Title: SpringBoot in Docker not connecting to Mongo in Docker
Post Body:

I have a Spring Boot Application and developed it with a mongo db which was running in brew services.

To get a connection to the db I just had to put the following into application.properties in Spring Boot

```
spring.data.mongodb.uri=mongodb://localhost:27017/db
```

changing the application properties to

```
spring.data.mongodb.uri=mongodb://mongo:27017/db
```

didtn't change anything, same Error as before.

Now I'm trying to put the SpringBoot Application and the MongoDB into Docker-Containers, but cant get any connection working.

So this is my Dockerfile in the SpringBoot Application:

```
FROM openjdk:8-jdk-alpine VOLUME /tmp EXPOSE 8080 ADD /build/libs/dg-0.0.1-SNAPSHOT.jar dg-0.0.1-SNAPSHOT.jar ENTRYPOINT ['jav
```

This is my Docker-Compose.yml file:

```
version: '3'  services:  mongo:  container_name: docker-mongo  image: mongo:latest  ports:    - '27017:27017'  volumes:   - ./
```

After executing Docker-Compose with

```
docker-compose up
```

I get the following error: (this is the actual error message)

```
2019-07-08 23:10:19.990  INFO 1 --- [localhost:27017] org.mongodb.driver.cluster: Exception in monitor thread while connecting
```

Does someone know whats the problem here? In development environment it works like a charm.

Thanks in advance

Adding the following line to the Dockerfile

```
'-Dspring.data.mongodb.uri=mongodb://mongo:27017/dg'
```

into Entrypoint like this solved the connection issue

```
ENTRYPOINT ['java', '-Dspring.data.mongodb.uri=mongodb://mongo:27017/dg','-Djava.security.egd=file:/dev/./urandom','-jar','/dg
```

I was able to get connection to the db without the mentioned line above in the Entrypoint in the Dockerfile. I guess this is if you like to connect your db over 'links'

But now I was able to connect over Network, this is my code:

```
version: '3.6'  services:   mongo:  container_name: docker_mongo  networks:    - gateway  ports:    - '27017:27017'  hostname:
```

and the following in application properties

```
spring.data.mongodb.host=docker_mongo spring.data.mongodb.port=27017 spring.data.mongodb.database=db
```

So it looks like the connection is working over Network now. The same code did not work with Version 3.0

To prevent SpringBoot to connect automatically to mongo over localhost it's also necessary to exclude MongoAutoConfiguration!

```
@SpringBootApplication(exclude={MongoAutoConfiguration.class})
```

Thank you all for your help

Accepted Answer: None
Highest Rated Answer:

EDIT:

I've never done spring-boot development, but the error you are saying is being displayed may very well be unrelated to the mongo issue. However, here is an explanation as to why your mongo-connection is failing:

`docker-compose` creates a virtual network if one hasn't been specified in the file (like in your case).

All your applications run inside of this network, completely isolated from each other. As such, `localhost` in your spring-boot container actually refers to itself. Meaning your spring-boot application is expecting the `mongo` instance to be running inside of its container (which its not, it's in a different container).

This would have been fine when both the database and application was running on your laptop's network. But as mentioned, they are now running in the `docker-compose` network, in complete isolation.

However, `docker-compose` is really clever! It creates a DNS for each of your containers which uses the service-name (in your case `mongo` and `spring`) specified in your `docker-compose` file to allow for easy access to the containers inside of the network.

So, you should be able to change `spring.data.mongodb.uri=mongodb://localhost:27017/db`to `spring.data.mongodb.uri=mongodb://mongo:27017/db` and that should allow it to connect.