Title: Connect To Docker Compose MongoDb Via Spring boot application

Post Body:

My web app can't connect to the MongoDB container

here are my application.yml

```
spring:   data:     mongodb:       uri: mongodb://mongo:27017       host: mongo       port: 27017       database: my-db-name
```

and this is my Docker-Compose

```
version: "3" services:    java:     build:       context: ./     ports:       - "8080:8080"     links:       - mongo     depen
```

and this is the Dockerfile wrote for running java

```
FROM openjdk:11 COPY ./code/lemon-backend/target/lemon-0.0.1-SNAPSHOT.jar /usr/src/ WORKDIR /usr/src/ EXPOSE 8080 CMD ["java",
```

I can't even build the application using these options I get this exception:

```
org.mongodb.driver.cluster: Exception in monitor thread while connecting to server mongo:27017
```

if possible try giving solutions with docker-compose, thanks

Accepted Answer:

OLD VERSION ANSWER

IMPORTANT NOTE: older versions of MongoDB ignore this configuration in `application.properties`, proceed ahead & use the new solutions I added

---

This workaround is used for old versions of spring and mongo that ignore the normal configuration (other than uri)

. *I had a warning that this property cant be resolved but hopefully, it worked :)*

```
dockerspring.data.mongodb.uri= mongodb://<your_mongodb_container_name>:27017/<name_of_your_db>
```

the mongodb part is not changeable but mongo before the port number is actually the name of the container witch you have specified in your docker-compose

---

SPRING BOOT SOLUTION

```
spring:   data:     mongodb:       host: <mongo-db-container-name>       port: <mongo-db-port>       database: <database-name>
```

DOCKER SOLUTION

In Your Dockerfile Add This Option For Executing Java

```
ENTRYPOINT [“java”,”-Dspring.data.mongodb.uri=mongodb://mongo:27017/name_of_your_db”, “-Djava.security.egd=file:/dev/./urandom
```

Linking Java And Mongo Containers + Giving Them Names

here this is my final docker-compose.yml, I hope that it helps you

```
version: "3" services:    java:     build:       context: ./     ports:       - "8080:8080"     container_name: java     links
```

Compare this version and the one specified in the question carefully

Highest Rated Answer: None