The flow explains how to set up a Spring Boot web application (RPSP in this case) to work against DB (MySql in this case).

In general, connection parameters to DB will be passed as environment variables to web application. In Web will application code, you will read them and put them on the datasource instead of the parameters that were taken from configuration files.

1.

The MySql container should be configured as explained in details in my-sql-setup documentation project.

Say, it was started like this ---sudo docker run --name demo-mysql -e MYSQL\_ROOT\_PASSWORD=password -e MYSQL\_DATABASE=demo -e MYSQL\_USER=demo\_user -e MYSQL\_PASSWORD=demo\_pass -p 3306:3306 -d mysql

2.

RPSP image is created like this (should be executed from the directory containing RPSP and Dockerfile) ---sudo docker build -t shpboris/rpsp .

3.

To start the container do --- sudo docker run --name rpsp -e DB\_URL=jdbc:mysql://10.0.2.15:3306/rpsp -e DB\_USER=rpsp -e DB\_PASSWORD=rpsp -p 9999:8080 -d shpboris/rpsp

All parameters after “-e” sign in instruction above (in this case , DB connection parameters) are passed as environment variables to the Web app by the docker.

Pay attention to IP=10.0.2.15. It can’t be localhost or 127.0.0.1 – since this is resolved to app container IP itself !!! So access to DB will not work with localhost or 127.0.0.1 !!! To get real IP of Ubuntu within Virtualbox, follow steps on screens below and take “IP address”. <http://www.howtogeek.com/howto/17012/how-to-find-your-ip-address-in-ubuntu/>

4.

Last step is to read the parameters from #4 in Web app code and apply on the datasource.

The code for this can be found at the following path inside the current documentation mini project - \app-src\RPSP\src\main\java\com\emc\rpsp\config\ DatabaseConfiguration.java or inside the \app-src for the convenience.

The relevant code is this

**if**(org.apache.commons.lang3.StringUtils.*isNoneEmpty*(env.getProperty(DB\_URL))){

config.addDataSourceProperty("url", env.getProperty(DB\_URL));

**if**(org.apache.commons.lang3.StringUtils.*isNoneEmpty*(env.getProperty(DB\_USER))){

config.addDataSourceProperty("user", env.getProperty(DB\_USER));

}

**if**(org.apache.commons.la.StringUtils.*isNoneEmpty*(env.getProperty(DB\_PASSWORD))

{

config.addDataSourceProperty("password", env.getProperty(DB\_PASSWORD));

}

}