

Title: Run spring boot docker image with a different configuration in docker compose

Post Body:

Hello I'm starting with docker and docker compose and I have the following problem:

I'm working in a spring micro services architecture where I have one configuration service, one discovery service, one gateway service and multiple resource services. To run these services, I build jar files, which I place in separated folder per service with their config files (application.yml and bootstrap.yml): e.g:

config-service/

- config-service.jar
- application.yml

discovery-service/

- discovery-service.jar
- bootstrap.yml

gateway-service/

- gateway-service.jar
- bootstrap.yml

crm-service/

- crm-service.jar
- bootstrap.yml

This works so far on my server.

Now I want to deploy my services in different environments as docker images (created with mvn build image and buildpack) using docker compose, where the configuration files vary depending on the environment. How can I deploy a service as a container using an existing image but with a different configuration file?

Thank you in advance!

Accepted Answer:

There are a few possibilities when handling configuration in a containerized environment.

One of the options is that Spring boot allows you to use environment variables for each application property. For example, let's say you have a `spring.datasource.url` property, in that case you could also define that property by setting a `SPRING_DATASOURCE_URL` environment variable:

```
version: '3.8'  services:    my-spring-boot-app:      image: my-image:0.0.1      environment:        - SPRING_DATASOURCE_URL=jdbc:
```

Alternatively, you could use volumes to put an external file on a specific location within a container:

```
version: '3.8'  services:    my-spring-boot-app:      image: my-image:0.0.1      volumes:        - ./my-app/bootstrap.yml:/etc/my-ap
```

In this example I'm copying `bootstrap.yml` from a relative folder on my host machine, to `/etc/my-app` within the container. If you put these files within the same folder as your JAR file, you can override the configuration.

Highest Rated Answer: None