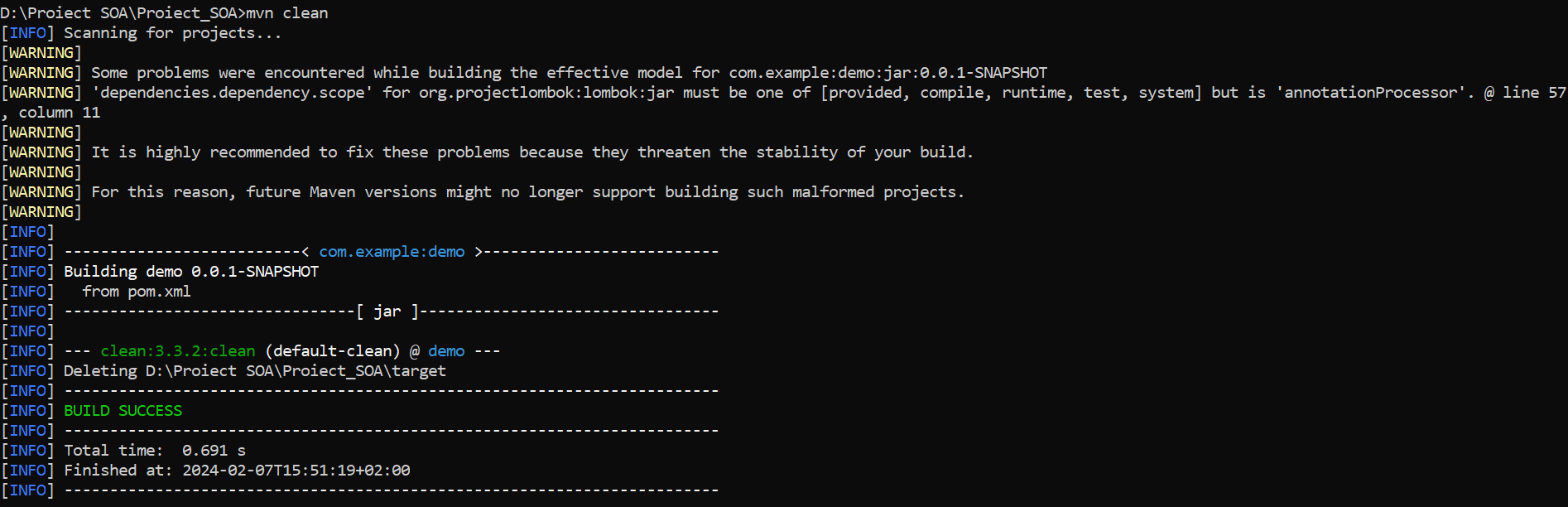
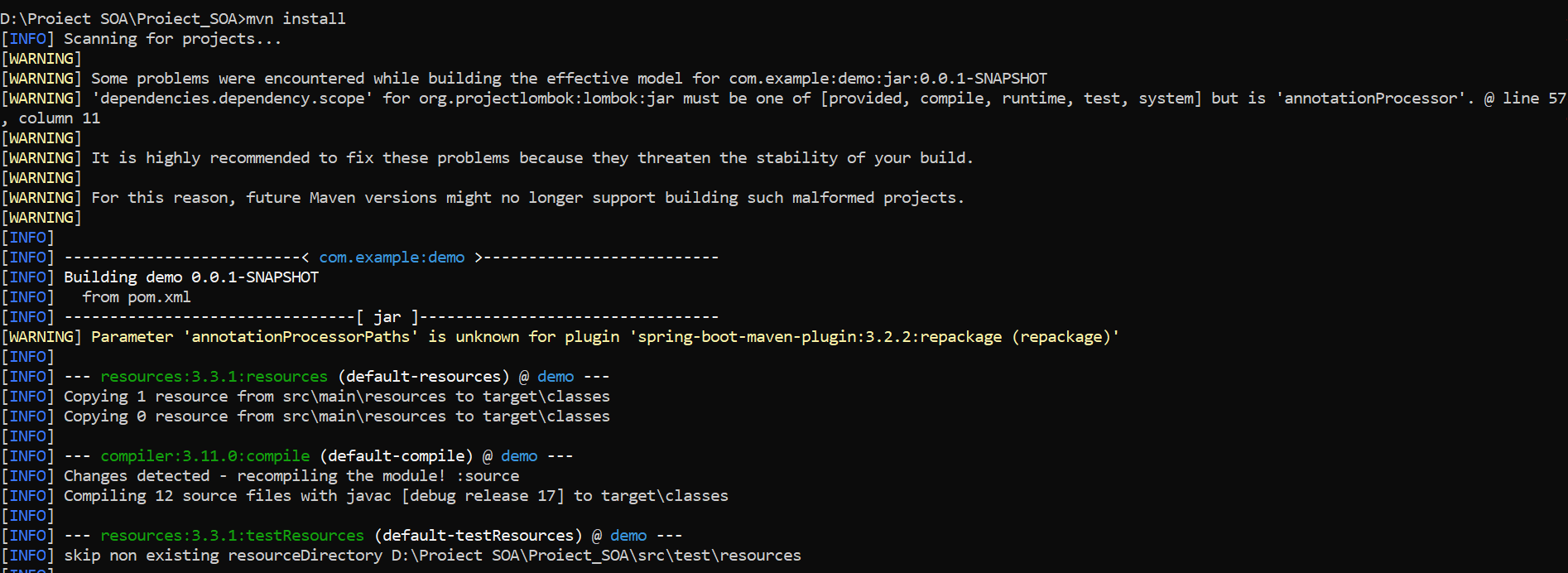
Deploying REST API + PostgreSQL database with Docker

The goal of this tutoria lis to present who I did the following steps:

1. Dockerize Spring Boot Application (my secured REST API)

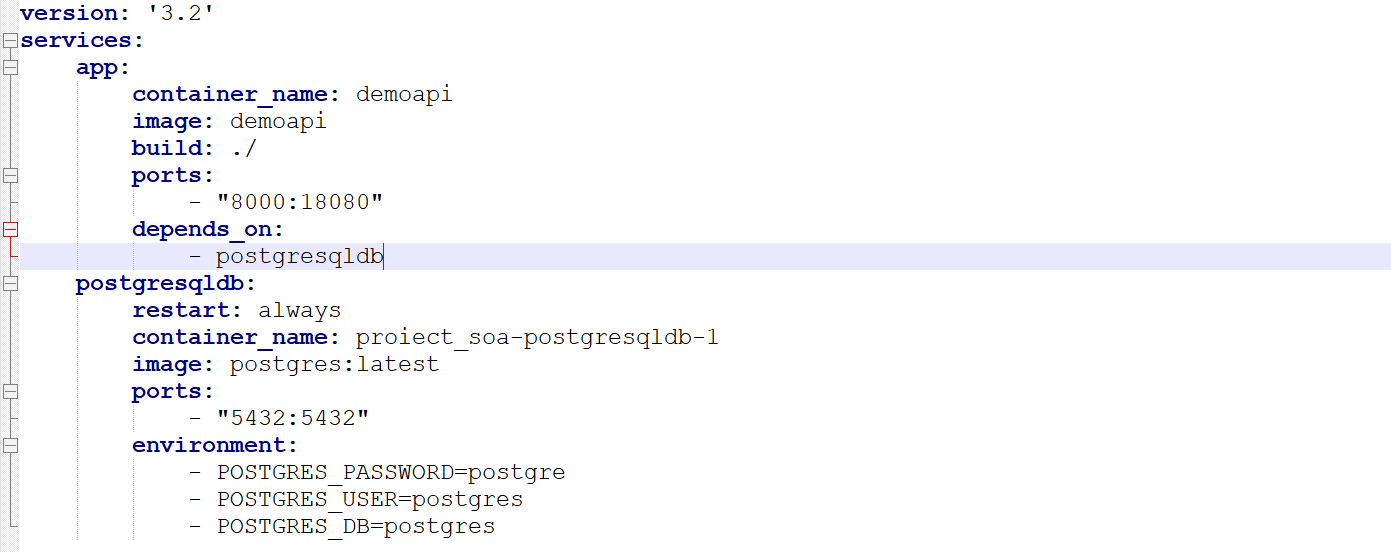
Giving the fact that I have used Maven, first 2 steps were to clean and install the project:





Since we’ll use a container of postgreSQL and not a local stored database, it is usefull to add ”-DskipTests=true” in the command line. This will prevent our build to fail as the test package cannot connect to the database yet.

Next, I created a docker-composed.yml file in order to be able to deploy the app and database at the same time:



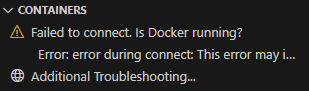
The link between the app and database it is made with the „depends\_on” variable. We can see that the docker container will run on the 8000 port and it will be linked with the 18080 port (that I set for my REST API in the application.properties file).

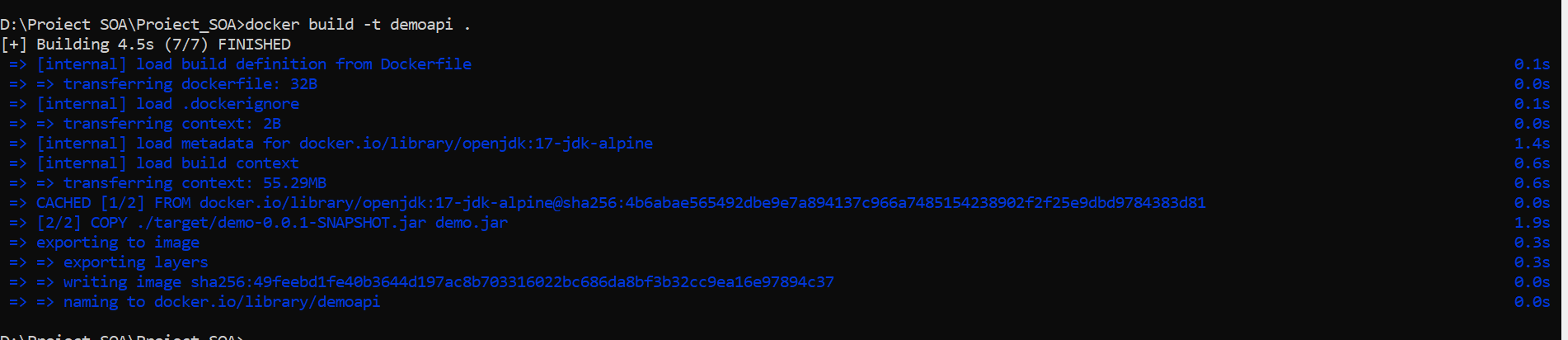


In order to use the postgre container and not the localhost address, in the same application.properties file, I have modifies the connection to the database: the commented line was for when I created the app, and now I replaced the IP adress for localhost with the name of the dockerized postgre container.

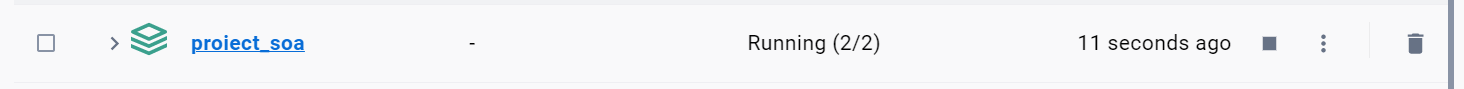


NOTE: Verify that your docker service is running in order not to have this error message:



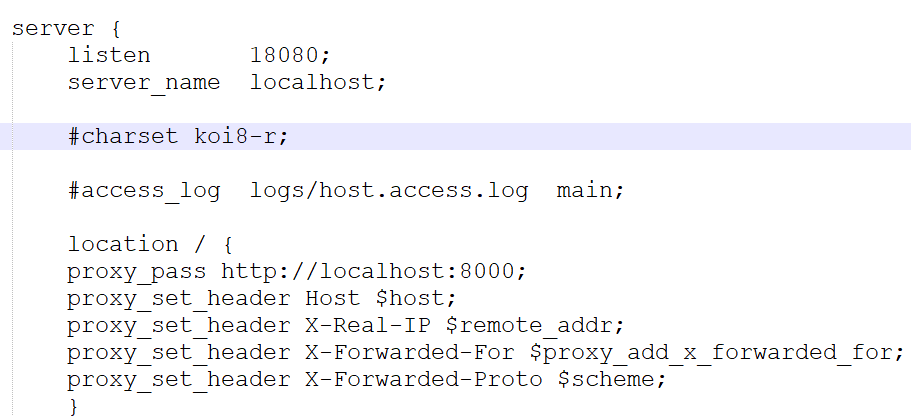
Now, I could have built the two separately, for example:   
 

But because I have created the composed file, I can run the command: docker-compose up, that will run both of the containers in the docker-compose file under the name proiect\_soa container:



1. Nginx configurations

In the nginx.conf, I have added the following:



Because my microfrontends are trying to fetch from the localhost:18080 port, I need to redirect them to the dockarized container where I now run my REST API.