Title: Spring Boot with spring-boot-maven plug-in - unable to RESTART the application server after running integration tests

Post Body:

I have a Spring Boot application and that application has integration tests.

Here is the relevant snippet from my pom.xml for running the integration tests.

```
<plugin>     <groupId>org.springframework.boot</groupId>     <artifactId>spring-boot-maven-plugin</artifactId>     <configurat
```

When I run `mvn verify`, I see the unit tests running first. Then the application server starts and the integration tests run successfully. The mvn process exits with a success.

I also see the following in the console log which leads me to believe that the application server is being shutdown correctly -

```
[INFO] --- spring-boot-maven-plugin:2.1.1.RELEASE:stop (post-integration-test) @ application --- [DEBUG] Configuring mojo org.
```

Subsequently, when I run `mvn verify` again, I get the following bind exception.

```
[DEBUG] Application argument(s):  [DEBUG] Connecting to local MBeanServer at port 7654 [DEBUG] Waiting for spring application
```

Using `netstat` and `ps` I am able to ascertain that there is some process that is still holding on to port 7654, and `jps` tells me that it is this process -

```
47148 Application -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=7654 -Dcom.sun.management.jmxremote.authe
```

I don't know how to get the JMX process to shutdown. I assumed that the `spring-boot-maven` plug-in would do that. And that does not seem to be the case.

What am I doing wrong and can someone please help? Really appreciate you taking time to look at this (potential) problem.

Accepted Answer: None

Highest Rated Answer:

I was facing the same issue, not being able to rerun tests before killing the JMX process. Next to that, my Azure DevOps pipeline was hanging because of the JMX process not being stopped.

```
The STDIO streams did not close within 10 seconds of the exit event from process '/usr/share/apache-maven-3.6.3/bin/mvn'. This
```

When I browsed through the source code: https://github.com/spring-projects/spring-boot/blob/master/spring-boot-project/spring-boot-tools/spring-boot-maven-plugin/src/main/java/org/springframework/boot/maven/S I discovered that the JMX agent would not be booted if I did not fork the maven process.

I'm not sure this is applicable for you, but I did not require forking per se, and also did not need JMX metrics, so this fixed it for me:

```
<configuration>   <fork>false</fork> </configuration>
```

I also tried to override the jmx arguments (-Dcom.sun.management.jmxremote), by specifying them in the plugin configuration, but since the arguments are added (hardcoded) as the final arguments, that was not working for me.