Title: docker-compose - externalize spring application.properties

Post Body:

I have a spring boot application that connects to a mongo db and deployed the app with docker. I am using this docker-compose.yml file, which works fine:

```
version: '2' services:   db:       container_name: app-db       image: mongo       volumes:       - /data/db:/data/db
```

Currently, the app is using the application.properties file embedded in the spring app docker image (spring-app). How do I externalize / pass-in the application.properties file using docker-compose?

Thank you for your help

Accepted Answer: None

Highest Rated Answer:

You must make use of the Spring Profiles to define the environment variables depending on your requirement.

```
server:     port: 9000 ---  spring:     profiles: development server:     port: 9001  ---  spring:     profiles: production se
```

Reference:
https://docs.spring.io/spring-boot/docs/current/reference/html/howto-properties-and-configuration.html#howto-change-configuration-depending-on-the-environment

You can define which profile needs to be picked up during the runtime.

```
version: '2' services:   db:       container_name: app-db       image: mongo       volumes:       - /data/db:/data/db
```

But this will require you to rebuild the docker image if there is a change in the configuration which is not ideal. Here comes the Spring Cloud Config (Vault) comes in handy which helps you to externalize your configuration.

http://cloud.spring.io/spring-cloud-static/spring-cloud-config/1.3.0.RELEASE/