**[hudson集成](http://www.cnblogs.com/halfacre/archive/2012/07/03/2575063.html)**

Jenkins是一个持续集成工具，前身叫做Hudson，在实际项目应用中非常重要，本文介绍这一工具的使用方法。   
  
首先我们访问Jenkins的网站：   
  
   
  
Jenkins 的网址是：[http://jenkins-ci.org/](http://jenkins-ci.org/" \t "_blank)   
  
从网站下载Jenkins: <http://mirrors.jenkins-ci.org/war/latest/jenkins.war>   
  
可以看到jenkins的发布形式非常简单，就是一个war。运行Jenkins也十分简单：

Bash代码 [复制代码](http://bluedash.iteye.com/blog/1544511) [收藏代码](javascript:void())

1. java -jar jenkins.war

java -jar jenkins.war

启动时日志输出如下：

Bash代码 [复制代码](http://bluedash.iteye.com/blog/1544511) [收藏代码](javascript:void())

1. Running from: /Users/weli/Downloads/jenkins.war
2. webroot: $user.home/.jenkins
3. ...
4. INFO: Jenkins is fully up and running

Running from: /Users/weli/Downloads/jenkins.war

webroot: $user.home/.jenkins

...

INFO: Jenkins is fully up and running

从日志中可以看到jenkins会把配置文件和数据保存在$user.home/.jenkins当中。此时我们访问Jenkins的HTTP服务页面：

Bash代码 [复制代码](http://bluedash.iteye.com/blog/1544511) [收藏代码](javascript:void())

1. http://127.0.0.1:8080

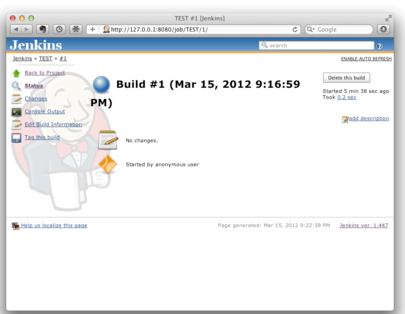
http://127.0.0.1:8080

此时进入界面会发现Jenkins正在初始化，稍等片刻后，就可以看到主界面：   
  
   
  
从上图中可以看到，左侧有New Job可以新建任务。Jenkins的设计思路比较贴近于Linux下面的cron：我们都知道，cron允许我们将一个任务按一定的时间周期去执行。对于cron来讲，任务的触发条件就是时间。而Jenkins将这一思路加以扩展，把触发条件这一块大大加强了。比如Jenkins允许我们以代码库中的代码提交为触发条件，去执行任务。并且支持用户自己去写代码，写插件，去扩展各种需要的触发条件。   
  
我们现在可以创建一个任务试试看：   
  
   
  
可以看到，Jenkins默认支持我们创建多种任务。我们选择第一种，即：

Bash代码 [复制代码](http://bluedash.iteye.com/blog/1544511) [收藏代码](javascript:void())

1. Build a free-style software project

Build a free-style software project

并将Job名命令为TEST：   
  
在指定触发条件时，可以看到Jenkins默认支持多种触发条件，其中 "Build periodically" 就是类似于cron的按时间触发，而Poll SCM则是允许用户让Jenkins定期查询某一个项目的代码库，如果有代码变动则触发执行定义的任务：   
  
   
  
当然，如果要使用Poll SCM的方式来执行任务，则需要在Jenkins中配置一个项目代码库所在的位置。在TEST任务中，我们将需求砍到最简，既不准备指定一个特定的项目，更不存在项目的代码库，我们只准备让Jenkins帮我们执行一行Linux命令ls。此外，在我们的TEST任务中， 我们也不指定任务触发条件，这样的情况下，这个任务就是手工触发的，只有我们让它执行，它才执行。   
  
接下来我们配置要执行的任务：   
  
   
  
我们在TEST任务中使用Execute Shell，即我们的任务是要执行脚本：   
  
   
  
我们的任务非常简单：让Jenkins执行ls命令，没有触发条件，我们手工执行。保存这个任务，试着执行它：   
  
  
   
  
看到左边有个'Build Now'，这就是手工执行这个任务的链接。执行完成后，会出现结果：   
  
   
  
蓝灯代表着任务正确完成。Jenkins判断任务是否正常完成的逻辑很明确：对于执行脚本而言，返回值为0代表任务正常完成。对于Jenkins支持的其它Jobs类型，比如Maven任务，Jenkins会有更为细节的绑定，比如会检查到mvn在执行任务时是否成功，单元测试是否通过，等。   
  
下面是我们任务的执行结果：   
  
   
  
注意到左边有"Console Output"，里面有我们的任务执行的详细日志。这就是Jenkins提供的另一个核心价值：任务执行过程可审计。   
  
相信读完本文，你已经对Jenkins是一个什么样的工具，它的用法都已经有所了解。接下来就是深入使用它，让它在你的项目中充分发挥价值。

转自<http://bluedash.iteye.com/blog/1544511>