Title: Connect from one Docker container to the other one

Post Body:

I am running a Java app inside a Docker container which is supposed to connect MySQL inside the other container. Trying multiple options suggested in the forms, nothing really works. Here is my Docker Compose file:

```
version: '3' services:    app:     build:       context: ./       dockerfile: /src/main/docker/Dockerfile     image: app1
```

Where `DB_HOST=Imrans-MacBook-Pro.local` is my laptop's name. This did not work. Some suggest that the container name can be used so tried `DB_HOST= mysql`, never worked.

The only thing works from times to time when I pass the laptop's IP address, which is not I want to do. So, what is a good way to create communication between those containers?

Accepted Answer:

I have, in the past, gotten this to work without explicitly setting the host networking part in Docker Compose. Because Docker images inside a Docker Compose File are put into a Docker Network with each other, you really shouldn't have to do anything to get this to work: by default you should be able to attach into the container for your Spring app and be able to `ping mysql` and have it work out.

Highest Rated Answer:

The mysql is running in the container so there are two things that you should consider here:

> If the mysql is running in the container then you will need to link the app container to the mysql container. This will allow them to talk to each other using docker's inter container communication. The containers talk to each other using hostnames to resolve their respective internal IP addresses. See later in my answer I will show you how to get the two containers to communicate with each other using a compose file.
>
> The mysql container should make use of a docker volume to store the database. This will allow you to store the database and related files on the file system of the host (server or machine where the containers are running on). The docker volume will then be mounted as a directory in the container. Thus the container can now read and write to a directory on the machine where the docker containers are running on. This means that even if the containers are all deleted or removed you will still have the database data persist. Here is a nice beginner friendly article on docker volumes and using them with MySQL:

https://severalnines.com/blog/mysql-docker-containers-understanding-basics

Container communication using only docker without compose:

You have container 'app' and 'mysql', you want to be able to access 'app' on localhost and you want 'app' to be able to connect to mysql. How are you gonna do this? 1. You need to expose a port for container 'app' so we can access it on localhost. The docker containers have their own internal network and it is closed to you unless you expose some ports with docker.

2. You need to link the 'mysql' container to 'app' without exposing 'mysql' 's ports to the rest of the world.

This config should work for what you want to achieve:

```
    version: '2'     services:       app:         build:           context: ./           dockerfile: /src/main/docker/Dockerf
```

Now you can just start it up with:

```
$ docker-compose up
```

If you ran this before then just make sure to run this first before running docker-compose up:

```
$ docker-compose down
```

Let me know if that helps.