[jenkins](http://www.51testing.com/html/28/javascript:;" \t "http://www.51testing.com/html/28/_self)是一个用[Java](http://www.51testing.com/html/28/javascript:;" \t "http://www.51testing.com/html/28/_self)编写的开源自动化服务器,它是Hudson的一个分支project ; 它是一个[持续集成](http://www.51testing.com/html/28/javascript:;" \t "http://www.51testing.com/html/28/_self)软件(continuous integration),它以节点为单位,连接整个工作流, 通过各种类型插件支持构成具有个性化要求的项目持续集成, 通过各种各样的插件(plugin)来实现各个节点的功能, 它们共同完成持续集成(自动部署)/自动[测试](http://www.51testing.com/html/28/javascript:;" \t "http://www.51testing.com/html/28/_self)或者持续交付等工作.

　　它和大多数IDE一样,支持在线/离线方式安装插件. 这里有在jenkins中使用插件指南,以及插件下载.

**点击这里下载jenkins安装包/war包, jenkins安装有以下方式:**

　　1.下载对应环境安装包,例如 Dockerfile [Windows](http://www.51testing.com/html/28/javascript:;" \t "http://www.51testing.com/html/28/_self) linux 各分支版本例如FreeBSD/Gentoo/openSUSE/Red Hat/Fedora/CentOS/Ubuntu/Debian 以及Mac OS X . 好像么有archlinux支持,可能是很少人使用arch做服务器的原因吧,不过可以使用第二种方式解决.

　　2.下载war包支持,使用java -jar或者放tomcat/webapp下启动jenkins服务. 这种方式是最方便的.

　　notice :

　　使用之前记得检查是否已安装jdk 或者java 运行时环境.

**准备jenkins运行环境**

|  |
| --- |
| 系统环境:  　　OS: CentOS6.5 2.6.32-573.22.1.el6.x86\_64  　　JDK:  　　java version "1.7.0\_79"  　　Java(TM) SE Runtime Environment (build 1.7.0\_79-b15)  　　Java HotSpot(TM) 64-Bit [Server](http://www.51testing.com/html/28/javascript:;" \t "http://www.51testing.com/html/28/_self) VM (build 24.79-b02, mixed mode) |

　　●新建jenkins应用主目录,jenkins/jenkins\_home

|  |
| --- |
| ## 新建 jenkins/jenkins\_home 然后进入jenkins目录  　　mkdir -p jenkins/jenkins\_home && cd jenkins |

　　●下载jenkinswar包到当前目录(~/jenkins), 或者从别的目录将下载的war包拷贝到~/jenkins目录下, 例如,cp ~/Downloads/jenkins.war .

　　为了方便, 我使用了java -jar 命令来启动 jenkins , 将启动jenkins需要的环境准备,写入到一个sh文件内,启动jenkins直接运行该sh 文件即可.

|  |
| --- |
| #!/bin/bash  　　##定义jenkins home目录  　　JENKINS\_ROOT=/home/tianzhiwang/jenkins  　　##将jenkins环境变量导入到 /etc/profile 或者 .bash\_profile  　　export JENKINS\_HOME=$JENKINS\_ROOT/jenkins\_home  　　## 修改jvm运行时内存,避免运行jenkins过程中,内存溢出  　　java -Xms1024m -Xmx1024m -XX:PermSize=512M -XX:PermSize=1024M -jar  　　## 指定jenkins应用访问端口, 例如,http://192.168.0.1:8000  　　$JENKINS\_ROOT/jenkins.war --httpPort=8000 |

　　将这个启动脚本保存到自己方便使用的位置.

　　然后将这个脚本文件的权限设置为可执行sh文件.

　　chmod a+x startup.sh

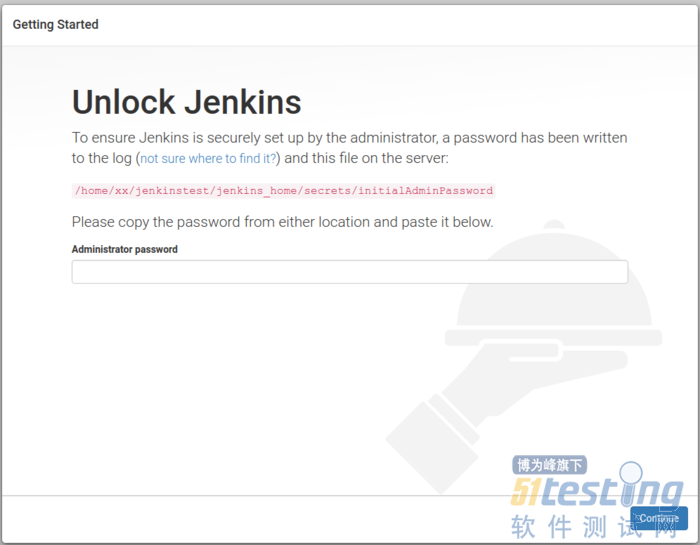
　　然后就可以启动了,进入启动脚本存放目录

　　sh startup.sh &

　　第一次启动,查看启动日志滚动到:

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  　　\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  　　\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  　　Jenkins initial setup is required. An admin user has been created and a password generated.  　　Please use the following password to proceed to installation:  　　4cc8b496097745aaba38d014c2f88c8d  　　This may also be found at: /home/tianzhiwang/jenkinstest/jenkins\_home/secrets/initialAdminPassword  　　\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  　　\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  　　\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

　　这时候就访问jenkins应用完成安全验证,访问地址http://192.168.0.1:8008/ 第一次需要根据提示进入对应目录文件或者直接粘贴命令行窗口输出的的字符串,使用初始化管理员密码登陆,如:

[](http://www.51testing.com/batch.download.php?aid=79466)

　　然后下一步,安装插件,可以自定义安装,也可以按照推荐安装,默认例如 git/sh 等类似插件会被安装,然后勾选安装完成自动重启,点击右上角的允许自动刷新,

　　等待安装完成后,新建用户设置密码,然后使用该用户登陆,新建一个job, 如果是maven项目,还需要安装maven支持插件,jenkins默认核心插件不包含maven项目支持插件.

　　系统管理 > 插件管理 > 可选插件 搜索 maven 找到 Maven Integration plugin 安装,然后就可以选择构建maven项目了.

　　然后,根据描述设置对应构建节点信息,例如,源码管理工具 (git/Subversion或则CVS等).

　　然后设置maven 项目pom文件位置,项目构建的时候jenkins会首先扫描该项目的pom.xml文件,这里可以是绝对路径. 以及maven编译参数,例如 不编译[单元测试](http://www.51testing.com/html/28/javascript:;" \t "http://www.51testing.com/html/28/_self)代码. 如:

　　clean package -Dmaven.test.skip=true

　　然后,到这里可以先保存这个job,然后在jenkins服务首页右边就可以看到刚才新建的job了.

　　进入对应job,然后点击左边的配置 可以再次修改这个job的构建各个节点设置. 例如svn配置. 构建前/后的操作.

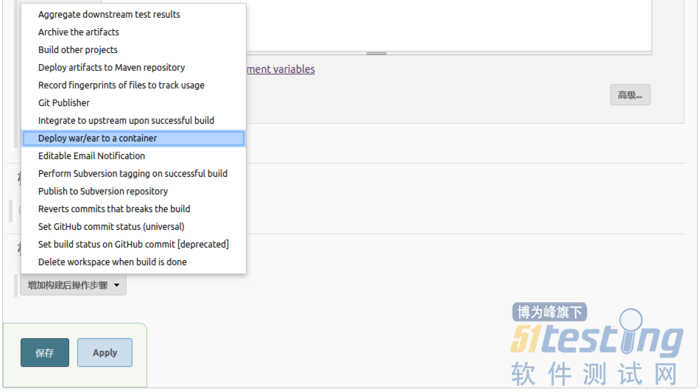
　　在系统管理 > Global Tool Configuration >可以设置全局配置,例如,java环境 JDK环境变量, maven环境变量/setting文件/仓库目录等.

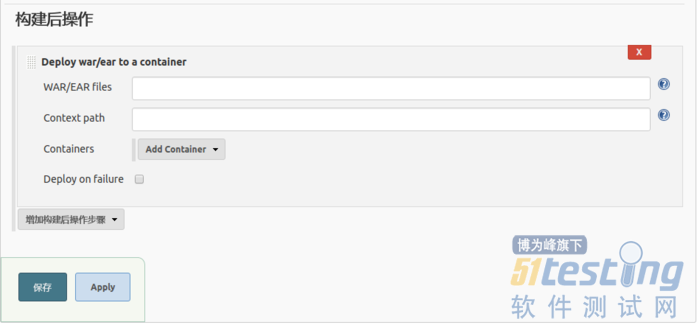
　　默认情况下,项目源码会存放在jenkins\_home/jobs/myproject\_deploy/workspace/下. 而且默认设置为第一次才会checkout操作,以后构建操作都会从指定源码存放服务器更新代码到该目录下. 也可以通过更改设置,使jenkins每次构建都会从版本服务器checkout源码到本地workspace/目录下.

**自动部署项目到容器**

　　通过jenkins插件 - deploy.hpi 可以进行构建完成之后自动发布到指定应用服务器,例如tomcat,weblogic等. 不过需要注意的是,tomcat 和weblogic引用服务器自动部署支持插件是不一样的, 具体可以看插件说明, 如果你跟我一样倒霉,那么不出意外,你下载该插件将会失败, 具体原因,我也不清楚, 那么这时候可以使用第二种方式进行插件安装,去这里下载这个插件离线安装, 更多插件可以在这里找到 .

　　插件安装完毕后,进入job 构建后节点选择deploy war/ear to a container ,如:

[](http://www.51testing.com/batch.download.php?aid=79472)

[](http://www.51testing.com/batch.download.php?aid=79474)

　　然后填写下列信息:

　　war/ear 文件位置,这里需要填写相对路径, 相对workspace root路径, 例如,一般maven项目war包默认都会在target/\*.war .

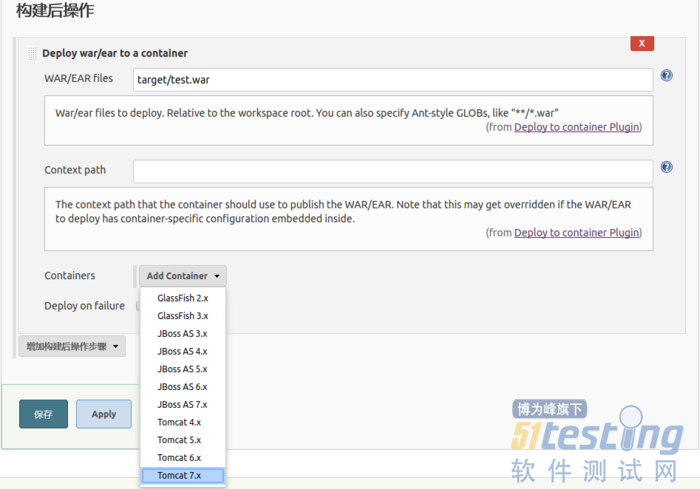
　　点击右边问号帮助,查看关于这个元素的说明:

　　War/ear files to deploy. Relative to the workspace root. You can also specify Ant-style GLOBs, like "\*/.war"

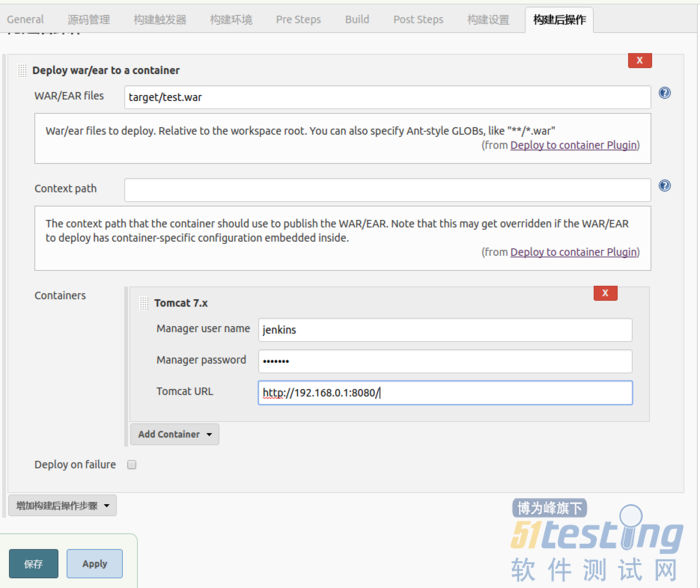
　　Context path 这个是项目path,类似于在tomcat中修改config/server.xml 文件方式来部署项目的path值:

　　<Context   path="/helloworld"   docBase="/home/projects/hello-world" >

　　然后选择具体的容器,并配置其信息比如tomcat:

[](http://www.51testing.com/batch.download.php?aid=79475)

　　列表里的containers都是这个插件所支持的应用服务器. 这个插件支持远程部署, 在构建完成后部署操作jenkins是作为一个客户端来访问tomcat. 所以即使tomcat在本地,这种方式下依然需要通过登陆来访问tomcat的服务,然后部署构建之后的项目文件. 具体信息如下:

[](http://www.51testing.com/batch.download.php?aid=79476)

　　在填写这些信息之前,需要打开构建项目应用使用的tomcat配置,默认使用tomcat没有打开角色配置.

　　进入$CATALINA\_HOME/conf, 然后修改 tomcat-users.xml 文件打开tomcat对用户和角色的配置, 然后新建一个用户(jenkins)和角色(manager-script) ,如:

|  |
| --- |
| .......................  　　<!--  　　  NOTE:  The sample user and role entries below are wrapped in a comment  　　  and thus are ignored when reading this file. Do not forget to remove  　　  <!.. ..> that surrounds them.  　　-->  　　  <role rolename="tomcat"/>  　　  <role rolename="role1"/>  　　  <role rolename="manager-gui"/>  　　  <role rolename="admin-gui"/>  　　  <role rolename="manager-script"/>  　　  <user username="jenkins" password="jenkins" roles="manager-script"/>  　　  <user username="tomcat" password="tomcat" roles="manager-gui"/>  　　  <user username="both" password="tomcat" roles="tomcat,role1"/>  　　  <user username="role1" password="tomcat" roles="role1"/>  　　<!--  　　  <user username="tomcat" password="tomcat" roles="tomcat"/>  　　-->  　　  <user username="role1" password="tomcat" roles="role1"/>  　　</tomcat-users> |

　　然后重新启动tomcat, 然后应用-保存这个job , 然后就可以构建项目了, 左边会出现构建进度条, 下拉选择Consle Output 可以查看 这个job下全部节点下有的日志输出. 例如,经常有人少提交代码导致编译失败等都可以看到. 以及后续节点部署项目重启服务等日志都可以看到,当整个job的全部节点都完成后可以看到一下日志:

|  |
| --- |
| .........  　　Finished: SUCCESS |

　　表示本次构建工作完成并成功.

　　利用jenkins 使用maven 插件编译 -构建可部署文件后,还可以使用自定义脚本来完成deploy.hpi 插件实现的功能. 假设,只需要部署到本地tomcat(和jenkins服务所在机器相同),则需要两个脚本来保证构建部署项目功能稳定.

　　第一个脚本包含功能为:

　　在每次构建之前删除jenkins\_home/jobs/myproject/workspace/target目录, 以及WEB-INF下的classes 目录.

　　第二个脚本功能为:

　　构建完成后,先停止tomcat服务,然后删除tomcat/webapps下的上次部署的项目(包含war包)以及 work/Catalina/localhost 目录下的缓存文件,

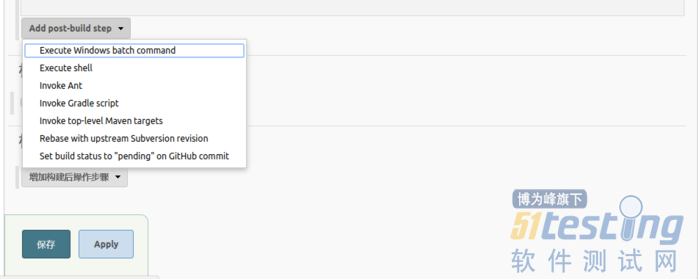
　　将jenkins构建的war从workspace/target目录下复制到tomcat/webapps下

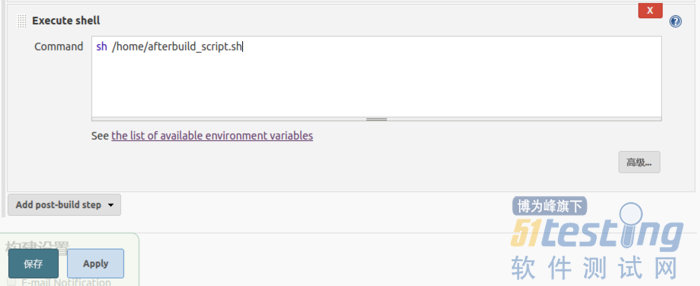
　　然后进入tomcat/bin目录 运行startup.sh文件启动tomcat服务 , 并让这个线程在后台运行.

　　结束.

　　将第二个脚本保存到一个.sh文件中,然后在构建节点 - Post Steps 中执行该脚本.

　　如:

[](http://www.51testing.com/batch.download.php?aid=79478)

[](http://www.51testing.com/batch.download.php?aid=79479)

　　保存配置, 后就可以正常构建项目并自动部署项目到服务器了.

**注意**

　　<code>

　　默认情况下, jenkins跑完一个job中的所有节点后就会关闭这个线程,所以很有可能使用脚本无法启动tomcat服务, 就像我们在命令行启动tomcat:

　　sh startup.sh

　　刚看到一行tomcat启动输出日志, 结果被一个二货按下 Ctrl + c 或者关闭了这个窗口. 结果就是tomcat服务启动一半被中断了,tomcat服务没有启动成功. jenkins 默认就会干这样的事情, 网络上很多博客 问答等都说在脚本执行首行添加如下类似的代码:

|  |
| --- |
| BUILD\_ID=dontKillMe  　　sh  /home/afterbuild\_script.sh |

　　经过实践,这种方法比较扯淡, 中文网站互相copy的习惯确是解决不了问题, 通过 Stack Overflow了解到, 通过脚本tomcat启动不起来确是因为被jenkins中断了线程导致,所以我想如果让这个线程晚一会儿挂掉,等tomcat服务启动后再挂掉应该问题就解决了,所以在第二个脚本中利用sleep 延迟一定事件,如:

|  |
| --- |
| ..................  　　nohup sh startup.sh &  　　## s秒，m分，h时，d天  　　sleep 8s  　　echo 'restarted........' |

　　</code>

　　这样就解决了在jenkins启动tomcat服务不成功的问题.

　　附上我写的部署/重启tomcat服务的bash脚本[本地机器部署],第一次写sh脚本,一边学语法一边写,幼儿园水平,走过路过,别一笑而过! \_#

|  |
| --- |
| #!/bin/bash  　　export JAVA\_HOME=/home/hostname/jdk/jdk1.7.0\_79  　　tomcat\_pid=$(ps -ef|grep 'apache-tomcat-7.0.63'|grep -v grep|grep -v kill|awk '{print $2}')  　　[ -n "$tomcat\_pid" ] && kill -9 $tomcat\_pid  　　echo 'killed pid is '$tomcat\_pid  　　echo '===================================================='  　　sleep 2s  　　## tomcat => webapp/\*.war  　　tomcat\_home=/home/hostname/tomcat/apache-tomcat-7.0.63  　　webapp\_path=$tomcat\_home/webapps/  　　cache\_path=$tomcat\_home/work/Catalina/localhost/  　　jenkins\_job\_war=/home/hostname/jenkins/jenkins\_home/jobs/myproject\_deploy/workspace/target  　　echo 'clear tomcat cache in '$webapp\_path'========================================='  　　cd $webapp\_path && rm -rf myproject\*  　　echo 'clear tomcat cache in '$cache\_path'============================================'  　　cd $cache\_path && rm -rf myproect  　　echo 'deploy war > copy '$jenkins\_job\_war/myproject.war'==>'$webapp\_path  　　cd $jenkins\_job\_war && cp myproject.war $webapp\_path  　　echo 'copy done, instantly restart........'  　　echo '================================================  　　cd $tomcat\_home/bin  　　source /etc/profile  　　### ./startup.sh &  　　nohup sh startup.sh &  　　sleep 8s  　　echo 'restarted........' |