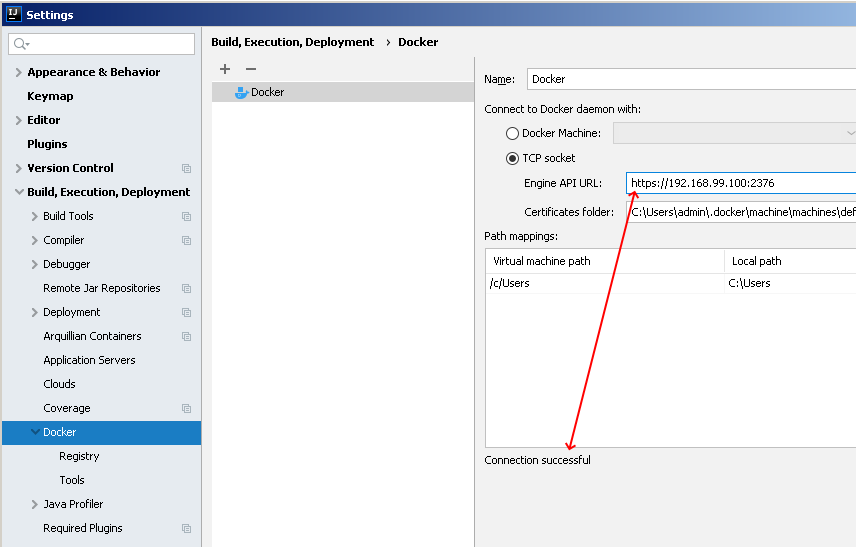
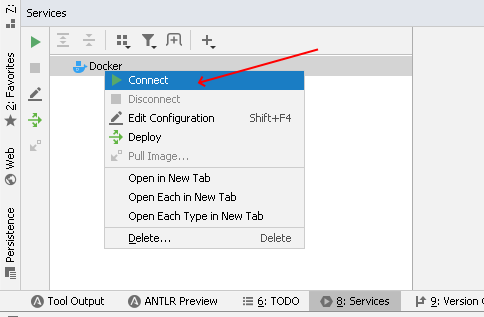
<https://www.youtube.com/watch?v=ck6xQqSOlpw>

1) Установить docker в Windows

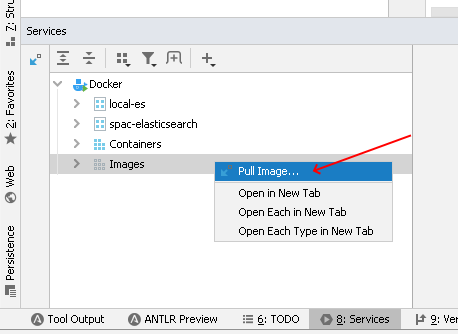
2) Настроить docker в Idea (должно быть connection successful)



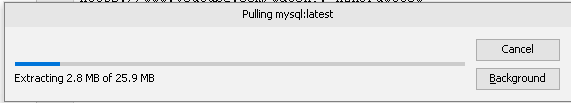
3) Connect to Docker



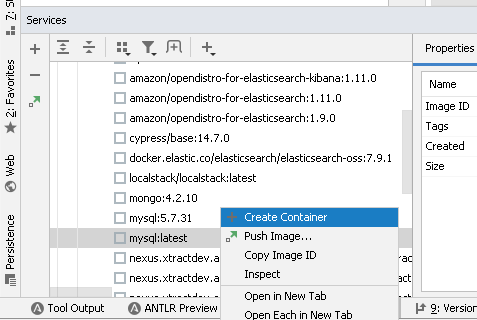
4) Pull Image...



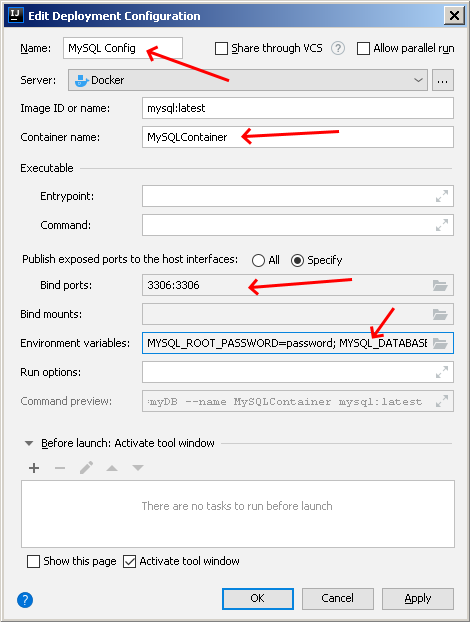
5) write mysql, enter



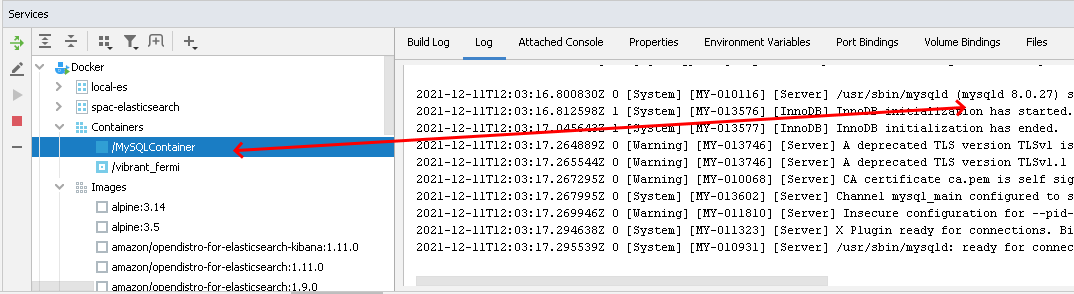
6) Создать контейнер из image



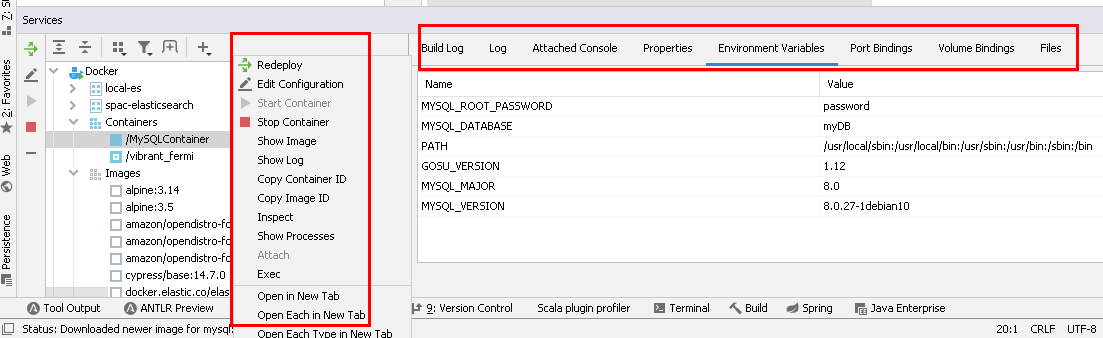
7) Изменить конфиг



8) Start container

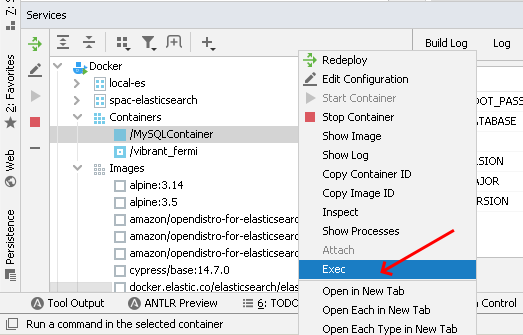


Настройки и манипуляции с контейнером:

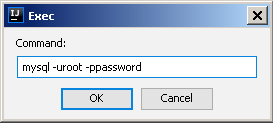


9) Run command on mysql

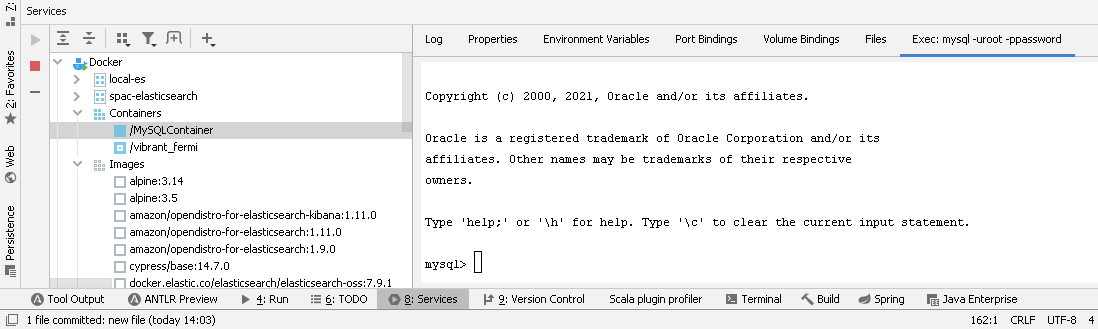
Click Exec



Enter



this will open mysql console:



Now can run commands:

mysql> use myDB;

Database changed

mysql> CREATE TABLE EMPLOYEES(

-> id INT NOT NULL PRIMARY KEY,

-> firstName VARCHAR(40) NOT NULL,

-> lastName VARCHAR(40) NOT NULL);

Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO EMPLOYEES(id, firstName, lastName)

-> values('1', 'Dalia', 'Abo Sheasha');

Query OK, 1 row affected (0.01 sec)

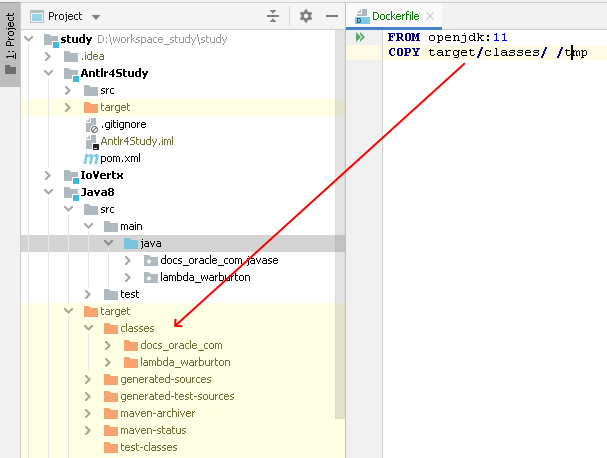
### 1.2. Create our own image

Build and run 'hello world' in image and docker container.

1) create docker file. Create file with name 'Dockerfile'. Content:

**FROM** openjdk:11 // image to use

**COPY** target**/**classes**/ /**tmp // what files to copy to image



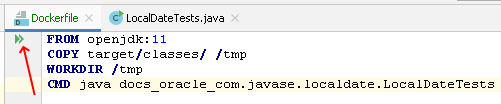
**WORKDIR /**tmp // switch to tmp directory as a work directory

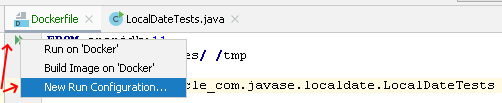
**CMD** java docs\_oracle\_com.javase.localdate.LocalDateTests

// now specify command for main class

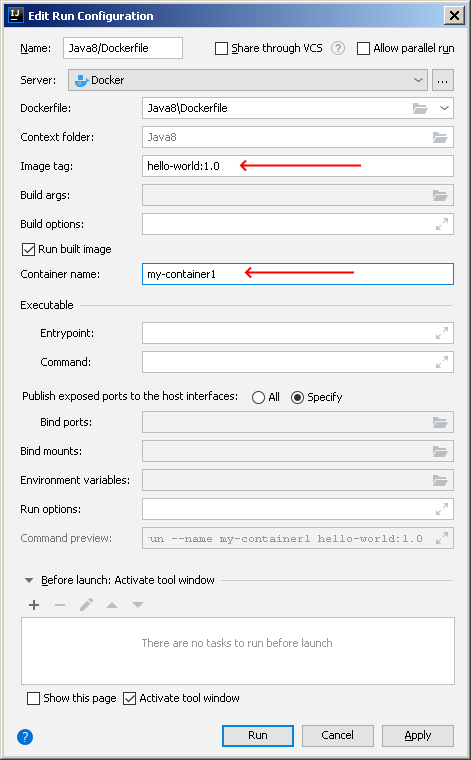
2) run image and run container

Click:

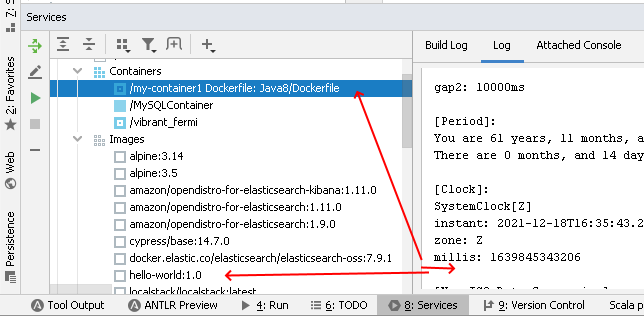




Write configuration:



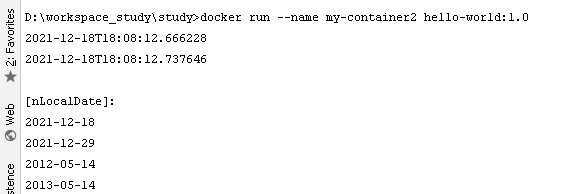
Run will build image and start application.



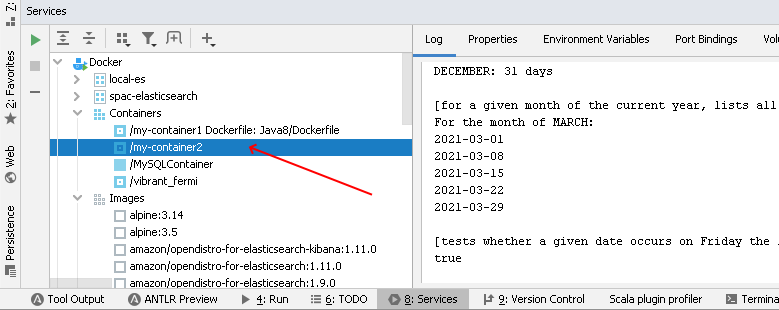
3) Command line to run any docker images

$ docker run --name my-container2 hello-world:1.0

this will run container:

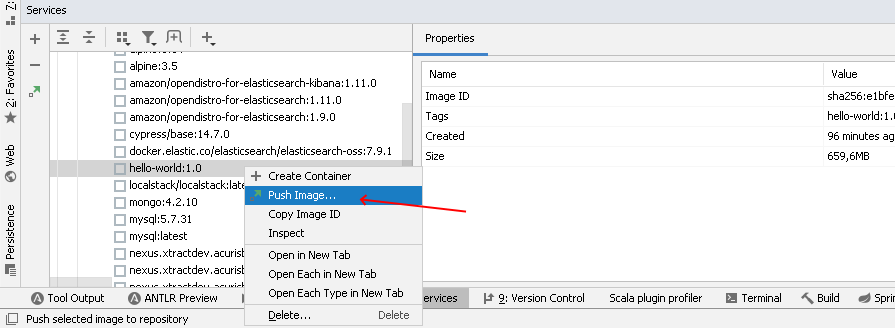


New container, under containers list:

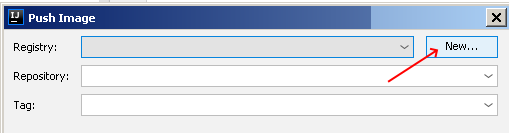


### 1.2. Push an image

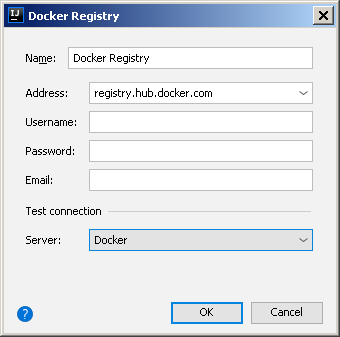
Push an image on my repository at docker hub.



Click new



Enter data



Then enter:

Repository: hello-world

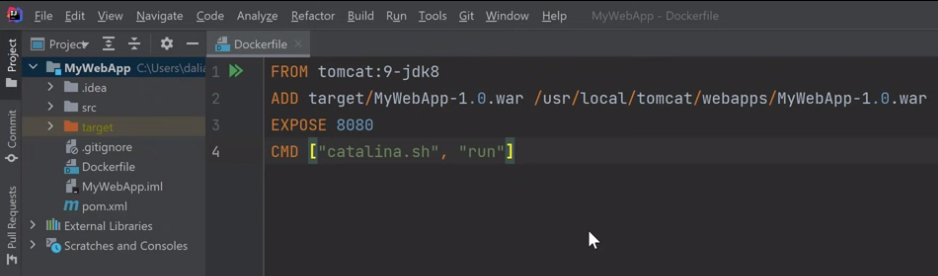
Tag: 1.0

Now https://hub.docker.com/repositories will show uploaded image. And other team members can pull image and run it.

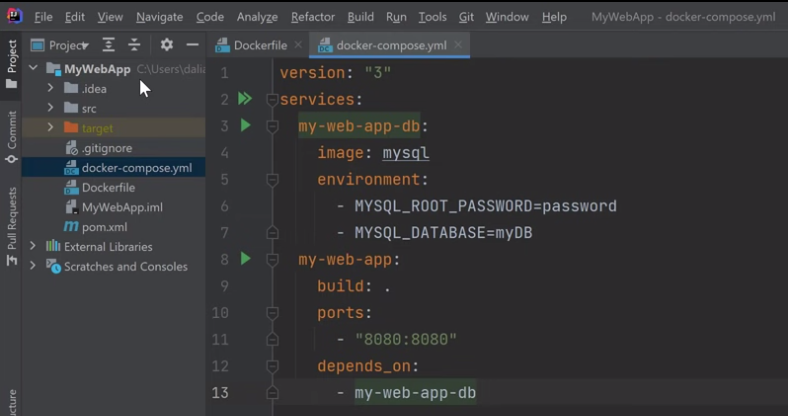
## 2. Docker compose

Have web application and want anyone else will use it.

1) Create docker file for an application image

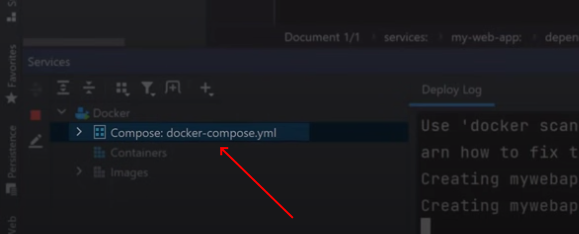


2) Create docker compose file (define all the services I need, the application to run)

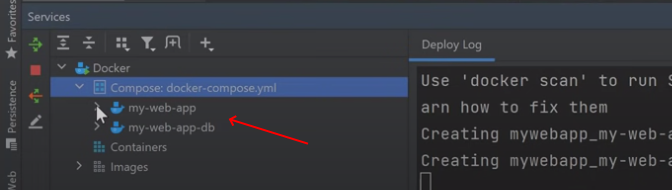


3) run docker compose app. Click on double green arrows.

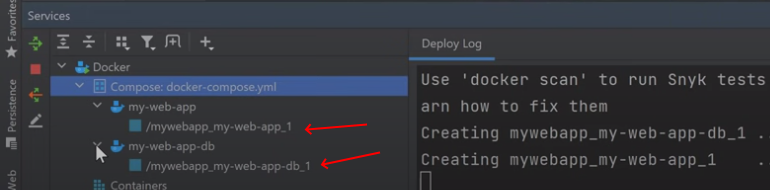
New type of node for docker compose which provides a grouping of our application services



If expand, two services we defined in docker compose file.

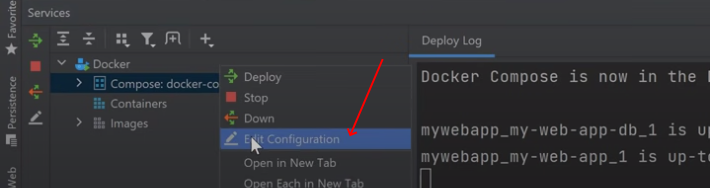


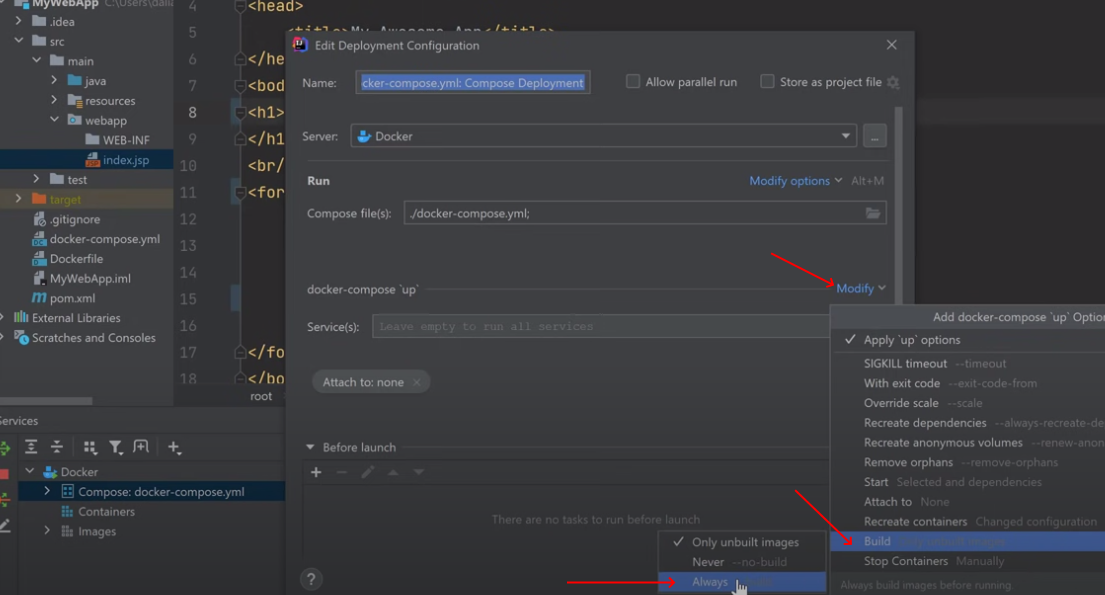
Under each service, we see running container for that service:



### 2.1. Refresh container

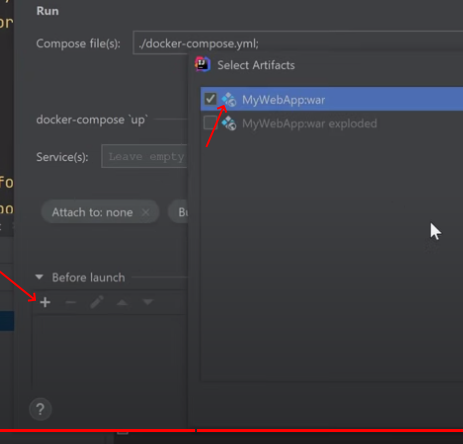
Need to update config, so application will refresh war before Deploy docker.

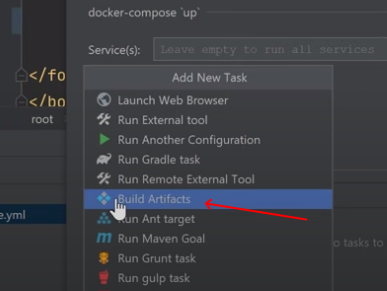




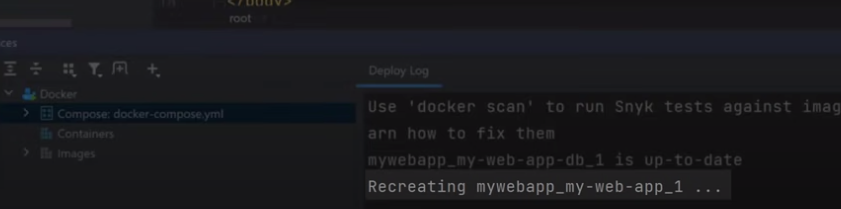
Now docker compose will rebuild my image, even if it already was built.

Also need to rebuild war file before each deployment.

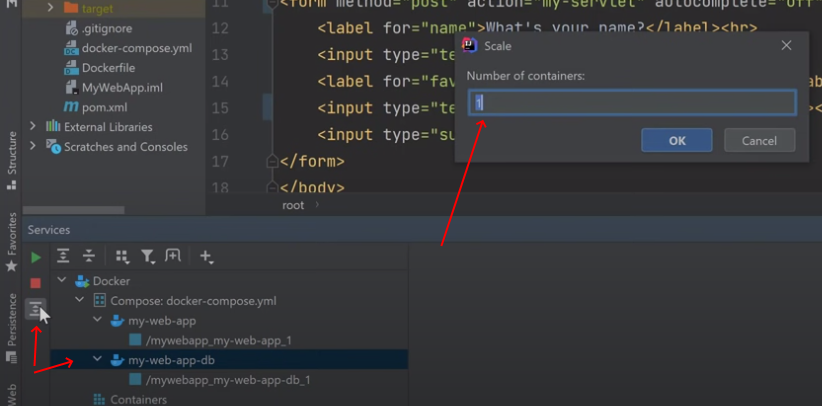




Now log shows, the container for web services was recreated:



### 2.2. Number of containers can be increased



Now 3 containers under database services:

