Title: Unable to reach tomcat, war file not deployed while integrating with docker

Post Body:

War file generated successfully using `mvn clean install`, Then tried with `docker compose up --build` in terminal.The logs in terminal shows that there is no tables created and unable to reach endpoints properly using postman, tomcat is in unreachable state, because of improper deployment of war file. I need a solution to connect the database with the server and to reach the endpoints. I don't find any helpful information to solve this issue.Help me to attain proper deployment.

[File structure]

pom.xml

```
<modelVersion>4.0.0</modelVersion> <parent>     <groupId>org.springframework.boot</groupId>     <artifactId>spring-boot-starte
```

Terminal logs:

```
sowbharnika@SowbharnikasMBP EcomApplication % docker compose up --build  [+] Building 4.7s (11/11) FINISHED
```

While debugging tomcat container logs doesn’t show any error stating the problem in reaching tomcat server.So it is difficult to debug.

Postman

[Endpoint status]

Docker Compose File

```
version: '2' services:   ctomcat:     build:       context: .       dockerfile: ./docker/tomcat/Dockerfile     ports:       -
```

Docker File for postgres

```
FROM postgres:10.2 ENV POSTGRES_USER postgres ENV POSTGRES_PASSWORD password ENV POSTGRES_DB ecomdb
```

Docker File for tomcat

```
FROM tomcat:8.0-jre8 ADD target/ecom.war /usr/local/tomcat/webapps ENV JPDA_ADDRESS 18000 ENV JPDA_TRANSPORT=dt_socket EXPOSE
```

application.properties

```
spring.datasource.url=jdbc:postgresql://cpostgres:5432/ecomdb spring.datasource.username=postgres spring.datasource.password=p
```

Accepted Answer:

In Docker compose file, pulled image for tomcat is having different version which does support particular java versions only. Therefore, either you change tomcat image version file which supports Java version 11(in pom file) or change Java version as mentioned below:

In POM file under properties do changes as such -

```
<properties>     <java.version>1.8</java.version> </properties>
```

> ***Error shown in terminal logs is tough to debug as it does not show any error at all instead of it, it completely ignores the artifactId if it can't be deployed.***

Note

Three things we need to make sure…

War should be copied to the correct directory inside tomcat as - ***(/usr/local/tomcat/webapps)***.

Make sure your java version matches the tomcat image you are are using.

Make sure your sprint boot application class is proper and has implemented ***SpringBootServletInitializer***

After doing above mentioned changes, project will get deployed successfully, will get connected to database & will access endpoints as well.

Highest Rated Answer: None