

Title: Switching spring profile based on docker environment variable does not work

Post Body:

*docker-compose.yml*:

```
services:  server:      image: server:latest      environment:      - SPRING_PROFILES_ACTIVE=dev      ports:      - 8080:8080
```

*Dockerfile*:

```
FROM openjdk:8-jre-alpine  ENV SPRING_OUTPUT_ANSI_ENABLED=ALWAYS \      SLEEP=0 \      JAVA_OPTS='' \      RUN adduser -D -s /bin/sh
```

*entrypoint.sh*:

```
#!/bin/sh  echo 'The application will start in ${SLEEP}s...' && sleep ${SLEEP}  exec java ${JAVA_OPTS} -Djava.security.egd=file
```

I have 3 application.yml: application.yml, application-dev.yml and application-prod.yml which differ from database address.

But when I run `docker-compose up`, the server always use the default setting even I mentioned as in *docker-compose.yml* that the active profile is dev.

I'd like to know how to enable different profile in docker-compose file. Thanks.

**edit:** the *server.jar* file is built using assemble of gradle.

Accepted Answer:

Add a command: name space like below:

```
services:  server:      image: server:latest      environment:      - SPRING_PROFILES_ACTIVE=dev      ports:      - 8080:8080
```

After your container runs this will be appended to your entry point and spring boot will pickup this profile. It will be executed like :

Java -jar yourJar.jar --spring.profiles.active = prod.

And if you want to run your project app in various environment of your company and for various environment you have different kubernetes cluster configured then configure this setting differently on different kubernetes environment for your app . Means while running your docker image in qa environmner qa Kubernetes cluster will pass --spring.profiles.active = qa and similarly staging and Prod cluster.

And even if you you want to use environment variable you are using then use

```
exec java ${JAVA_OPTS} -Dspring.profiles.active={your environment variable name describe in docker compose} -Djava.security.egd
```

Check this on github: [https://github.com/vaneetkataria/MicroService\\_Architecture/blob/master/docker-compose.yml](https://github.com/vaneetkataria/MicroService_Architecture/blob/master/docker-compose.yml)

Highest Rated Answer:

.yaml files are handled differently than .properties files.

properties will look for a file with the active profile in the name:

application-dev.properties

application-prod.properties

...

yaml files, however you store all profiles in a single file separated by '---'.

Example [using your existing docker-compose.yaml]:

application.yaml:

```
spring:      application:      your-application-name      [other global application settings]      ...      --- spring:      p
```