Title: Docker MySQL - can't connect from Spring Boot app to MySQL database

Post Body:

What I'm trying to do is, connect from my spring-boot app to mysql database in Docker. Each in their own container.

But I must be having something wrong because I can't do it.

To keep it simple :

application-properties :

```
# URL for the mysql db spring.datasource.url=jdbc:mysql://workaround-mysql:3308/workaround?serverTimezone=UTC&max_allowed_pack
```

docker-compose for MySQL:

```
version: '3' services:   workaround-mysql:     container_name: workaround-mysql     image: mysql     environment:       MYSQL_
```

So pretty simple right ? Database I start with `docker-compose up`:

All seems to be working fine so far.

Now that I have db started, to the application, this is its `docker-compose.yml`:

```
version: '3' services:    workaround:     restart: always     # will build ./docker/workaround/Dockerfile     build: ./docker/
```

For its `Dockerfile` I use Linux Alpine and Java.

```
FROM alpine:3.9 ....add java... RUN apk update RUN apk add dos2unix --update-cache --repository http://dl-3.alpinelinux.org/al
```

Super simple. Now let's start the application :

Unknown host, so let's try the IP then :

```
    docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' workaround-mysql  # URL for the mysql db spri
```

Now I get timeout:

As you can see I get error. What is wrong with my setup and how to fix this? Either I have unknown host exception or Refused to connect or connection timeout.

I have tried:

- Using ip of a container in my application.properties, didn't work
- Different ports for MySQL and application
- Different images and versions of MySQL
- Having everything in one docker compose with wait
- timer for database.
- Minimal setup with https://github.com/hellokoding/hellokoding-courses/tree/master/docker-examples/dockercompose-springboot-mysql-nginx Also resulted in communication link failure, Site was accessible but I doubt that db was connected properly.

Notes:

I run this all on one computer I use port 3308 because I have local MySQL db at 3306.

Here is `docker ps -a`

@Vusal ANSWER output :

Only thing different from code in answer I did wait for database to be ready 30 seconds

```
command: /bin/bash -c 'sleep 30;mvn clean spring-boot:run;'
```

Accepted Answer:

Try this `docker-compose.yml`:

```
version: '3' services:   workaround-mysql:     container_name: workaround-mysql     image: mysql     environment:       MYSQL_
```

And update your `application.properties` to use the next JDBC connection url:

```
spring.datasource.url=jdbc:mysql://workaround-mysql:3306/workaround?serverTimezone=UTC&max_allowed_packet=15728640
```

It should work when both containers in the same docker-compose file, because docker-compose creates default network for containers, so they can resolve each other by name.

Highest Rated Answer:

What you haven't tried so far is running both containers on the same Docker network.

First, forget about IP addressing - using it should be avoided by all means.

Second, launch both compose instances with the same Docker network.

Third, **do not** expose ports - inside bridge network all ports are accessible to running containers.

Create global network

```
docker network create foo
```

Modify both compose files so that they use this network instead of creating each one its own:

```
version: '3.5'  services:   ....   networks:    default:      external: true      name: foo
```

Remove `expose` directives from compose files - inside one network all ports are exposed by default

Modify connection strings to use default `3306` port instead of `3308`

Enjoy