# Jekins基本介绍

## Jenkins简介 -- Jenkins 是一个**可扩展的持续集成引擎**。

1.1 概述

Jenkins是一个开源的、提供友好操作界面的持续集成(CI)工具，起源于Hudson（Hudson是商用的），Jekins是一个可扩展的持续集成引擎。

* 1. 应用背景

1. 持续、自动的构建/测试软件项目，**主要是三个功能**：从github拉源码打包到jenkins服务器，通过ssh将jar包传输到应用服务器，执行shell脚本启动项目。
2. 监控外部的一些定时任务的运行（暂时不做了解）。

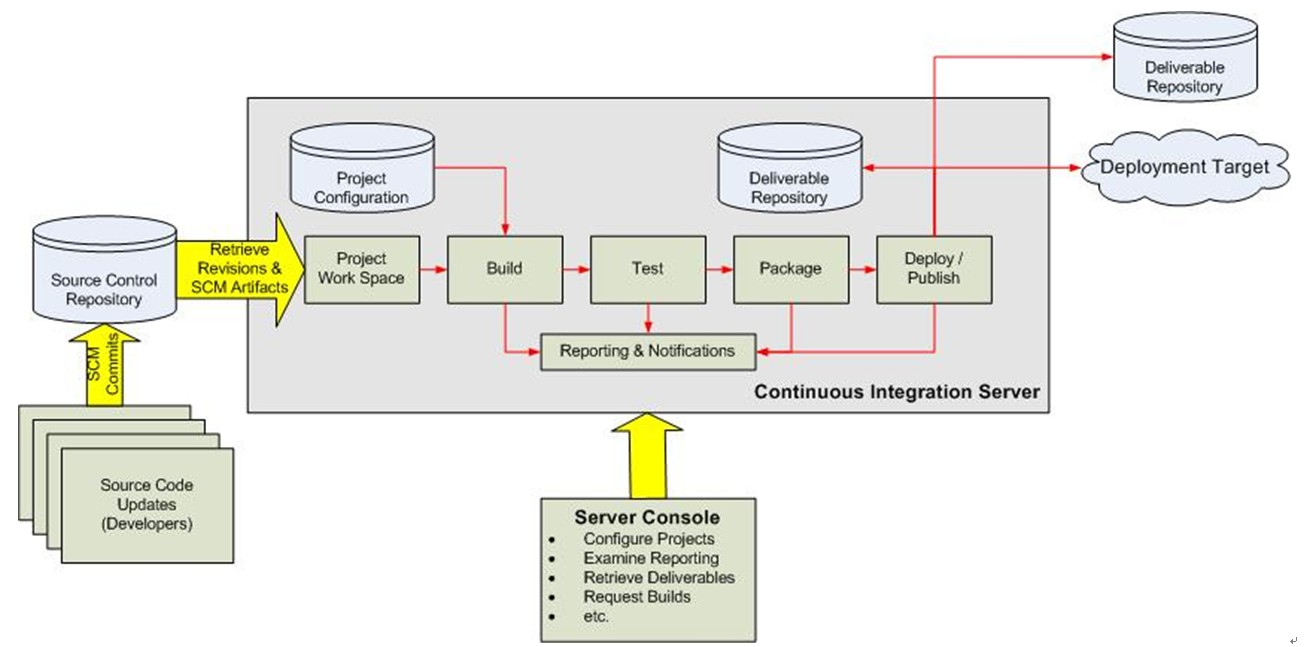
1.3 特点

1. 易于安装-只要把jenkins.war**部署到servlet容器**，不需要数据库支持。可在Tomcat等流行的servlet容器中运行，也可独立运行。
2. 易于配置-所有配置都是通过其提供的web界面实现。
3. 集成RSS/E-mail通过RSS发布构建结果或当构建完成时通过e-mail通知。
4. 生成JUnit/TestNG测试报告。
5. 分布式构建支持Jenkins能够让多台计算机一起构建/测试。
6. 文件识别:Jenkins能够跟踪哪次构建生成哪些jar，哪次构建使用哪个版本的jar等。
7. 插件支持:支持扩展插件，你可以开发适合自己团队使用的工具。
8. 通常与版本管理工具(SCM)、构建工具结合使用；常用的版本控制工具有SVN、GIT，构建工具有Maven、Ant、Gradle。

## 持续集成 CI 系统 – Jenkins就是一个CI系统

1.1 工作流程

(1) 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。



* 1. 系统的各个组成部分是按如下顺序来发挥作用的：

1. 开发者检入代码到源代码仓库。
2. CI系统会为每一个项目创建了一个单独的工作区。当预设或请求一次新的构建时，它将把源代码仓库的源码存放到对应的工作区。
3. CI系统会在对应的工作区内执行构建过程。
4. 配置如果存在，构建完成后，CI系统会在一个新的构件中执行定义的一套测试。完成后触发通知(Email,RSS等等)给相关的当事人。
5. 配置如果存在，如果构建成功，这个构件会被打包并转移到一个部署目标(如应用服务器)或存储为软件仓库中的一个新版本。软件仓库可以是CI系统的一部分，也可以是一个外部的仓库，诸如一个文件服务器或者像Java.net、 SourceForge之类的网站。
6. CI系统通常会根据请求发起相应的操作，诸如即时构建、生成报告，或者检索一些构建好的构件。

## Jenkins的主要目标 -- 监控软件开发流程，快速显示问题。

1.1 主要作用

(1) CI系统在整个开发过程中的主要作用是**控制**：

当系统在代码存储库中探测到修改时，它将运行构建的任务委托给构建过程本身。

如果构建失败了，那么CI系统将通知相关人员，然后继续监视存储库。

它的角色看起来是被动的；但它确能快速反映问题。

* 1. Jenkins的优点

1. Jenkins一切配置都可以在web界面上完成。有些配置如MAVEN\_HOME和Email，只需要配置一次，所有的项目就都能用。当然也可以通过修改XML进行配置。
2. 支持Maven的模块(Module)，Jenkins对Maven做了优化，因此它能自动识别Module，每个Module可以配置成一个job。相当灵活。
3. 测试报告聚合，所有模块的测试报告都被聚合在一起，结果一目了然，使用其他CI，这几乎是件不可能完成的任务。
4. 构件指纹(artifact fingerprint)，每次build的结果构件都被很好的自动管理，无需任何配置就可以方便的浏览下载。

# Jenkins配置

## 工作流程

1. 打包项目。获取源码仓库以及Maven构建Pom文件，就可以根据要求打包项目。
2. 传输项目到服务器上。配置Linux免密登录，就可以传输到对应的服务器上。
3. SSH执行启动脚本。通过SSH执行服务器上的启动脚本，就完成项目的部署。

## 用户/凭据配置

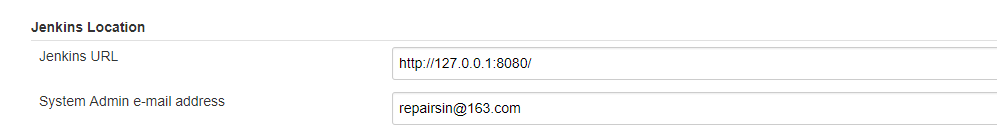
1. 系统管理—管理用户，作用是添加或者删除用户。
2. 凭据配置，作用是添加凭据（SSHkey）。先选择域，再添加凭据。

## 全局工具配置

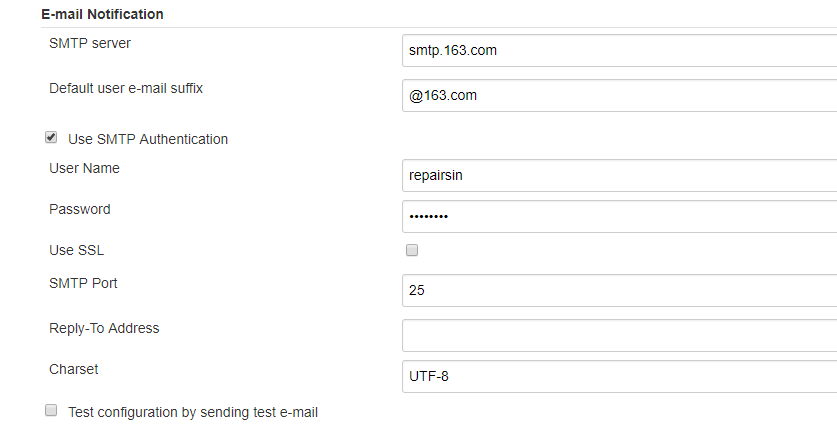
1. 配置JDK MAVEN GIT DOCKER等常用的工具，最好选择本地的配置。

## 邮件配置(系统设置里)

1. 配置一个邮件账号作为管理员通知账号



1. 填写管理员邮件账号的密码，由jenkins代发



## 插件配置-常用插件

1. Maven Integration plugin：在构建新job时有Maven项目选项
2. Build-timeout Plugin：任务构建超时插件
3. Naginator Plugin：任务重试插件
4. Build User Vars Plugin：用户变量获取插件
5. Build Pipeline Plugin View ：Pipeline 管道流图表展示插件
6. Build Flow Plugin：工作流插件，支持DSL脚本定义工作流
7. Build Graph View Plugin：build Flow插件视图（安装后需要重新才能生效）
8. Multijob Plugin：多任务插件
9. Build-timeout Plugin：job构建超时插件
10. Build Timestamp Plugin ：任务log时间戳插件，使得job log的每次输出前面都增加当时的时间
11. Parameterized Trigger Plugin：这是一个扩展型的插件，使各个job连接的时候可以传递一些job相关的信息
12. Join Plugin：这也是一个触发job的插件，亮点在于它触发job的条件是等待所有当前job的下游的job都完成才会发生。
13. Files Found Trigger：检测指定的目录，如果发现指定模式的文件则启动build。
14. BuildResultTrigger Plugin：根据其他的job的成功或失败来启动此build。
15. Publish Over SSH Plugin：通过ssh发布文件
16. Rebuild Plugin：重新执行插件
17. ws-cleanup Plugin ：workspace清理插件
18. Cron Column Plugin： 通过定时任务例行的运行一些job
19. Job Configuration History Plugin：使用心得：使job具备版本管理的能力，diff和rollback功能更是非常赞
20. HTTP Request Plugin：使用心得：在构建前后可以通过该插件以http形式调用各种api接口实现和内部系统的联动
21. Periodic Backup：使用心得：备份是运维一个系统必须要保障的事情，该插件的恢复功能可能不可用，需要手工进行，好处在于可以定时备份
22. Job Import Plugin:使用心得：可以快速导入其他jenkins集群的已有job，需要认证的jenkins系统导入需要提供凭证才可以
23. Status Monitor Plugin：构建状态插件
24. Build Monitor View ：使用心得：基于该插件可以实现dashboard功能
25. Build Environment Plugin：构建环境插件，可以进行构建环境比较。
26. FTP插件：Monitoring：Monitoring of Jenkins
27. 基础依赖插件：jQuery Plugin：jQuery插件

# Win/Linux安装

## Windows安装

1. 下载war包，直接用java -jar Jenkins.war命令启动即可。
2. 可以修改环境变量JENKINS\_HOME变量来修改安装地址。
3. 也可以用tomcat容器发布。

## Linux安装

1. 配置java开发环境
2. 可以修改环境变量JENKINS\_HOME来修改安装地址。
3. 利用命令java -jar Jenkins.war安装即可。