### 微服务发布

* 上传JDK

micro-parent/doc/tools/pom.xml

<properties>

<serverUrl>scp://deng:deng@192.168.79.139</serverUrl>

*<serverPath>/home/deng/prod</serverPath>*

<jdkFile>E*:/CentOsShare/jdk-8u72-linux-x64.tar.gz</jdkFile>*

*</properties>*

三个值分别对应服务器url、服务器路径和本地jdk1.8压缩包

1.选pom.xml，右键->Run As->Maven build...,在Goals中输入wagon:upload-single wagon:sshexec(注：上传并解压到当前目录下)

* 部署代码

在对应工程的pom.xml中

<properties>

<profile.id>test</profile.id> <serverUrl>scp://deng:deng@192.168.79.139</serverUrl>

<serverPath>/home/deng/test/</serverPath>

</properties>

三个值分别对应环境、服务器url和服务器路径

1. 发布全量代码包括依赖的jar
2. 选pom.xml，右键->Run As->Maven build...,在Goals中输入package wagon:upload wagon:sshexec -Doriginal= -P test(test为对应环境,上传所有代码的jar文件和启动脚本bin/startup.sh，并创建target、backup、logs和bin四个目录,备份之前jar文件)
3. 发布全量代码(包括common代码),不包括依赖的jar
4. 选pom.xml，右键->Run As->Maven build...,在Goals中输入package wagon:upload wagon:sshexec -P test(test为对应环境,上传全量代码的jar，不包括依赖的jar和启动脚本bin/startup.sh, 解压替换代码，备份之前jar文件)
5. 如common代码被修改，需要在common先运行install