# 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

(1) 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）。

(2) 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、
2. 监控外部的一些定时任务的运行（暂时不做了解）。

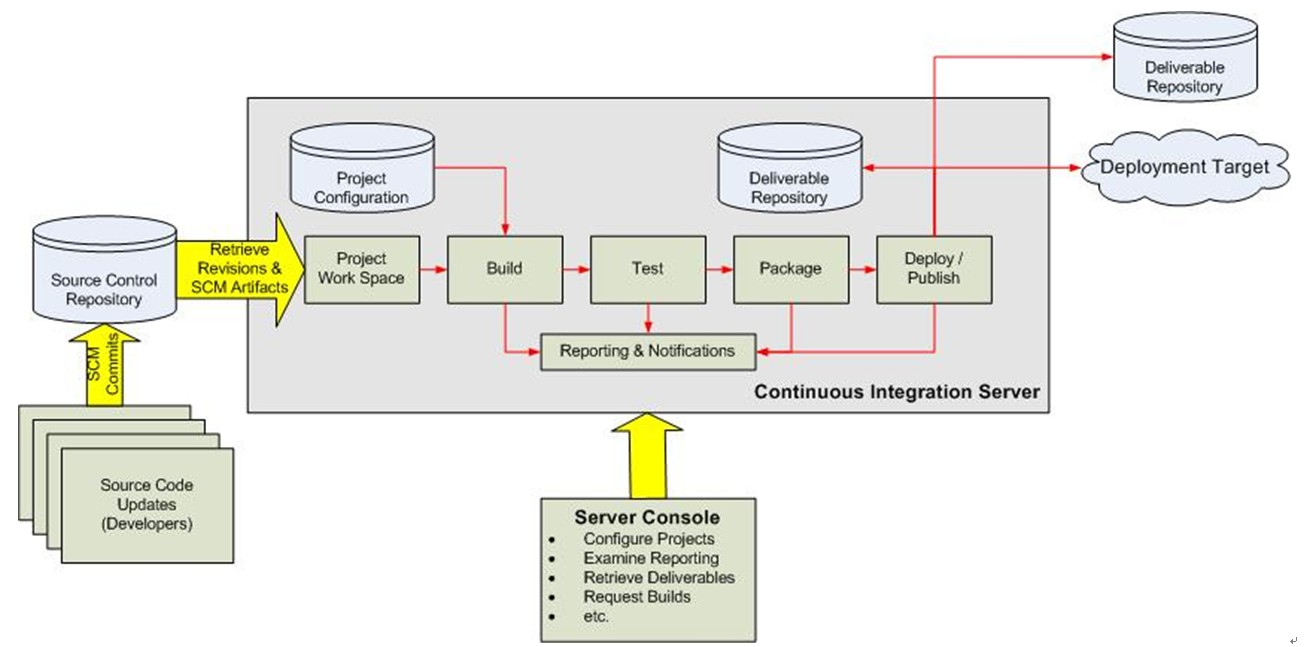
1.3 特点

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。

# 安装配置