

Jenkins概述：

和maven类似, 插件式开发

1. Jenkins概述和环境配置

1. 关于可持续集成(CI)

简化开发人员的重复构建工作.

2. Jenkins概述

CI是一种软件开发实践, 即团队开发成员经常集成他们的工作, 每次集成都通过自动化的构建(包括编译, 发布, 自动化测试)来验证, 从而尽快的发现集成错误, 让团队能够高效的开发软件

持续集成的要点:

1. 统一的代码库(git)
2. 统一的包依赖管理(nexus)
3. 测试自动化
4. 构建全自动化(maven)
5. 部署自动化
6. 可追踪的集成记录

Jenkins就是为了满足上述持续集成的要点而设计的一款工具, 其主题框架采用Java开发, 实质内部功能实现都是由各种插件实现, 极大提高了系统的扩展性, 其不仅满足Java系统的集成, 也可以实现PHP等语言的集成发布, 通过其pipeline插件, 用户可以随自己需要定制集成流程.

3. 安装Jenkins

下载war包通过命令运行:

```
java -jar -Dfile.encoding=UTF-8 jenkins.war --httpPort=8080
```

4. 基础环境配置与常用插件下载

官方网站下载Jenkins

必须用jdk8

配置:

下载完成后, 进入启动页localhost:8080/会有一个验证过程, 验证码在

`${user_home}\.jenkins\secrets\initialAdminPassword`中, 接着就是进入安装插件页, 选择默认即可, 这个过程很长.

基础环境配置与常用软件下载

在集成的时候, Jenkins用到了maven . git 所以服务器中必须提前安装好这些环境, 插件下载:

更换源->系统管理->管理插件->高级->升级站点

把:<http://updates.jenkins-ci.org/update-center.json>

换成:<http://mirror.esuni.jp/jenkins/updates/update-center.json>

镜像源查询:<http://mirrors.jenkins.jenkins-ci.org/status.html>

基本插件列表插件:

插件名称	插件描述
maven integration	maven管理插件
deploy to container	容器部署插件
pipeline	管道集成插件
Email Extension	邮件通知插件
SSH	用于ssh通信

全局工具配置

maven配置:默认是用户下的.m2/setting.xml

jdk使用系统默认的就行

git系统默认

最后是maven的安装目录

简单的部署和构建:

新建任务

build上需要添加参数 clean install -U

构建后操作 选择部署war包maven项目的target下, 然后target/*.war

构建后部署到maven仓库

pipeline 通过脚本构建测试, 打包, 部署整个流程