

Title: spring-boot cannot start in docker

Post Body:

I've a little problem with run my spring boot application in docker.

stack: maven 3+, spring boot(jpa/rest/jetty) - mysql - deploy in docker

So, i've have in my pom file

```
<parent>          <groupId>org.springframework.boot</groupId>          <artifactId>spring-boot-starter-parent</artifactId>          <version>
```

Environment: Ubuntu 16.04 x64 The problem: Locally: I try to run my app with follow command in terminal

```
user$ java -Xmx768m -jar /mnf-backend.jar --spring.datasource.url=jdbc:mysql://$MYSQL_PORT_3306_TCP_ADDR/app_1?autoReconnect=true
```

it's not good by i can tolerate it. But not docker. When commands above will be run in docker then docker stop container (because -> app exit with status 1)

```
ENTRYPOINT ['java', '-Xmx768m', '-jar', '/mnf-backend.jar', '--spring.datasource.url=jdbc:mysql://$MYSQL_PORT_3306_TCP_ADDR/app_1?autoReconnect=true']
```

Docker will start container 1 second and immediately stop container because java return control. I look for method which allow me to configure spring app for predictable behavior or any ideas how to improve my docker instructions. my dockerfile content:

```
FROM frovlad/alpine-oraclejdk8:slim ENV MNFB_ENV production ENV SERVER_PORT 9000 ADD ./builds/mnf-latest.jar mnf-backend.jar
```

docker logs of a container

For example: when i've start nodejs app control not return until application not finished

```
user$ node ./server.js [...here program output and stdout strings] [... it may be stopped by ctrl+c for example]
```

Accepted Answer:

I think the problem is the ampersand (&) in your command line:

```
--spring.datasource.url=jdbc:mysql://$MYSQL_PORT_3306_TCP_ADDR/app_1?autoReconnect=true&useSSL=false']
```

Try to escape it:

```
--spring.datasource.url=jdbc:mysql://$MYSQL_PORT_3306_TCP_ADDR/app_1?autoReconnect=true\&useSSL=false']
```

The ampersand denotes the shell to start your process in the background. That's exactly what's happening on your local machine. If you start your jar, the process should start in foreground... and the prompt shouldn't return directly.

Highest Rated Answer:

To keep it simple and clean, we have added database properties in the database.properties

## mongo db database configuration

```
spring.data.mongodb.database=abc-auth spring.data.mongodb.host=192.168.2.2 spring.data.mongodb.port=27017 spring.data.mongodb.username=
```

We push this file while running the Docker, so only database properties will get override with the existing application.properties

```
ENTRYPOINT ['java', '-jar', '/home/docker/service/abc.jar', '--spring.config.location=application.properties']
```