## jenkins+tomcat+ant+java+linux 自动部署项目

2017-10-31 [程序员技术前沿](https://mp.weixin.qq.com/s?__biz=MzIxMTI5NTQxNQ==&tempkey=OTI4X0ZaQ2x1bXRzZERUdG8zR2stRDJpS1V5TmpncU1TckM1QW9YTzJZb1hxZ1FkQXdqb2ZFeDVtUnBHQUFodTg4bU9ZYjZiVEY3ZWJ6VENUcUdhd1JXUGtKbXlGaVFzckMyaklGNHFLVkdRM0wzaVZHa3owUHZLUkZvTTRvSmhmOWQxbmlOcGxuQlRUNS1NcllVclhwQWl0TWxJanJQYThqN3VlWHB1WkF+fg==&chksm=0f5c997d382b106b4d3f8dac7ebee3c70ee385bb16963b3c1aeaa40f9ed1e98309b5efed9d3e&client=tim&ptlang=2052&ADUIN=917597939&ADSESSION=1509497246&ADTAG=CLIENT.QQ.5531_.0&ADPUBNO=26745" \l "#)

### 最终效果：

每1小时自动查下svn，如果svn有提交，则自动更新部署。  
  
具体环节包括：更新代码、编译代码、打包代码、拷贝代码、备份旧war、删除缓存、停启tomcat。

### 使用环境：

java+tomcat+ant+jenkins+linux

### ****1、准备jdk****

由于需要编译java，所以必须是jdk，oracle 官网的jdk下载地址如下，文件选择后缀名为tar.gz的jdk下载

http://www.oracle.com/technetwork/Java/javase/archive-139210.html

然后解压

tar  -zxvf   压缩文件名.tar.gz

编辑环境变量

vim /etc/profile

最后面添加以下配置

export JAVA\_HOME=/usr/local/jdk1.7.0\_80/  
export PATH=$PATH:$JAVA\_HOME/bin  
export CLASSPATH=.:$JAVA\_HOME/lib/tools.jar:$JAVA\_HOME/lib/rt.jar

激活环境变量

source /etc/profile

### ****2、准备ant****

wget https://www.apache.org/dist/ant/binaries/apache-ant-1.9.7-bin.zip

unzip apache-ant-1.9.7-bin.zip

### ****3、准备jenkins****

wget http://mirrors.jenkins-ci.org/war/latest/jenkins.war

官网下载很慢，特放到网盘http://pan.baidu.com/s/1eSfS4SU，当前版本号是2.8

将jenkins放入tomcat并启动，然后访问并next next初始化

jenkins所在tomcat,和应用所在tomcat必须是两个独立的tomcat

(因为要通过jenkins启停web应用的tomcat，此时jenkins所在tomcat不能停)

### ****4、准备项目build.xml，我放在了项目根目录****

**[html]** view plain copy

**<?xml** version="1.0" encoding="UTF-8"**?>**

**<project** name="ecp" default="deploy" basedir="."**>**

    <!--

        Must set these args in jenkins!!!

        TOMCAT\_HOME=/usr/local/apache/apache-tomcat-8.0.35

     --**>**

**<property** name="webapp.name" value="ROOT" **/>**

**<property** name="dist.dir" value="${basedir}/dist" **/>**

**<property** name="ant.dir" value="/root/.jenkins/plugins/ant/WEB-INF/lib" **/>**

**<property** name="webRoot.dir" value="${basedir}/ROOT" **/>**

    <!-- <property name="src.dir" value="${basedir}/src" /> -->

**<property** name="config.dir" value="${basedir}/config" **/>**

**<property** name="lib.dir" value="${webRoot.dir}/WEB-INF/lib" **/>**

**<property** name="build.dir" value="${basedir}/build" **/>**

    <!-- 使用eclipse jdt进行编译，而不使用JDK编译

**<property** name="build.compiler" value="org.eclipse.jdt.core.JDTCompilerAdapter" **/>**

     --**>**

    <!-- 初始化classpath -->

**<path** id="project.classpath"**>**

**<fileset** dir="${lib.dir}"**>**

**<include** name="\*\*/\*.jar" **/>**

**</fileset>**

        <!-- 添加tomcat类路径 -->

**<fileset** dir="${TOMCAT\_HOME}/lib"**>**

**<include** name="\*.jar" **/>**

**</fileset>**

        <!-- ant lib包  -->

**<fileset** dir="${ant.dir}"**>**

**<include** name="\*\*/\*.jar" **/>**

**</fileset>**

**</path>**

    <!-- get the source compile classpath in a printable form -->

**<pathconvert** pathsep="${line.separator}|   |-- " property="echo.path.compile" refid="project.classpath"**>**

**</pathconvert>**

    <!-- show classpath jars -->

**<target** name="print\_classpath"**>**

**<echo** message="|-- compile classpath and basedir"**/>**

**<echo** message="|   |"**/>**

**<echo** message="|   |-- ${echo.path.compile}"**/>**

**<echo** message="|   |-- ${basedir}"**/>**

**</target>**

**<tstamp** prefix="my"**>**

**<format** property="day" pattern="yyyy-MM-dd"**/>**

**<format** property="time" pattern="HH:mm:ss"**/>**

**<format** property="dt" pattern="yyyy-MM-dd\_HH-mm-ss"**/>**

**</tstamp>**

**<target** name="checks"**>**

**<available** file="${TOMCAT\_HOME}/webapps/${webapp.name}.war"  property="found.war"**/>**

**</target>**

    <!-- 备份上次war包 -->

**<target** name="backup" depends="checks" description="备份旧文件" if="found.war"**>**

**<echo** message="begin backup..." **/>**

**<move** file="${TOMCAT\_HOME}/webapps/${webapp.name}.war"

            tofile="${TOMCAT\_HOME}/webapps/${webapp.name}.war.${my.dt}"**></move>**

**</target>**

    <!-- 删除之前的目录结构 -->

**<target** name="clear" depends="backup" description="清理旧文件"**>**

**<echo** message="begin clear..." **/>**

**<delete** dir="${build.dir}" **/>**

**<delete** dir="${dist.dir}" **/>**

        <!-- <delete file="${TOMCAT\_HOME}/webapps/${webapp.name}.war" /> -->

**<delete** dir="${TOMCAT\_HOME}/webapps/${webapp.name}" **/>**

**<delete** dir="${TOMCAT\_HOME}/work/Catalina/localhost/" **/>**

**</target>**

    <!-- 创建目录结构 -->

**<target** name="init" depends="clear" description="创建初始化目录结构"**>**

**<echo** message="begin init..." **/>**

**<mkdir** dir="${build.dir}/classes" **/>**

**<mkdir** dir="${dist.dir}" **/>**

**</target>**

    <!-- 编译java -->

**<target** name="compile" depends="init" description="编译java文件"**>**

**<echo** message="begin compile..." **/>**

**<javac** destdir="${build.dir}/classes" includeantruntime="false" nowarn="on"

            source="1.7" target="1.7" deprecation="true" debug="true"

            encoding="UTF-8" classpathref="project.classpath" **>**

**<src** path="config" **/>**

**<src** path="core" **/>**

**<src** path="crm" **/>**

**<src** path="em" **/>**

**<src** path="fa" **/>**

**<src** path="hr" **/>**

**<src** path="oa" **/>**

**<src** path="scm" **/>**

**<src** path="src" **/>**

**<src** path="sso" **/>**

**<src** path="test" **/>**

**<src** path="thirdParty" **/>**

**<src** path="wf" **/>**

**<compilerarg** line="-Xlint:unchecked" **/>**

            <!-- <classpath refid="project.classpath" /> -->

**</javac>**

**<copy** todir="${build.dir}/classes"**>**

**<fileset** dir="${config.dir}"**>**

**<include** name="\*\*/\*.xml" **/>**

**<include** name="\*\*/\*.properties" **/>**

**<include** name="\*\*/i18n/\*.properties" **/>**

**</fileset>**

**</copy>**

**<echo** message="end compile..." **/>**

**</target>**

    <!-- 将class文件打成 jar包 -->

    <!--

**<target** name="pack" depends="compile"**>**

**<jar** jarfile="${build.dir}/${webapp.name}.jar"**>**

**<fileset** dir="${build.dir}/classes"**>**

**<include** name="\*\*/\*.class"**/>**

**</fileset>**

**</jar>**

**</target>**

    --**>**

    <!-- 打成war包, 名称默认为 项目名 -->

**<target** name="war" depends="compile" description="将工程打成war包"**>**

**<echo** message="begin war...${webRoot.dir}..." **/>**

**<war** destfile="${dist.dir}/${webapp.name}.war" basedir="${webRoot.dir}" webxml="${webRoot.dir}/WEB-INF/web.xml"**>**

**<lib** dir="${lib.dir}" **/>**

**<classes** dir="${build.dir}/classes" **/>**

**<fileset** dir="${webRoot.dir}"**>**

**<include** name="\*\*\*.\*" **/>**

**</fileset>**

**</war>**

**<echo** message="end war..." **/>**

**</target>**

    <!-- copy war包 tomcat的deploy目录 -->

**<target** name="deploy" depends="war" description="部署项目"**>**

**<echo** message="begin deploy..." **/>**

**<copy** file="${dist.dir}/${webapp.name}.war" todir="${TOMCAT\_HOME}/webapps" **/>**

**<echo** message="end deploy..." **/>**

**</target>**

**</project>**

### ****5、jenkins所在虚拟机和应用所在虚拟机，不是同一虚拟机时（可选）****

如上脚本，限于jenkins部署到本机的tomcat，（注意应用**不要**和jenkins共用tomcat）

如果需要部署到另一台虚拟机的tomcat,可以使用以下部署脚本

**[html]** view plain copy

<!--把WAR包远程上传到tomcat -->

**<target** name="deploy" depends="war" description="部署项目"**>**

    <!-- 删除服务器上的war包 -->

**<sshexec** host="${tomcat.server}" username="${tomcat.user}" password="${tomcat.password}"

        trust="true" command="cd ${remote.dir}; rm -rf ${packagename}.war;cd ${tomcat.apps.dir}; rm -rf ${tomcat.apps.appname}" **/>**

**<scp** file="${result}/${packagename}-${version}/${packagename}.war" trust="true" username="${tomcat.user}"

      password="${tomcat.password}" todir="${tomcat.user}@${tomcat.server}:${remote.dir}" **/>**

**</target>**

### ****配置jekins的全局设置,关联前面准备的jdk**和ant**



### ****7、jenkins 新建自由风格的job****

(1)配置“源码管理”，把项目的svn地址帐号密码配置进去

(2)配置“构建触发器”->“Poll SCM”，输入\*/60 \* \* \* \*，代表每60分钟构建一次，如果代码未更新则不构建，此处为cron表达式，不懂的可以百度，不会的可以用在线工具生成，http://cron.qqe2.com/

(3)配置“构建”，新建“Execute Shell”,录入如下内容，关闭tomcat服务：

**[python]** view plain copy

export TOMCAT\_HOME=/usr/local/tomcat/apache-tomcat-8.0.35

echo "[Deploy] Shutting down tomcat"

**if** [ `ps auxwwww|grep $TOMCAT\_HOME|grep -v grep|wc -l` -gt 0 ]

then

**for** pid **in** `ps auxwww|grep $TOMCAT\_HOME|grep -v grep|tr -s ' '|cut -d ' ' -f2`

do

kill -9 $pid 2>&1 > /dev/null

done

fi

(4)配置“构建”，新建“Invote Ant”，并且点“高级”：

AntVersion 选择刚才配置的ant

BuildFile 输入：build.xml(如果此文件不在项目根目录，则应该写上其路径)

Properties输入：TOMCAT\_HOME=/usr/local/tomcat/apache-tomcat-8.0.35，这个参数是在build.xml中调用的  
(5)配置“构建”，新建“Execute Shell”,录入如下内容，启动tomcat服务：

**[python]** view plain copy

echo "[Deploy] Starting up tomcat"

export BUILD\_ID=XXXXXX

/usr/local/tomcat/apache-tomcat-8.0.35/bin/startup.sh

### ****8、启动吧！注意查看Console Output。****