Title: Spring Boot Build Image with Health Check

Post Body:

I'm building my Docker image using Spring Boot's built in Gradle `:bootBuildImage` task, which is quite convenient, because I don't have to maintain my own `Dockerfile`.

The Gradle task uses the Paketo Bionic Base Stack under the hood and will build a layered Docker image just fine.

Now, some orchestration engines like Docker Swarm (or simply Docker Compose for dev purposes) execute health checks *within* the container. Unfortunately, however, the resulting Spring Boot Docker image doesn't have any health checker tools like `curl` or `wget` installed, so something like

```
version: '3.7'  services:   springBootApp:     image: my/springboot/docker-image     healthcheck:       test: ["CMD-SHELL", "c
```

in `docker-compose.yml` will fail. (I checked that actuators themselves are working fine)

I know that `curl` or `wget` aren't ideal. I was actually hoping that the Paketo Builder would pick up something like this Health Checker BuildPack.

Is there a way to configure my `bootBuildImage` Gradle task to include that (or a similar) additional BuildPack?

As mentioned above, I'm looking for an easy to maintain solution and don't want to write my own `Dockerfile` to be able to profit from all the baked in best practices Paketo offers.

Accepted Answer:

The Health Checkers buildpack hasn't been added to the main builders just yet, but it is fully available to use with your apps.

Instructions

Modify your `build.gradle` file:

```
tasks.named("bootBuildImage") {   environment["BP_HEALTH_CHECKER_ENABLED"] = "true"   buildpacks = ["urn:cnb:builder:paketo
```

This will set an env variable at build time instructing the health-checkers buildpack to participate. It also tells Spring Boot build tools to append the health-checkers buildpack to the list of Java buildpacks that ship in the builder by default.

NOTE you may want to put in a particular version of health-checkers, that's up to you. It supports standard tag conventions like `:1`, `:1.4`, `:1.4.0` and `:latest` so you can pin to latest major/minor or a specific version.

Build. `./gradlew bootBuildImage`. You'll see the health checkers buildpack included in detection.

```
[creator]     ===> DETECTING [creator]     7 of 27 buildpacks participating [creator]     paketo-buildpacks/ca-certificates
```

Optional. Run `pack inspect <image>` against the produced image. You can see that health checkers has contributed a `health-check` process type.

```
Processes:   TYPE                  SHELL        COMMAND        ARGS                                               WORK DIR
```

Run your app. `docker run -d --health-cmd /cnb/process/health-check -p 8080:8080 -e THC_PATH=/actuator/health <image>`.

The health checker that's used by default is tiny-health-checker. It has other env variables you can configure but for a Spring Boot app you just need to set the actuator path (unless you have management port set differently, then set `THC_PORT=<management-port>` also.

If you are a Maven user, the process is very similar. Just substitute these instructions for step #1 and then #2 build with `./mvnw spring-boot:build-image`.

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

Highest Rated Answer: None