SOMETHING YOU NEED FOR BUILD PROJECT:

1. Install nodejs
2. Install node package manager (npm).
3. Install yarn.
4. Install java.
5. Install VSCODE.
6. Install INTELLIJ IDE.

\* ALL THE THINGS I JUST MENTIONED ON YOUTUBE HAVE THE EASIEST TO UNDERSTAND INSTALLATION INSTRUCTION, SO IF YOUR LAPTOP OR PC DOESN’T HAVE ANYTHING, JUST SEARCH WITH KEYWORD “HOW TO INSTALL ..... TO WINDOW/ MACBOOK”

I have write all statement below of file “ deployment.yaml “ you can see the code in there. The statement has “-s -f “ is statement for remove and remain for pulling container.

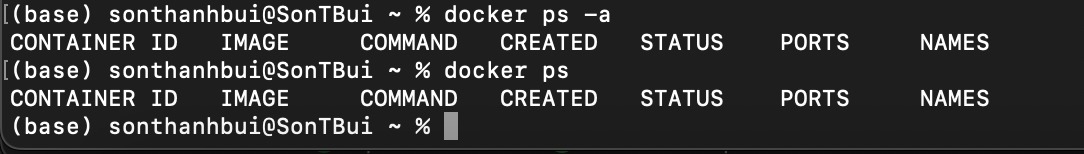
Correct statements pulling always have “ up -d”, it’s look like this:

“ docker-compose -f ./deployment.yaml up -d mysql8-container “



You can check all container by : “ docker ps “ for The container is running

and “ docker ps -a “ for all container you have.

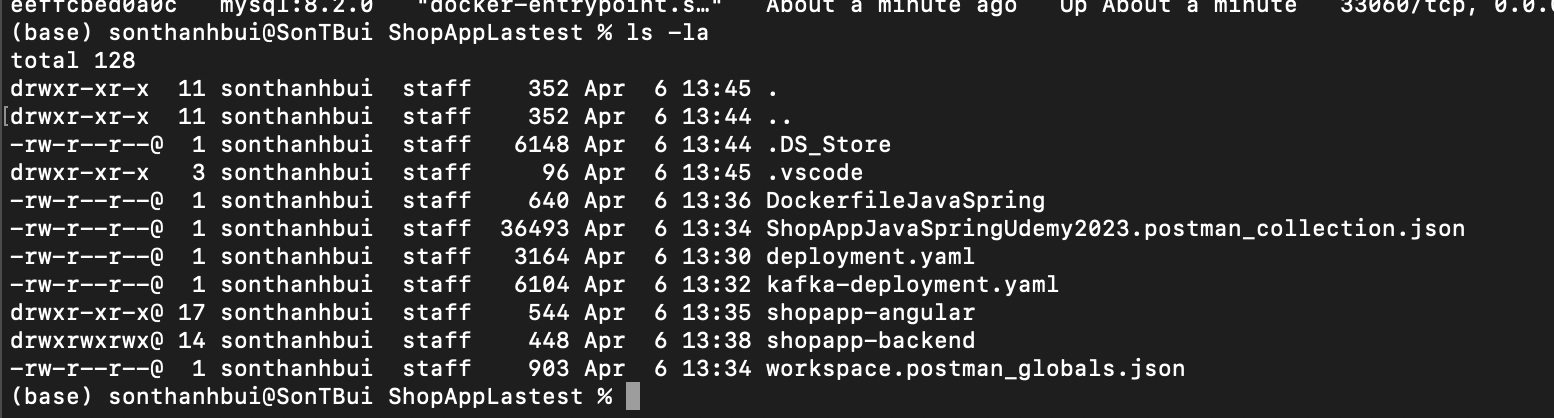


Below are step by step running my project.

Go to folder contain project

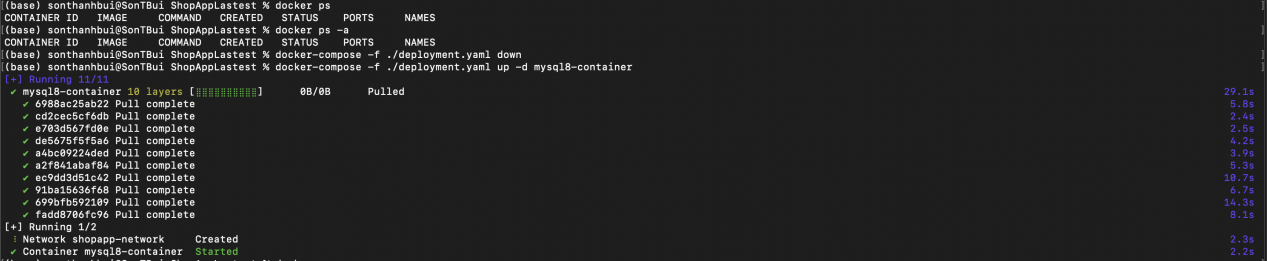
Check position: run:-> ls -la

If you have the same result like this, you are correct

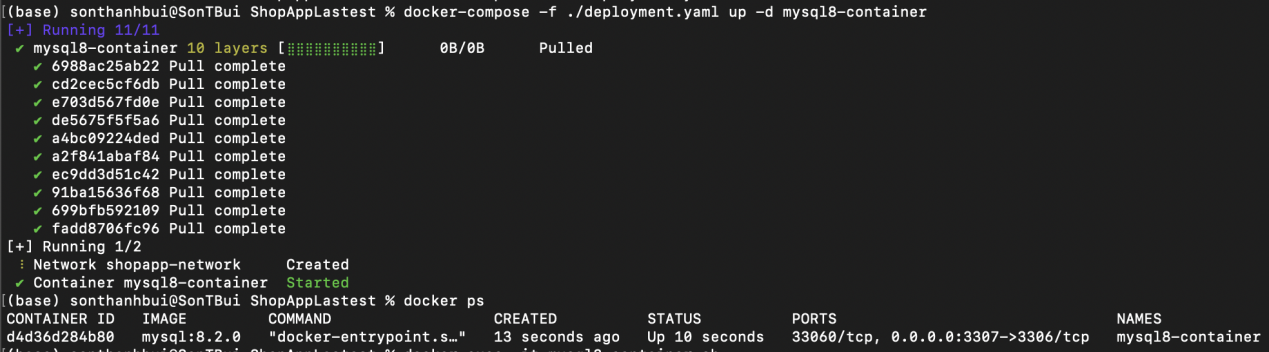


Run this statement: docker-compose -f ./deployment.yaml up -d mysql8-container

The result:

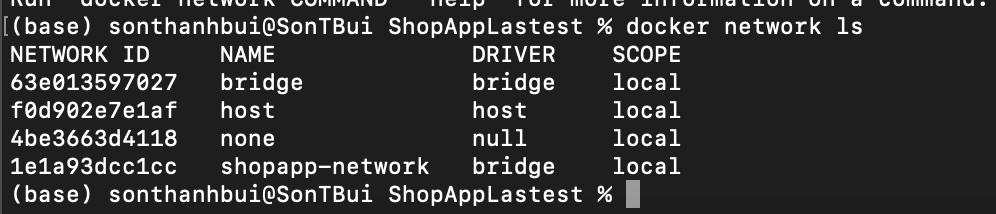


After that need check container was pulling in docker by: docker ps



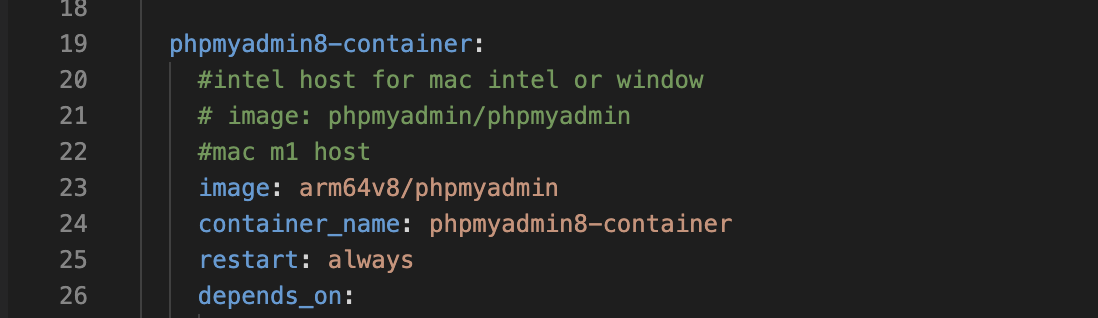
You need to check network after create container mysql8, statement: docker network ls.

The result like this:

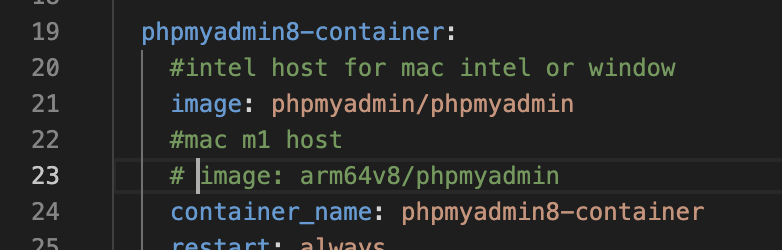


**\* Pulling container phpmyadmin.**

**Attention:** If you are using *window or mac intel,* you need to **NO** comment **line 21** in file deployment.yaml. And comment lines **23**. And if you are using mac m1 you need to **NO** comment **line 23**. And comments lines **21**.



**For Mac M1**

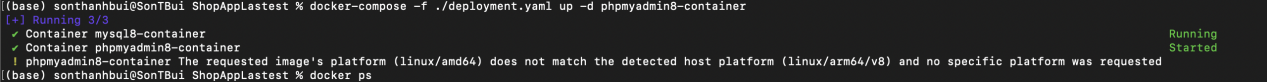


**For window or mac intel**

Next step is creating phpmyadmin8-container. **READ ATTENTION BEFORE RUN THIS STATEMENT.**

**Run this statement :**  docker-compose -f ./deployment.yaml up -d phpmyadmin8-container

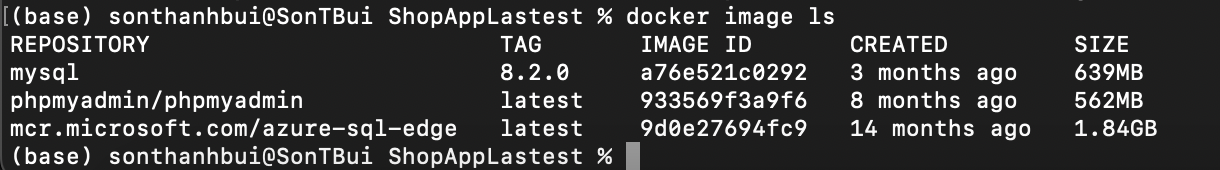
Result same as create mysql8-container. My docker have phpmyadmin before, so my result like this:



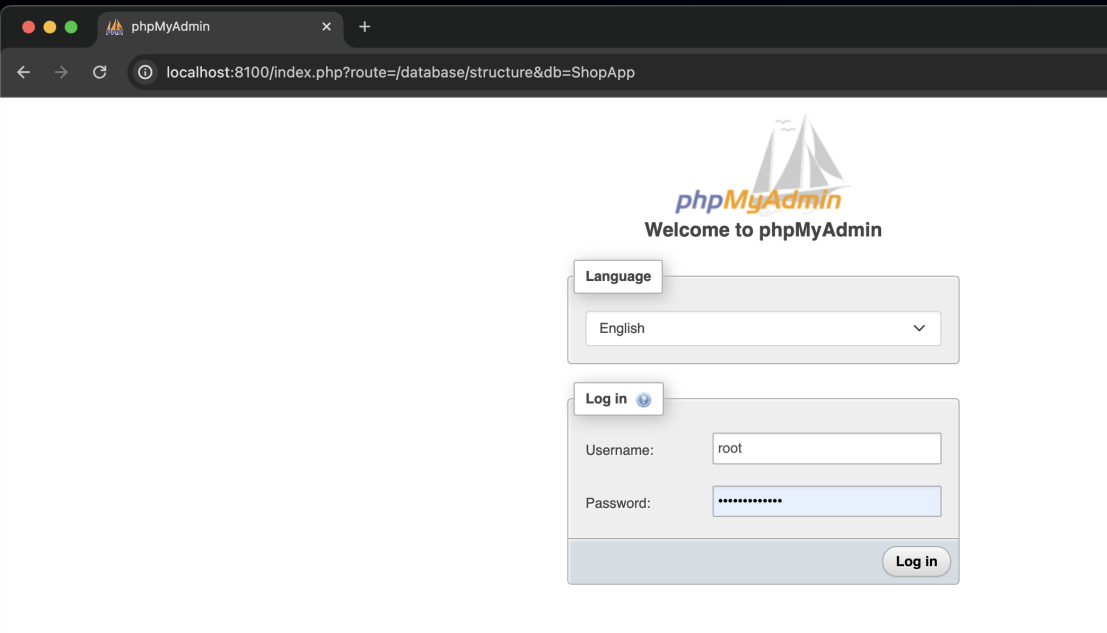
**Then**, check image in your laptop or pc. statement is: “ **docker image ls** “ or

“ **docker images** ”

result here:



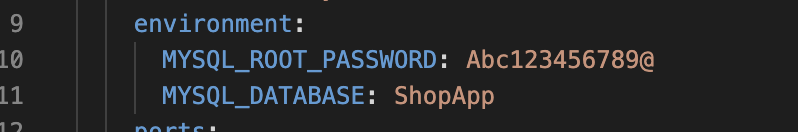
Next stage, open browser and find url: “ localhost:8100 ”(recommend using chrome for build project).



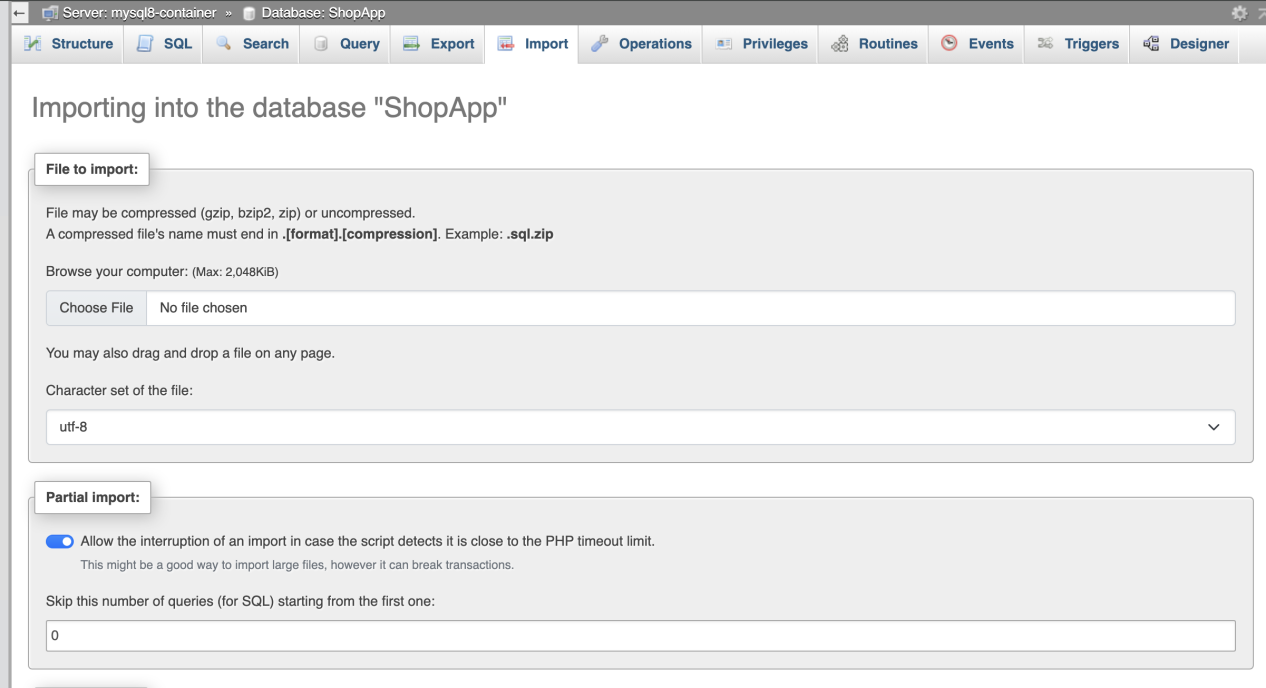
**username** is: root

**password** in deployment.yaml: Abc123456789@.

you can change the password before you create phpmyadmin and mysql8

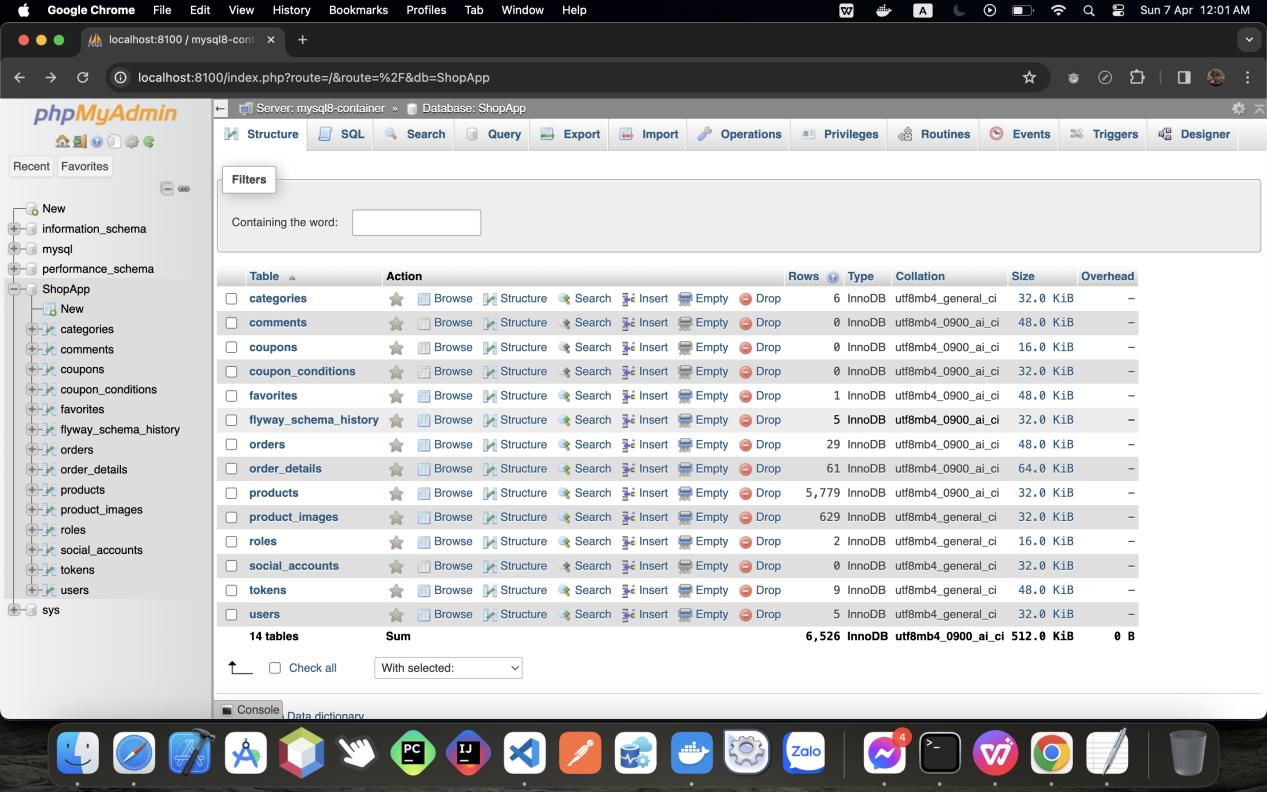


Next step you in import file database i was already build and put it on folder contain project.



<-- Chose file shopapp.sql

Rolling down and press import.



**You have successfully for project’s database.**

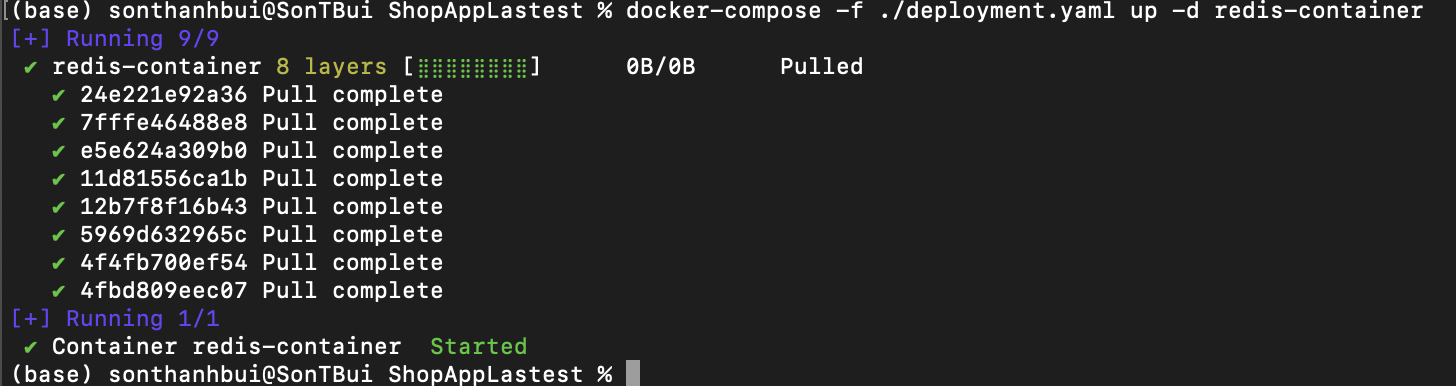
**Now we need connect backend to frontend. Follow instruction.**

***I have written statement below file deployment.yaml.***

**Run statements create redis**:

“docker-compose -f ./deployment.yaml up -d redis-container”

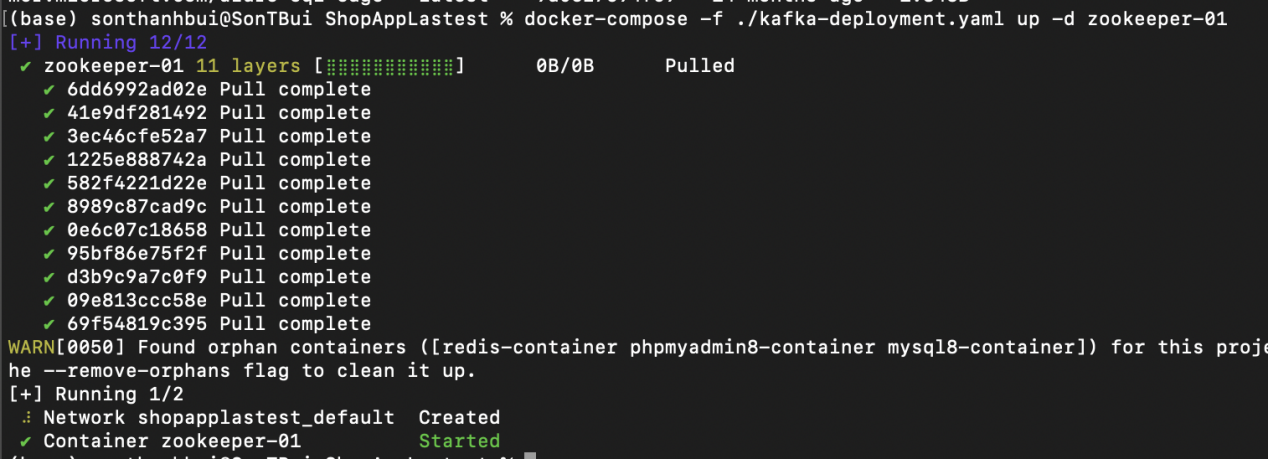
**Result:**

****

**Now go to file kafka-deployment.yaml to build zookeeper and kafka.**

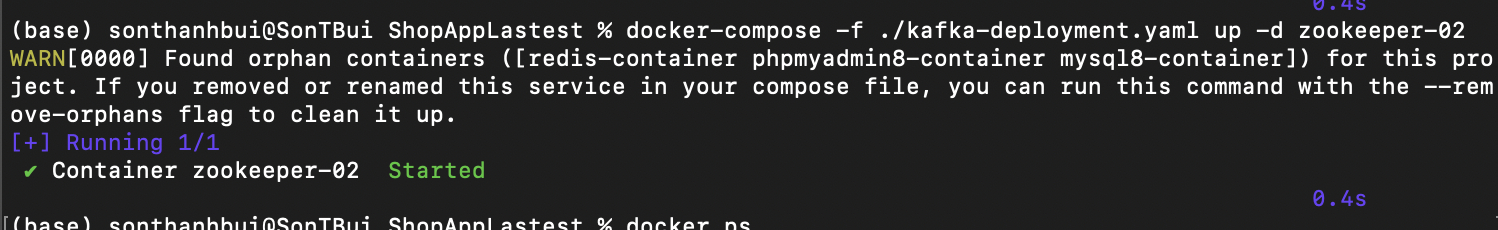
**Run statement: create zookeeper-01**

“ docker-compose -f ./kafka-deployment.yaml up -d zookeeper-01 ”



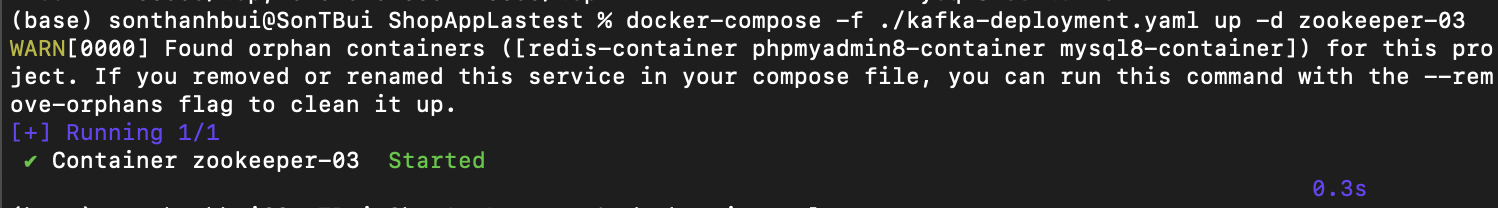
**Run statement: create zookeeper-02**

“ docker-compose -f ./kafka-deployment.yaml up -d zookeeper-02 ”

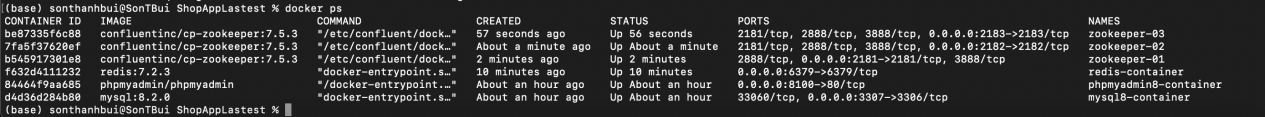


**Run statement: create zookeeper-03**

“ docker-compose -f ./kafka-deployment.yaml up -d zookeeper-03 ”



\*DON’T WORRY ABOUT WHY CREATE ZOOKEEPER 02 AND 03 SO QUICKLY. BECAUSE THEY REUSE IMAGES\*

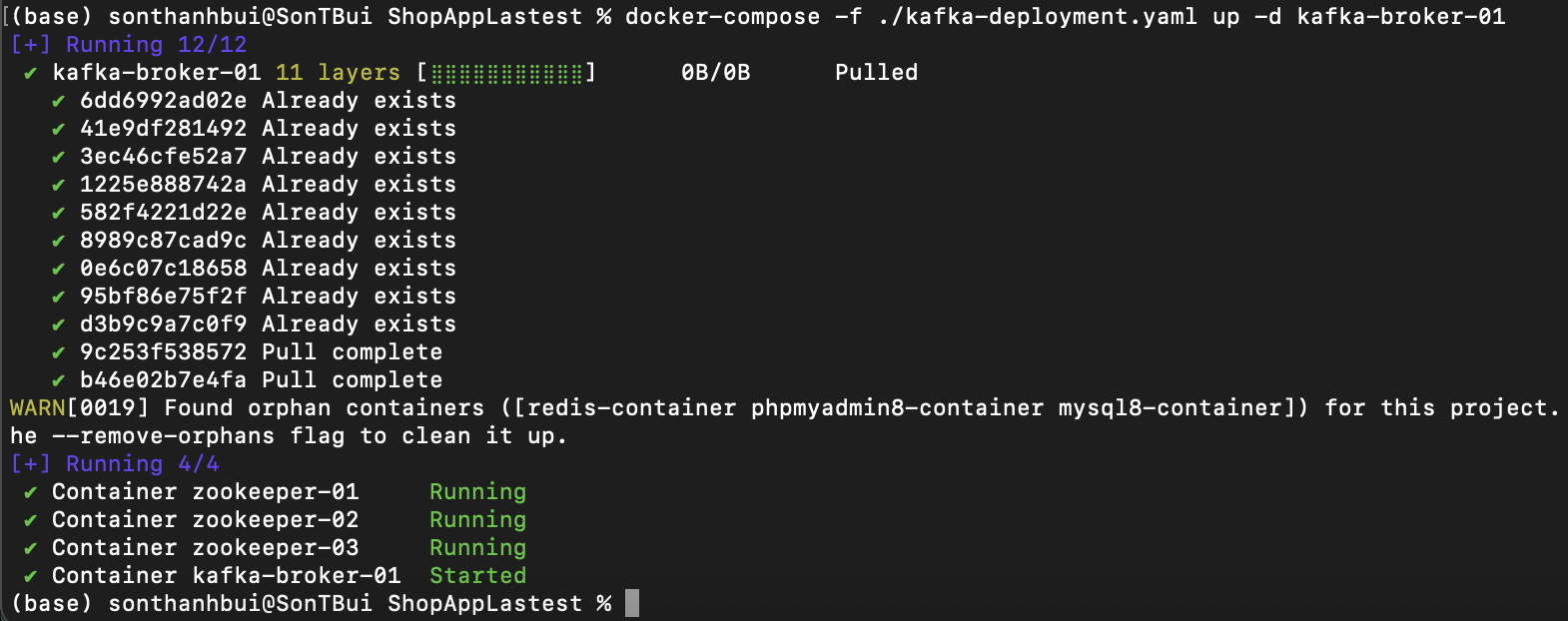
If you run docker ps and you have results same as me, you are correct.

**\*Create kafka-broker-01.**

**Run statement:**

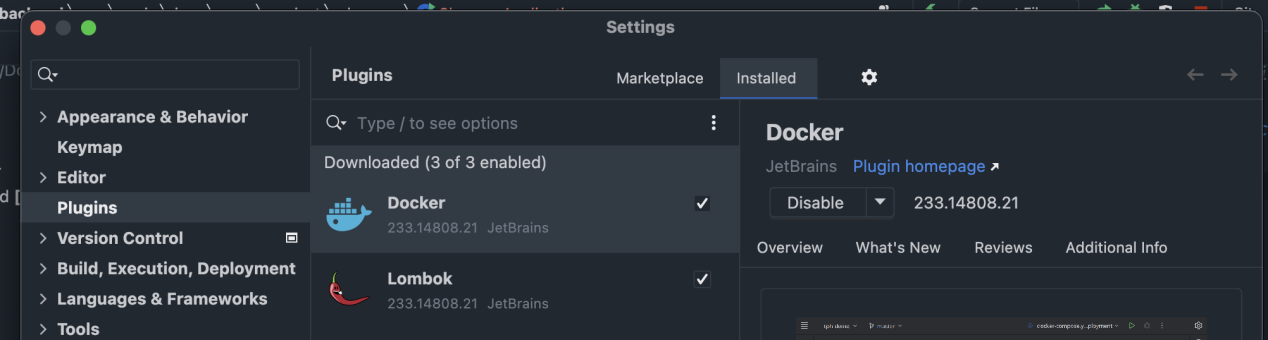
“ docker-compose -f ./kafka-deployment.yaml up -d kafka-broker-01 ”

The result:

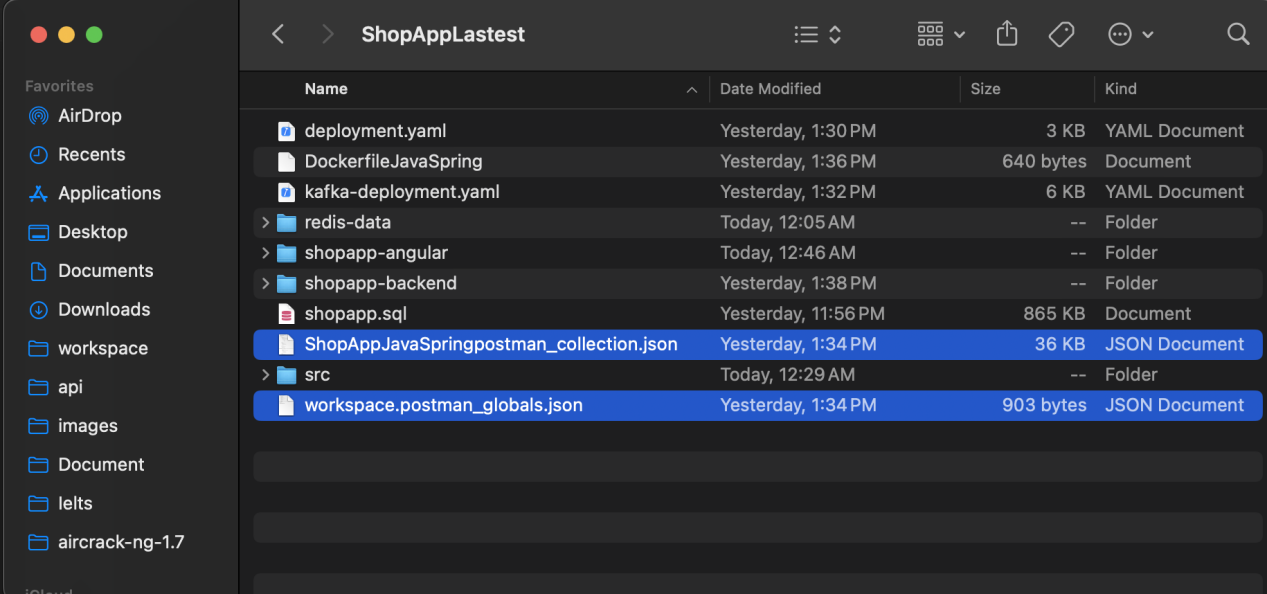


Next step, I prefer using IDE **INTELLIJ**. Open Folder “shopapp-backend”. And add plugin

**“LOMBOK”.** Then press apply - > ok.



**Open Postman and import two file into postman.**

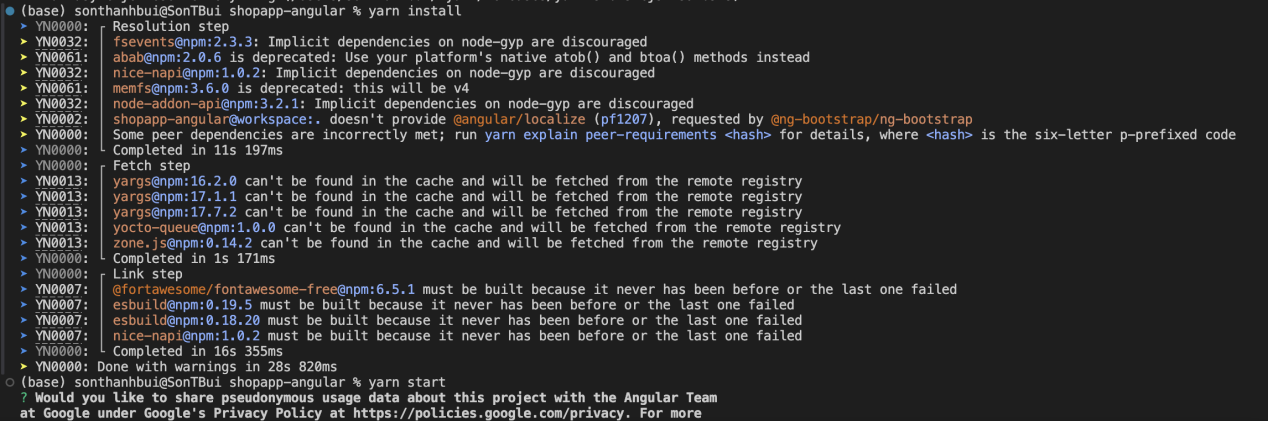
****



 If you have screen same as picture you are done.

Open VSCODE. Open folder contain project. Use terminal in vscode to go to deep folder

shopapp-angular. And run code “ **yarn install** ”



Press yes if you permission or no if you don’t want.

Then run “ **yarn start** ” to run this project.

