



链滴

Jenkins 自动化部署持续交付学习

作者: [seanlee](#)

原文链接: <https://ld246.com/article/1515858784686>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

#学习Jenkins的原因

在项目组待了半年了，项目经过两个月的开发后本身在10月份的时候就已整体上线，但是后续小的需变更及bug修复使得在上线后的三个月几乎每周都有更新，而且在几次更新中更是出现了较为严重的产事故，事后自己每每反思原因，除去团队人员代码编写的问题，其余问题多半是要归咎于现在这种动且落后的更新方式上。

现在交付给客户的项目是基于Maven构建的，生产环境就是以war包形式部署在tomcat当中，在应用服务器前置放了几层负载均衡和请求代理转发。每次更新的时候大都是手动打包然后挑出更新的内容来进行增量更新，中间只有一两次是碰到了大的需求变更才全量打了war包丢在服务器tomcat下，每次的产更新都是主要由自己协助客户运维人员进行，所以但凡是出现了问题，自己多多少少都要背些锅。

被更新折磨了这么久，慢慢感悟到交付是一个项目周期中很重要的一环，特别是在面临持续性交付这业务场景时，一定要选择一个好的线上更新方案，这样才能尽可能的减少在交付过程中由于操作人员身操作不当而导致的一些生产事故，所以我在私下开始研究这个Jenkins这个持续交付工具。顺带一句，大四在亚信实习的时候我的岗位就是Java交付工程师，但是入职的时候一直都是做着开发的事，直到现在进了这家公司，在项目大体完成后还一直驻在客户现场做些持续的更新维护和支持，我才是明白什么是交付工程师。下面是自己在尝试摸索使用Jenkins时结合网上资料自己的一些总结。

#Jenkins是什么

jenkins是一个广泛用于持续构建的可视化web工具，持续构建说得更直白点，就是各种项目的"自动化编译、打包、分发部署。jenkins可以很好的支持各种语言（比如：java, c#, php等）的项目构建，也全兼容ant、maven、gradle等多种第三方构建工具，同时跟svn、git能无缝集成，也支持直接与源代码托管网站，比如github、bitbucket直接集成。

Jenkins的主要作用主要为：

1. 持续的软件版本发布/测试项目。
2. 监控外部调用执行的工作。

随着容器技术的发展，Jenkins+docker似乎在现在自动化部署与维护上变得很流行，但是我还没有究这么透彻...

在这里简单介绍以下Jenkins工作的流程：

3. General：主要就是项目的创建和工作空间的选择。
4. 源码管理：选择你项目代码的管理工具，提供svn、git等选项，让jenkins连接到你的代码仓库并且出到你Jenkins选择的工作空间路径。
5. 构建触发器：构建工程的触发器，可以在这一步配置来进行定时构建工程
6. 构建：能够选择并配置你构建的工具，如mavan、ant、gradle等工具，注意Jenkins首先要集成些插件，或者直接使用你服务器上安装好的这些应用，配置以下路径即可。构建完之后便可以生成war包。
7. 构建后操作：这一步可以选择工程构建完之后是否自动发布，如果要自动话部署的话这也是很重要一步。

#安装与运行

Jenkins在Linux的安装主要两种方式，第一种时rpm包安装，第二种则是jar/war直接运行。我使用的在官网下载war包，直接丢在tomcat里运行这种简单的方式，不需要修改配置连接数据库等操作。

#具体配置

Jenkins在第一次登陆的时候需要做一个比较长时间的初始化，需要根据提示先在指定路径下找到一个默认密码，然后重置用户密码安装插件的等等的安装，这个过程较长，完成之后便可以进入Jenkins后台管理的主要界面。

其余

详细教程参考这篇博客吧，[# Jenkins+Maven+SVN自动部署配置文档](#)我也懒得截图写了，大致流程如博客所言，根据自身项目情况做下变动即可