



Find 10

Follow

May 12, 2021 · 3 min read · Listen



How to connect a MySQL Docker Container to a Spring Boot Application running locally

If you do not want to install MySQL Server and MySQL Workbench directly into your Windows machine, you can obtain it from Docker, and still link it to your Spring Boot Application running locally. Here's how!

Create docker-compose.yml file

First, create a docker-compose.yml file in a directory you want.

```
version: '3.1'

services:
  db:
    image: mysql
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: 'rootpassword'
    ports:
      # port exposed : mysql port running inside container
      - "3306:3306"
    expose:
      # opens port 3306 on the container
      - "3306"
    volumes: # where the data is persisted
      - my-db:/var/lib/mysql
  adminer:
    image: adminer
    restart: always
    ports:
      - 9000:8080

volumes:
  my-db:
```

docker-compose.yml

There are 2 images we specified— “mysql” and “adminer”. Adminer allows us to conveniently manage the databases created without having to use MySQL Workbench. It can also work for other databases such as PostgreSQL and MongoDB.

Start your Docker desktop app

Don't miss out this step!

Run docker-compose up

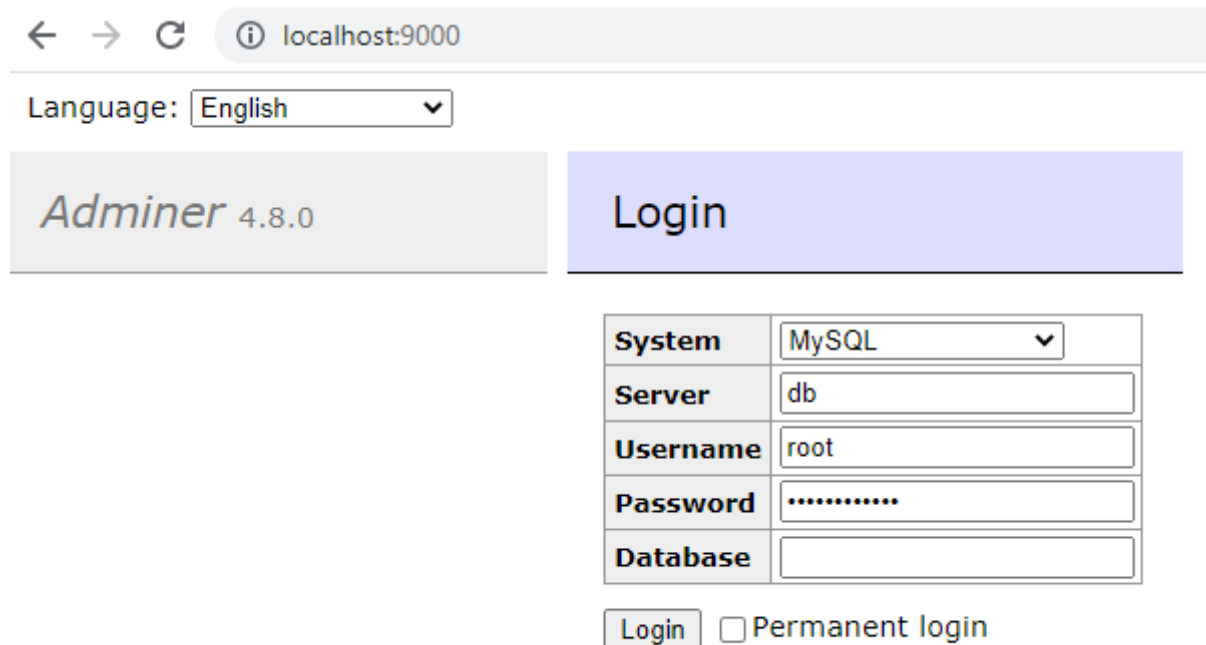
In the directory of the docker-compose.yml, run **docker-compose up**. This starts the MySQL and Adminer containers, bound to a command prompt terminal. You can see the logs there. On a command prompt, you can run **docker container ls -a** to view the containers in your system.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b2708946d044	mysql	"docker-entrypoint.s..."	About a minute ago	Up About a minute	0.0.0.0:3306->3306/tcp, 33060/tcp	springjava_jpa_db_1
5c6b9b0c836e	adminer	"entrypoint.sh docke..."	About a minute ago	Up About a minute	0.0.0.0:9000->8080/tcp	springjava_jpa_adminer_1

containers

In addition, you can head on to localhost:9000 to view the Adminer database management interface and login as the root user using the password defined in docker-compose.yml.





← → ↻ ⓘ localhost:9000

Language: English ▼

Adminer 4.8.0

Login

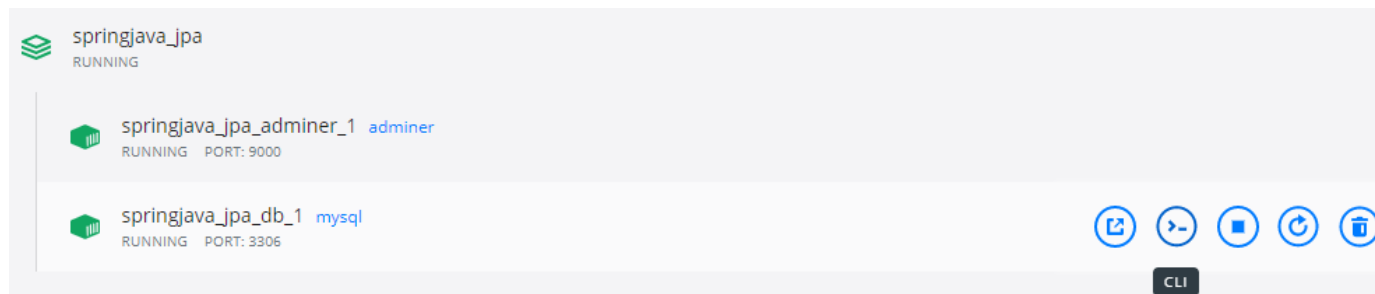
System	MySQL ▼
Server	db
Username	root
Password
Database	

Login ☐ Permanent login

Adminer interface @ localhost:9000

Start the interactive Bash shell in the MySQL Container

You can either do so via the Docker Desktop app (shown below), or by opening a new command prompt and entering **docker exec -it <container name or ID> bash**



opening the interactive Bash shell by clicking on the CLI icon, using the docker desktop app

Run MySQL Commands

If you want to run MySQL Commands through the MySQL Shell, first enter **mysql -uroot -p<root password>** where root password is what you defined in the docker-compose.yml file. This will bring you into the MySQL Shell.

```
# mysql -uroot -prootpassword
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 8.0.25 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Going from Bash to MySQL Shell

Here you can create a database, a user and grant him/her access to it.

```
mysql> create database testdatabase;  
Query OK, 1 row affected (0.02 sec)  
  
mysql> create user 'testuser'@'%' identified by 'testpassword';  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> grant all on testdatabase.* to 'testuser'@'%;  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> _
```

usual MySQL Commands

Configure your Spring Boot application.properties file

Now, in your Spring Boot application, define some properties to allow it to connect with the MySQL container.

```
1 spring.jpa.hibernate.ddl-auto=update  
2 spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3306/testdatabase  
3 spring.datasource.username=testuser  
4 spring.datasource.password=testpassword
```

application.properties file

Run your Spring Boot application. Then, head to your Adminer interface, click on the refresh button and you should see your new database appear. You can click

on it and see the tables and manage the database through the interface if you would like to.

Select: customer

10 items have been affected. 14:29:42 SQL command

Select data

Show structure

Alter table

New item

Select

Search

Sort

Limit

50

Text length

100

Action

Select

SELECT * FROM `customer` LIMIT 50 (0.001 s) Edit

<input type="checkbox"/> Modify	id	first_name	last_name
<input type="checkbox"/> edit	1	Jack	Bauer
<input type="checkbox"/> edit	2	Chloe	O'Brian
<input type="checkbox"/> edit	3	Kim	Bauer
<input type="checkbox"/> edit	4	David	Palmer
<input type="checkbox"/> edit	5	Michelle	Dessler

sample table in the Adminer interface

This is all, hope this was helpful! Do write down any of your thoughts as comments below.

Get an email whenever Find 10 publishes.

Your email

 Subscribe

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

[About](#) [Help](#) [Terms](#) [Privacy](#)