

Title: Using external config in spring boot application within docker-compose

Post Body:

I've got a spring boot application which uses `com.spotify.dockerfile-maven-plugin` to build a docker image of my application `org.rtu/some-importer`

My docker-compose.yml is:

```
version: '3' services:  some-importer:    image: org.rtu/some-importer    build: .    zookeeper:    image: wurstmeister/zoo
```

How can I say the during docker-compose up that it should be used an external `config.properties` from `/data/some-importer/config` folder?

Accepted Answer:

As mentioned in [the comments](#), the first step is to [mount a host directory in a docker container](#) (just like you did for Kafka). For example, you can use:

```
version: '3' services:  some-importer:    image: org.rtu/some-importer    build: .    # Adding a volume/mount    volumes:
```

This will map the `/data/some-importer/config` folder to `/config` in your Docker container.

**NOTE:** The linked answer also mentions that you can add it within your `Dockerfile` using `ADD`. However, this will add it to the image itself. If you make a change to the configuration, you'll have to rebuild your image to make those changes work.

The next step is to tell Spring boot to use this configuration file. If you want a completely customised location (eg. `/config/config.properties`), then you can use the [spring.config.location](#) parameter during startup.

**NOTE:** Spring boot will automatically pick up your configuration if it's located in [certain folders](#). Otherwise you'll have to configure it with `spring.config.location`.

I don't know how your image looks like, but you should be able to do something like this:

```
ENTRYPOINT [ 'sh', '-c', 'java -jar /app.jar --spring.config.location=${CONFIG_LOCATION}' ]
```

I'm using an environment variable called `$CONFIG_LOCATION` here, which makes it easier to customise the location by using environment variables. For example, you can add the following in your `docker-compose.yml` file:

```
version: '3' services:  some-importer:    image: org.rtu/some-importer    build: .    volumes:    - /data/some-importer
```

Highest Rated Answer: None