

Maven插件wagon-maven-plugin自动化部署Java项目到Linux远程服务器

🕒 发表于 2015-10-23

通常情况下，程序开发完成后部署到线上Linux服务器，可能需要经历打包、将包文件传到服务器、SSH连上服务器、敲命令启动程序等一系列繁琐的步骤。实际上这些步骤都可以通过Maven的一个插件wagon-maven-plugin来自动完成。

文件上传到服务器

Maven项目可使用 `mvn package` 指令打包，打包完成后包位于target目录下，要想在远程服务器上部署，首先要将包上传到服务器。

在项目的pom.xml中配置wagon-maven-plugin插件：

```
1  <build>
2      <extensions>
3          <extension>
4              <groupId>org.apache.maven.wagon</groupId>
5              <artifactId>wagon-ssh</artifactId>
6              <version>2.8</version>
7          </extension>
8      </extensions>
9      <plugins>
10         <plugin>
11             <groupId>org.codehaus.mojo</groupId>
12             <artifactId>wagon-maven-plugin</artifactId>
13             <version>1.0</version>
14             <configuration>
15                 <fromFile>target/test.jar</fromFile>
16                 <url>scp://user:password@192.168.20.128/home/xxg/Desktop</url>
17             </configuration>
18         </plugin>
19     </plugins>
20 </build>
```

文章目录

1. 文件上传到服务器
2. 在服务器上执行Linux命令
 - 2.1. 运行jar文件
 - 2.2. 上传war包并启动Tomcat
3. 配置execution

```
18         </plugin>
19     </plugins>
20 </build>
```

`<fromFile>target/test.jar</fromFile>` 是要上传到服务器的文件，一般来说是jar或者war包，`<url>scp://user:password@192.168.20.128/home/xxg/Desktop</url>` 配置服务器的用户、密码、地址以及文件上传的目录。

配置完成后，运行命令：

```
1 mvn clean package wagon:upload-single
```

`package`、`wagon:upload-single` 分别对项目进行打包和上传操作。命令运行结束后，文件就会成功上传到Linux服务器。

在服务器上执行Linux命令

部署项目不仅要把包传上服务器，而且还需要执行一些指令来启动程序。在程序启动之前，可能还需要将原来的程序关闭。

运行jar文件

启动jar包通常会使用 `java -jar test.jar` 命令，可以将命令配置在pom.xml中：

```
1 <build>
2     <extensions>
3         <extension>
4             <groupId>org.apache.maven.wagon</groupId>
5             <artifactId>wagon-ssh</artifactId>
6             <version>2.8</version>
7         </extension>
8     </extensions>
9     <plugins>
10         <plugin>
11             <groupId>org.codehaus.mojo</groupId>
12             <artifactId>wagon-maven-plugin</artifactId>
13             <version>1.0</version>
14             <configuration>
15                 <fromFile>target/test.jar</fromFile>
16                 <url>scp://user:password@192.168.20.128/home/xxg/Desktop</url>
17                 <commands>
18                     <!-- 杀死原来的进程 -->
19                     <command>kill -f test.jar</command>
```

```

20         <!-- 重新启动test.jar，程序的输出结果写到nohup.out文件中 -->
21         <command>nohup java -jar /home/xxg/Desktop/test.jar > /home/xxg/Desktop/nohup.out 2>&1 &&</com
22     </commands>
23     <!-- 显示运行命令的输出结果 -->
24     <displayCommandOutputs>true</displayCommandOutputs>
25 </configuration>
26 </plugin>
27 </plugins>
28 </build>

```

配置完成后，运行命令：

```
1 mvn clean package wagon:upload-single wagon:sshexec
```

`package`、`wagon:upload-single`、`wagon:sshexec` 分别对项目进行打包、上传、运行command命令的操作。命令运行结束后，在服务器上查看进程 `ps -ef|grep test.jar`，或者查看nohup.out文件，就可以看到Java程序在服务器上已经启动。

上传war包并启动Tomcat

如果是Web应用，可使用服务器上的Tomcat来部署。

```

1 <build>
2     <extensions>
3         <extension>
4             <groupId>org.apache.maven.wagon</groupId>
5             <artifactId>wagon-ssh</artifactId>
6             <version>2.8</version>
7         </extension>
8     </extensions>
9     <plugins>
10        <plugin>
11            <groupId>org.codehaus.mojo</groupId>
12            <artifactId>wagon-maven-plugin</artifactId>
13            <version>1.0</version>
14            <configuration>
15                <fromFile>target/javawebdeploy.war</fromFile>
16                <!-- 上传到Tomcat的webapps目录下 -->
17                <url>scp://user:password@192.168.20.128/coder/tomcat/apache-tomcat-7.0.55/webapps</url>
18            <commands>
19                <!-- 重启Tomcat -->
20                <command>sh /coder/tomcat/apache-tomcat-7.0.55/bin/shutdown.sh</command>
21                <command>rm -rf /coder/tomcat/apache-tomcat-7.0.55/webapps/javawebdeploy</command>

```

```

22         <command>sh /coder/tomcat/apache-tomcat-7.0.55/bin/startup.sh</command>
23     </commands>
24     <displayCommandOutputs>true</displayCommandOutputs>
25 </configuration>
26 </plugin>
27 </plugins>
28 </build>

```

完成以上配置后，同样可通过 `mvn clean package wagon:upload-single wagon:sshexec` 命令自动部署。

配置execution

如果你觉得 `mvn clean package wagon:upload-single wagon:sshexec` 命令太长了不好记，那么可以配置execution，在运行package打包的同时运行upload-single和sshexec。

```

1  <build>
2      <extensions>
3          <extension>
4              <groupId>org.apache.maven.wagon</groupId>
5              <artifactId>wagon-ssh</artifactId>
6              <version>2.8</version>
7          </extension>
8      </extensions>
9      <plugins>
10         <plugin>
11             <groupId>org.codehaus.mojo</groupId>
12             <artifactId>wagon-maven-plugin</artifactId>
13             <version>1.0</version>
14             <executions>
15                 <execution>
16                     <id>upload-deploy</id>
17                     <!-- 运行package打包的同时运行upload-single和sshexec -->
18                     <phase>package</phase>
19                     <goals>
20                         <goal>upload-single</goal>
21                         <goal>sshexec</goal>
22                     </goals>
23                     <configuration>
24                         <fromFile>target/javawebdeploy.war</fromFile>
25                         <url>scp://user:password@192.168.20.128/coder/tomcat/apache-tomcat-7.0.55/webapps</url>

```

```
26         <commands>
27             <command>sh /coder/tomcat/apache-tomcat-7.0.55/bin/shutdown.sh</command>
28             <command>rm -rf /coder/tomcat/apache-tomcat-7.0.55/webapps/javawebdeploy</command>
29             <command>sh /coder/tomcat/apache-tomcat-7.0.55/bin/startup.sh</command>
30         </commands>
31         <displayCommandOutputs>true</displayCommandOutputs>
32     </configuration>
33 </execution>
34 </executions>
35 </plugin>
36 </plugins>
37 </build>
```

配置完成后，即可使用 `mvn clean package` 来代替 `mvn clean package wagon:upload-single wagon:sshexec`。

📁 Maven 🏷️ Java Maven



上一篇：

◀ [Java WebSocket + tail命令实现Web实时日志](#)

下一篇：

▶ [Java开发Web Service客户端技巧：wsimport、jaxws-maven-plugin、整合Spring](#)

登录

来说两句吧...

畅言一下

评论

33 人参与, 11 条评论



★must↑

2017年5月25日 10:58

请问如何上传到多个服务器上？

回复



寂寞咖啡

2017年4月20日 18:33

问一下tomcat关闭，重启命令不好使怎么回事呀？谢谢。

回复



焦点房产网

焦点看房APP是搜狐旗下全方位的房产服务平台，业务涵盖新房、二手房、装修、租房。独具特色的视频直播和在线互动，让您足不出户即可充分了解房源，减少无谓的奔波之苦。



广告

[查看详情>](#)



Lusa

2016年12月30日 15:41

连接失败

回复



Lusa

2016年12月30日 15:41

连接失败

回复





绊

2016年10月21日 13:26

是，学习了！

回复



吴某人

2016年10月21日 12:32

Jenkins的确更好用，但是前提是要安装部署好，这个更加一键完成，简单粗暴

回复



绊

2016年10月21日 11:27

现在流行使用Jenkins来部署：)

回复



吴某人

2016年3月4日 20:59

还可以吧用户密码配置在Maven的settings.xml中，可以参考这一篇未删减版http://blog.csdn.net/xiao__gui/article/details/49420813

回复



LiuC

2016年3月4日 17:09

我的用户密码里面包含了@符号 怎么破？

回复



吴某人

2016年2月21日 16:48

你可以试试在url上加上端口号看行不行，我也没试过不太确定。

```
<url>scp://192.168.20.128:port/</url>
```

回复



程序辉

2016年2月18日 11:07

ssh 端口在哪配置 我的不是默认的22



分类

Android ²
Java ¹²
Jetty ²
Maven ⁶
MyBatis ¹
Nginx ³
Node.js ¹
Spring Framework ⁵
Struts 2 ²
XMPP ²
异步网络编程 ¹²

标签

Java ⁴⁰	Python ¹²	MINA ¹²	Netty ¹²	Twisted ¹²	TCP ¹²	Maven ⁸	Spring Framework ⁷	线程 ⁵	protobuf ³	JavaScript ³
Jetty ³	Nginx ³	Tigase ²	Android ²	Spring Security ²	Tomcat ²	MySQL ²	XMPP ²	Struts 2 ²		

友情链接

朱晔的博客
张兄才的博客
杨灿的博客
石尧的博客
倪升武的博客
王根深的博客
邹晶的博客

 RSS 订阅