Title: Testcontainers with Spring actuator

Post Body:

I would like to run a container with testcontainers on a network. My containers contains a spring application, which has an actuator endpoint to express its state, it is on:

```
/actuator/health
```

My container looks like the following:

```
private final static Network network = Network.newNetwork();  private static final GenericContainer<?> myContainer = new Gener
```

But when i start this container I get the following error:

```
Caused by: org.testcontainers.containers.ContainerLaunchException: Timed out waiting for URL to be accessible (https://localho
```

But if I curl this url (in this example: `https://localhost:33092/actuator/health`) while after the application inside the container has started but I am waiting for the evaulation, I get something like this:

```
HTTP Status Code: 200 {     "status": "UP" }
```

What am I missing? Any ideas? Is it a bug?

Accepted Answer: None

Highest Rated Answer:

Ok, so you'll have to decompose the problem.

I just built an example Spring Boot app with dependencies and packed it in an image:

```
implementation 'org.springframework.boot:spring-boot-starter-actuator' implementation 'org.springframework.boot:spring-boot-st
```

Nothing else, I downloaded it from https://start.spring.io/

The following test class works (I took your code as an example):

```
import lombok.extern.slf4j.Slf4j; import org.junit.jupiter.api.Test; import org.testcontainers.containers.GenericContainer; im
```

I can also put a break on System.out.println and while it's suspended curl also works

```
~  curl http://localhost:32776/actuator/health {"status":"UP"}■
```

I don't really have a straight guess right now what is happening in your case, but you have to decompose you problem layer by layer (getting rid of tls, network, etc) or build it up starting from scratch with the most simple image and adding features one by one.