Title: How do I keep the Eureka server url dynamic in the Eureka Client while using docker?

Post Body:

I am using the fabric8 docker-maven-plugin to build image for my Spring boot microservices.

```
<groupId>io.fabric8</groupId> <artifactId>docker-maven-plugin</artifactId>
```

The problem is that while running the application in docker containers I have to specify the Eureka Server Container name to Eureka Client. But if I run it directly as a 'Spring Boot APP' I have to use 'Localhost:8761/Eureka'. Is there a way to make it work both with/without docker something like given below ?

```
eureka:    client:        service-url:        defaultZone: ${EUREKA_SERVER:http://localhost:8761/eureka}
```

I am not able to pass the value of 'EUREKA_SERVER' from the fabrib8 plugin. I have tried the below code to pass the value but it does not work.

```
<docker.env.JAVA_OPTS>-DEUREKA_SERVER=http://discovery:8761/eureka</docker.env.JAVA_OPTS>
```

Accepted Answer:

Spring can pickup Environment Variables. So if you add Environment Variables to the Docker Container that Spring Boot is running in, they will work. This avoids the need to provide a static URL up front.

If you use Docker Compose, it could look like this:

```
services:    eureka:        image: springcloud/eureka        container_name: eureka        ports:        - '8761:8761'        networks:
```

Note: Since Environment Variables are not YAML, you need to change the format a bit.
https://docs.spring.io/spring-boot/docs/1.5.5.RELEASE/reference/html/boot-features-external-config.html#boot-features-external-config-relaxed-binding

Highest Rated Answer: None