Title: How to set 'spring.datasource.url' inside a Docker Container

Post Body:

I created a Spring Boot application which uses a MySQL database. I use a docker-compose to launch the database.

```
 services:   adminer:     image: adminer     restart: always     ports:       - 8888:8080   db:     image: mysql:latest     re
```

The Spring Boot Application (Backend) currently does not use Docker, I run it inside Eclipse. Before launching the Backend I have to grep the Docker Container for IPAddress:

```
docker inspect mysql_ex_db_1 | grep 'IPAddress'
```

which results something like this (this exact address changes time-to time)

```
                "IPAddress": "",                     "IPAddress": "172.21.0.2",
```

Then I take this value and I set `spring.datasource.url` inside Eclipse in the file `Application.properties` with it.

```
spring.datasource.url=jdbc:mysql://172.21.0.2:3306/employee_management_system?allowPublicKeyRetrieval=true&useSSL=false&create
```

After this I can launch the Backend in Eclipse the Connection to database is there, everything works.

Now I want to move the launching of Backend from Eclipse to the same docker-compose file I use to launch the database. Therefore I built an image, and appended the docker-compose file:

```
version: '3.1'  services:   adminer:     image: adminer     restart: always     ports:       - 8888:8080   db:     image: mysq
```

In this case how can I configure the IPAddress in `spring.datasource.url`? The exact IPAddress changes whenever I re-launch the mysql containers.

```
spring.datasource.url=jdbc:mysql://172.21.0.2:3306/employee_management_system?allowPublicKeyRetrieval=true&useSSL=false&create
```

So what should I write instead of '172.21.0.2' ?

I tried **localhost** here but it does not seem to work.

Accepted Answer:

First of all, you can set environment variables like `spring.datasource.url` outside of your docker image. This allows you to dynamically set these variables according to your deployment needs (like connecting to a dev or prod database).

All docker containers running from your docker-compose file run in the same virtual network and their service names correspond to their hostnames within this network. When you want to access your database from your dockerized spring backend the hostname and port will be `db:3306`. You can overwrite `spring.datasource.url` in your docker-compose file by introducing an environment variable like:

```
version: '3.1'  services:   adminer:     image: adminer     restart: always     ports:       - 8888:8080   db:     image: mysq
```

Highest Rated Answer:

Add this env variable to your `backend` in `docker-compose`:

```
backend:   ...   environment:     - SPRING_DATASOURCE_URL=jdbc:mysql://db:3306/employee_management_system?allowPublicKeyRetrie
```