

java服务打jar包发布运行以及maven配置

- (1) 在idea打包;
- (2) cmd运行成功!

```
java -jar 项目名称
```

- (3) 发送到linux目录 (用命令发送, 也可以使用工具发送!)

```
scp root@192.168.43.154 linux_jar-0.0.1-SNAPSHOT.jar /home
```

scp 发送命令

root@192.168.43.154 linux用户名和端口

linux_jar-0.0.1-SNAPSHOT.jar 服务名称

/home 发送到linux那个目录

- (4) 在Linux查看文件, 并编译jar命令。

```
nohup java -jar linux_jar-0.0.1-SNAPSHOT.jar &
```

nohup 命令

java -jar 编译jar命令

linux_jar-0.0.1-SNAPSHOT.jar 服务名称

& 命令

- (5) 运行服务

```
tail -f nohup.out 运行服务!
```

- (6) git创建分支

```
git checkout -b smartdragon 创建命令
```

```
git push origin smartdragon 提交到远程的分支
```

- (7) maven打jar包的核心配置

```
<!--打jar包配置-->
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
    <plugin><!--编译跳过测试文件检查的生命周期-->
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <configuration>
        <skip>true</skip>
      </configuration>
    </plugin>
  </plugins>
```

```
</plugins>
<resources>
  <!--如果pro和xml文件放在源码java包下，也需要编译-->
  <resource>
    <directory>src/main/java</directory>
    <includes>
      <include>**/*.properties</include>
      <include>**/*.xml</include>
    </includes>
    <filtering>false</filtering>
  </resource>
</resources>
</build>
```

(8) linux停用端口服务

```
netstat -nltp //查看所有端口号的情况!
kill //端口号的PID
```