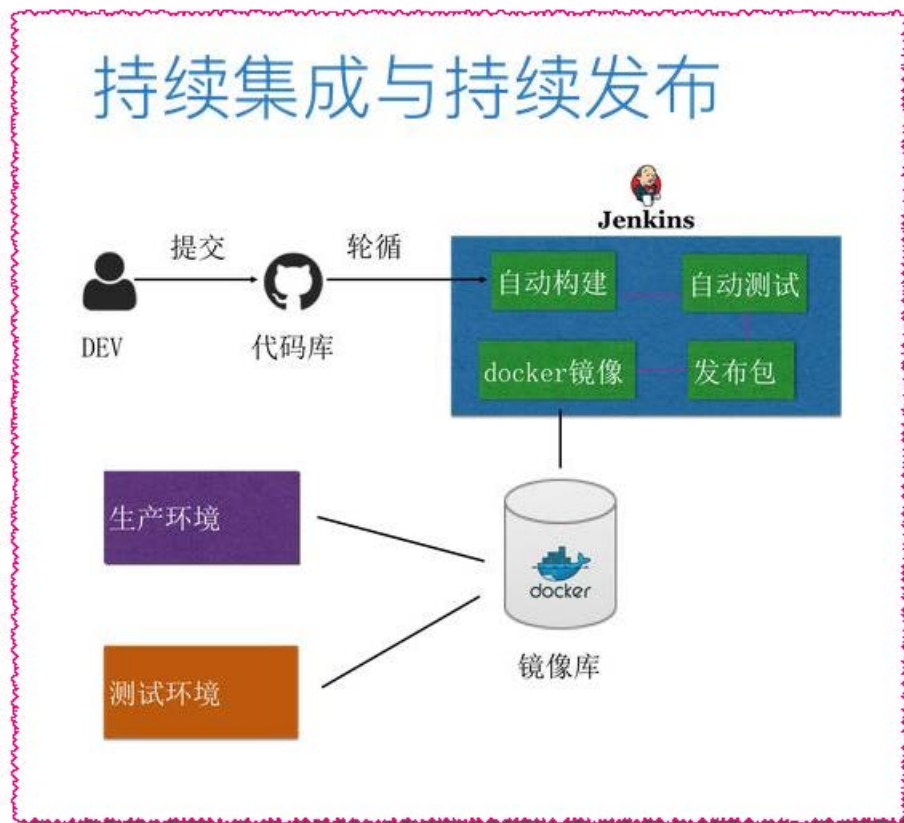


左边是半年更新一次的稳定版，右边是与小米系统更新频率一样（每周一更新）的开发版本，用起来都差不多。所以选择右边最下边的 **war** 包的方式下载，这样部署简单，管理方便，两全其美。

一般直接在浏览器下载速度也不算慢，如果嫌慢，可以用迅雷进行下载，或者使用阿里云服务器复制链接进行 **wget** 下载，这里不多赘述了。

架构简析

Jenkins 持续构建部署工具是有一套完整的架构体系的。如图所示（注，图片来自网络）：



介绍：

- 1, 需要用到的软件工具。
代码库 (SVN/Git), Jenkins, 如果是 Java 项目, 则还可能用到 nexus (maven 的私服仓库), maven (打包工具)。
- 2, 整套的工作流程是: 开发人员进行项目开发----提交代码到代码库----在 Jenkins 上进行部署构建----原本繁琐的一系列操作, 就这么被 Jenkins 给一条龙服务了。
- 3, 这是网上的图, 一般如果直接使用, 则可以忽略掉 docker 镜像和镜像库两步, 直接将发布包部署到相对应的服务器上就 ok 了。
- 4, 一般公司至少都是有测试与生产环境两种 (或者还会有预发的环境), 两种环境最好分开部署两个 Jenkins, 一方面利于权限管理, 另一方面避免掉项目过多的误操作。我曾听一个同事说他们之前公司没有测试环境, 直接拿线上当测试, 每次发布, 所有人都像如临大敌一样瞪大双眼监控着有没有问题,, , 想想就可怕哈。

部署开始。

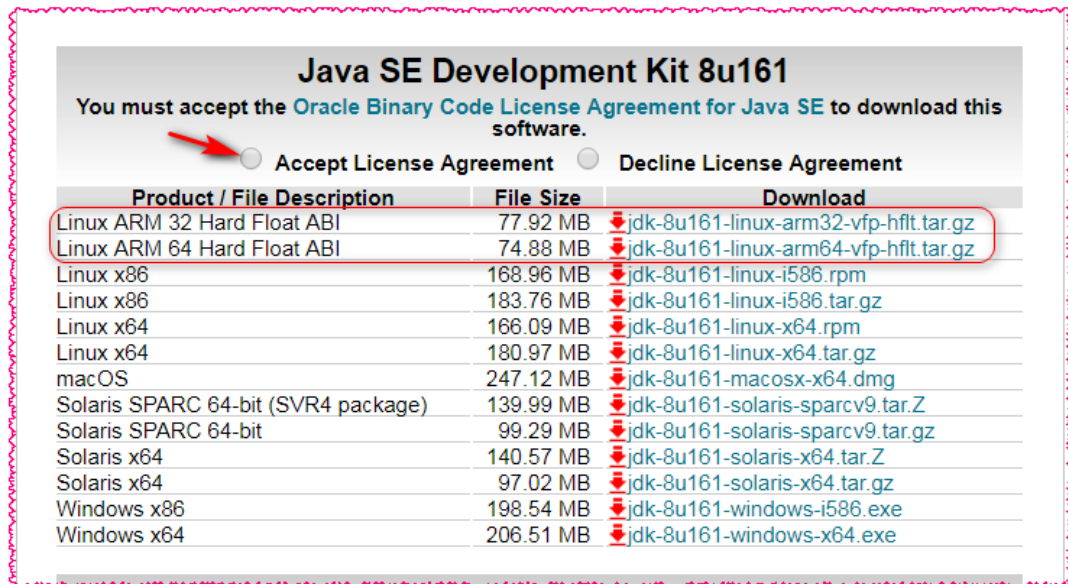
总览:

- 1, 部署 jdk。
- 2, 部署 tomcat。
- 3, 部署 Jenkins。
- 4, 配置 Jenkins。

部署 jdk。

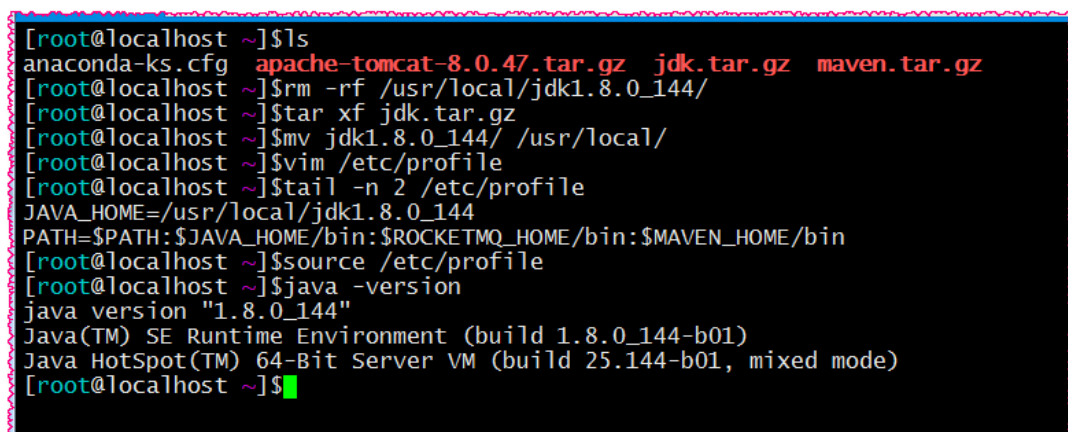
下载地址: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

最新的版本已经是 1.9, 但是这里使用 1.8 版本的。注意下载的时候选中一下接受许可协议。然后找到对应版本进行下载。



安装步骤是: 解压----移动到对应目录下----写入环境变量-----加载一下----验证。

```
[root@localhost ~]$tar xf jdk.tar.gz
[root@localhost ~]$mv jdk1.8.0_144/ /usr/local/
[root@localhost ~]$vim /etc/profile
[root@localhost ~]$tail -n 2 /etc/profile
JAVA_HOME=/usr/local/jdk1.8.0_144
PATH=$PATH:$JAVA_HOME/bin:$ROCKETMQ_HOME/bin:$MAVEN_HOME/bin
[root@localhost ~]$source /etc/profile
[root@localhost ~]$java -version
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)
```




```
[root@localhost ~]$mv apache-tomcat-8.0.47 /usr/local/tomcat
```

```
[root@localhost webapps]$../bin/startup.sh && tailf -n 500 ../logs/catalina.out
```

清空 webapps 目录下所有内容，导入 Jenkins 的 war 包，改名成默认的包名。启动，并监控启动日志的输出。

```

my
07-Mar-2018 17:01:32.329 信息 [Jenkins initialization thread] jenkins.install.SetupWizard.init
*****
*****
*****
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

ed7fbe822bcd4dec916d1dfdbf22f403

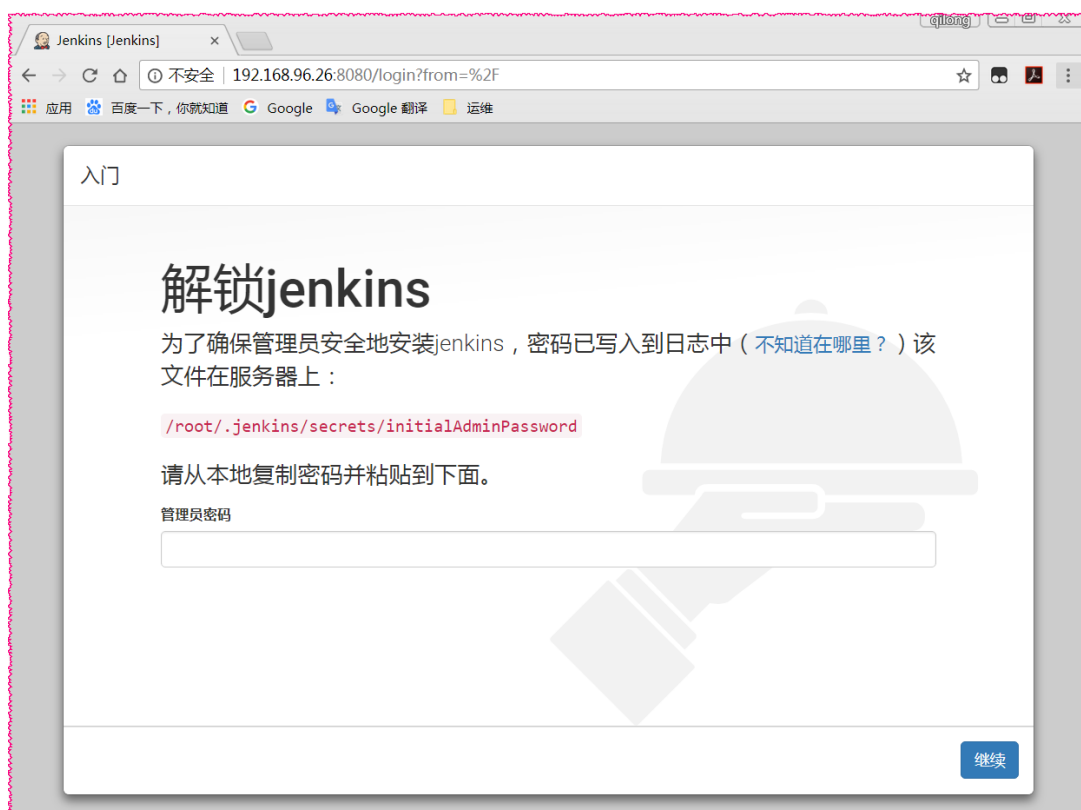
This may also be found at: /root/.jenkins/secrets/initialAdminPassword

*****
*****
*****

```

大概一两分钟后，可以看到日志输出，Jenkins 已经初始化安装成功，并且给出了管理员 admin 的 登 陆 密 码 为 ed7fbe822bcd4dec916d1dfdbf22f403，当然也可以在 `/root/.jenkins/secrets/initialAdminPassword` 文件中 cat 到。

接下来访问：192.168.96.26:8080



刚好，将刚才的密码填进去咯。然后稍候：



现在新版本的 Jenkins 已经对中文的支持非常全面了，所以使用的门槛也是越来越低，中文看起来到底还是要比外文亲切的呀。

插件在 Jenkins 中的地位非常重要，详细的在后边进行表述，这个地方就选择人家推荐

的进行安装。毕竟新手就暂时顺应安排咯。

新手入门

新手入门

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	⚙ Credentials Binding	** Script Security ** Command Agent Launcher
⚙ Timestamper	⚙ Workspace Cleanup	⚙ Ant	⚙ Gradle	Folders
⚙ Pipeline	⚙ GitHub Branch Source	⚙ Pipeline: GitHub Groovy Libraries	⚙ Pipeline: Stage View	** bouncecastle API ** Structs ** Pipeline: Step API ** SCM API ** Pipeline: API ** JUnit
⚙ Git	⚙ Subversion	⚙ SSH Slaves	⚙ Matrix Authorization Strategy	OWASP Markup Formatter ** Pipeline: Supporting APIs ** Pipeline: Job ** Token Macro
⚙ PAM Authentication	⚙ LDAP	⚙ Email Extension	⚙ Mailer	Build Timeout ** Credentials

** - 需要依赖

Jenkins 2.109

静静等待，安装过程略慢。

新手入门

创建第一个管理员用户

用户名:

密码:

确认密码:

全名:

电子邮件地址:

Jenkins 2.109

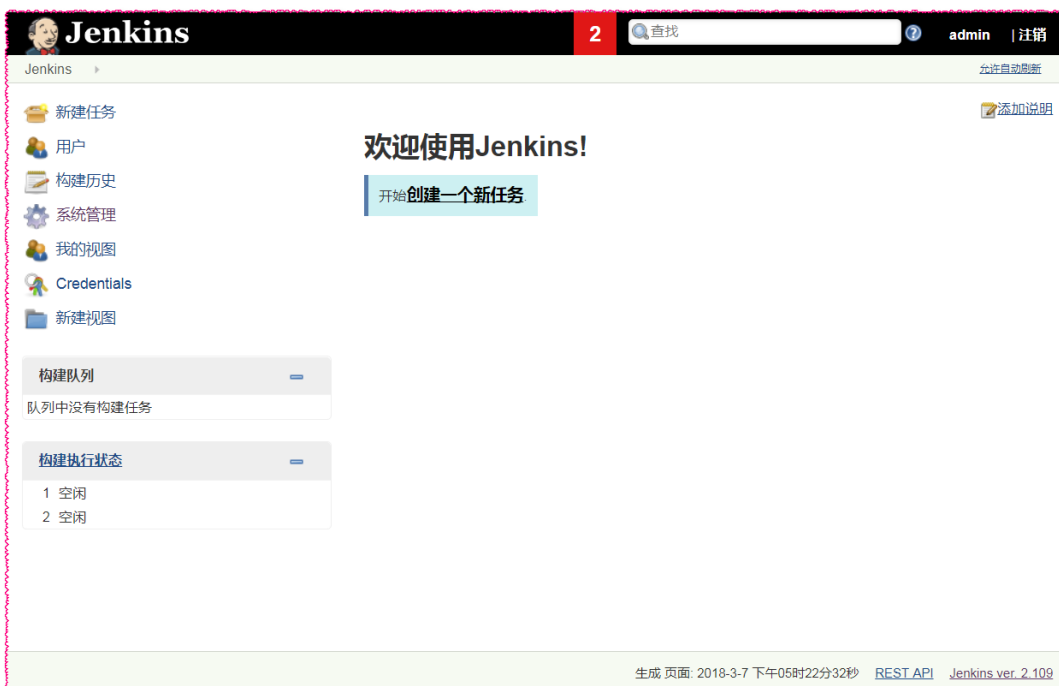
使用admin账户继续

保存并完成

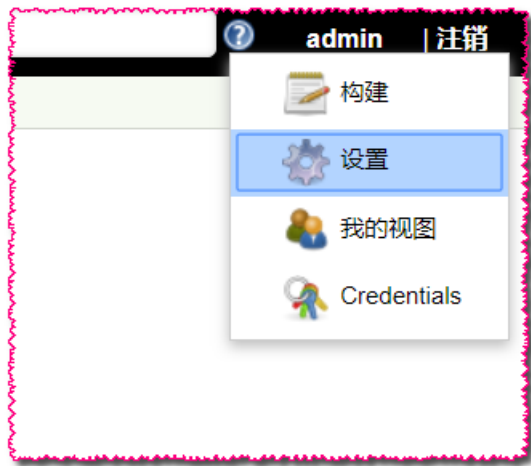
新手总算入门了，这个地方不多设置，直接使用 admin 账号继续登入：



等下登陆之后改一下密码就好了。



先修改 admin 的密码



点进去有惊喜，具体怎么改，不说了。

到这里安装基本上就完成了，详细选项的介绍以及配置，容后详述。

三，插件介绍

允许自动刷新

Go to plugin manager

Configure which of these warnings are shown



[系统设置](#)
全局设置&路径



[Configure Global Security](#)
Secure Jenkins; define who is allowed to access/use the system.



[Configure Credentials](#)
Configure the credential providers and types



[Global Tool Configuration](#)
Configure tools, their locations and automatic installers.



[读取设置](#)
放弃当前内存中所有的设置信息并从配置文件中重新读取仅用于当您手动修改配置文件时重新读取设置。



[管理插件](#)
添加、删除、禁用或启用Jenkins功能扩展插件。 (可用更新)



[系统信息](#)
显示系统环境信息以帮助解决问题。



[System Log](#)
系统日志从java.util.logging捕获Jenkins相关的日志信息。



[负载统计](#)
检查您的资源利用情况，看看是否需要更多的计算机来帮助您构建。



[Jenkins CLI](#)
从您命令行或脚本访问或管理您的Jenkins。



[脚本命令行](#)
执行用于管理或故障探测或诊断的任意脚本命令。



[管理节点](#)
添加、删除、控制和监视系统运行任务的节点。



[Manage and Assign Roles](#)
Handle permissions by creating roles and assigning them to users/groups



[About Jenkins](#)
See the version and license information.



[Manage Old Data](#)
Scrub configuration files to remove remnants from old plugins and earlier versions.



[管理用户](#)
创建/删除/修改Jenkins用户



[In-process Script Approval](#)
Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions.



[准备关机](#)
停止执行新的构建任务以安全关闭计算机。

一些常用的配置解析介绍：

1， 系统设置

此处定义 Jenkins 的全局消息，或者是邮箱的配置。

2， 全局安全配置

主要是配置用户访问的控制，这里仅开启允许用户注册，这样的话，可以让所有开发测试小伙伴在这里以自己的名字注册自己的账号，然后运维管理人员根据他们不同的项目与权限进行规则的设置就行了。



3， 全局工具设置。

里边包含了所有的代码库，或者打包工具的位置，根据情况进行设置就行。或者什么都不设置也可以。只不过以后脚本里边注意添加命令加载变量。

4， 管理插件。可参考：<https://www.w3cschool.cn/jenkins/jenkins-s5xk28l5.html>

里边有丰富的插件提供，详细的插件功能，不多赘述，有几个重要的需要在，

5， ant 打包的，LDAP 就是上边的访问控制功能，Git，从代码库拉取代码。Role-Based Strategy 角色管理。

Email Extension Plugin (邮件通知)

Publish Over SSH (远程 Shell)

Maven Integration plugin (Maven 集成，否则新建的时候没有 Maven Project)

Monitoring (监控 Jenkins 所消耗的系统资源，如内存使用等)

6， 管理节点

如果服务器数量过于庞大，则可能会使用到主从式的集群进行构建发布。

7， 管理用户

这些用户能够登录到 Jenkins。这是 [列表](#) 的子集，也包括那些只是提交了代码到某些项目但是从未登录 Jenkins 而自动创建的用户。这是官方的介绍，点击用户后边的设置，还可以更改他的信息以及密码的重置等。

8, 小黄锁。需要安装插件 Role-Based Strategy

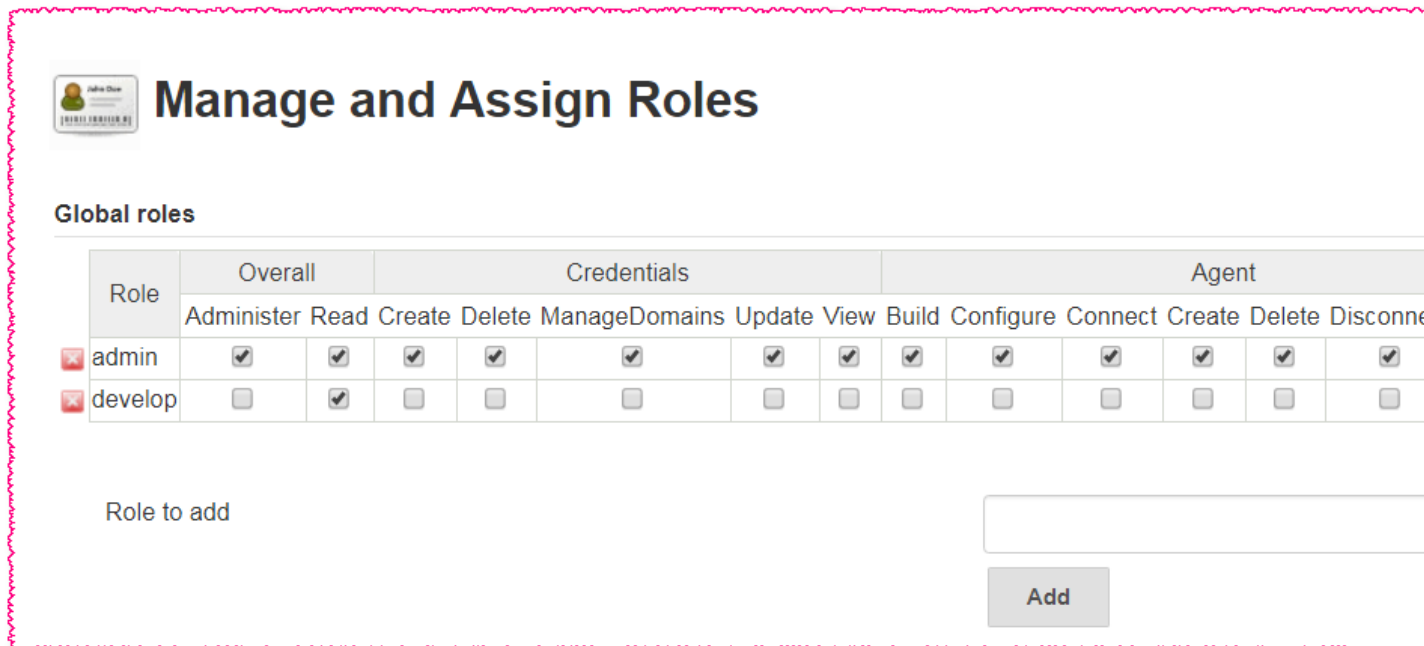
此处可以参考: <https://www.cnblogs.com/wjoyxt/p/6677115.html>

所有的用户分组规则权限都在这里设置了。



管理角色是规则的设置

1, 全局变量的设置，也可以理解成是组的添加设置，这里只设置出了开发者一个组。也可以通过下边的 add 新增。他们对全局有读的权限就好了。



2, 项目的分组。这个地方有点像 Git 里边的权限了，有分组的，也有项目的，经过这样两层的过滤，基本上每个人都会有合适的权限了。

Project roles

	Role	Pattern	Credentials					Job								
			Create	Delete	ManageDomains	Update	View	Build	Cancel	Configure	Create	Delete	Discover	Move	Read	Worksp
<input checked="" type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Role to add


Pattern

Add

Slave roles

分配角色

这里边就是对每个人的权限进行具体的分配了。这里边的又是与上边全局配置的规则对应的，因此先 **add** 把人添加进来，然后在后边勾选合适的权限就可以了。



Assign Roles

Global roles

User/group	admin	develop	
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

注意不同人身份的不同权限设置。

Item roles

	User/group					
		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

现在，基本上对 Jenkins 的基础配置已经完成了，接下来就能够进行项目的构建与部署了。