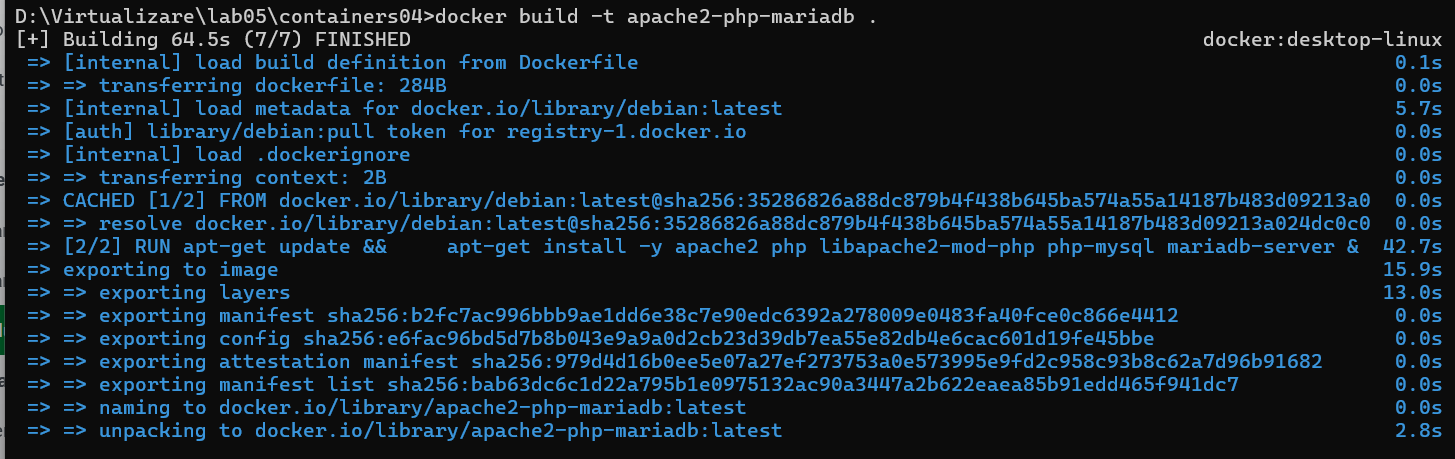
1.Crearea imaginii:

docker build -t apache2-php-mariadb .



2. Creați un container apache2-php-mariadb din imaginea apache2-php-mariadb și porniți-l în modul de fundal cu comanda bash.

docker run -d --name apache2-php-mariadb -p 8080:80 apache2-php-mariadb bash



Explicatie

 **docker run** – Creează și rulează un nou container.

 **-d** – Rulează containerul în fundal (detached mode).

 **--name my\_container** – Atribuie un nume (my\_container) containerului (poți alege orice alt nume).

 **-p 8080:80** – Mapează portul 80 al containerului la portul 8080 al mașinii gazdă (accesibil în browser la http://localhost:8080).

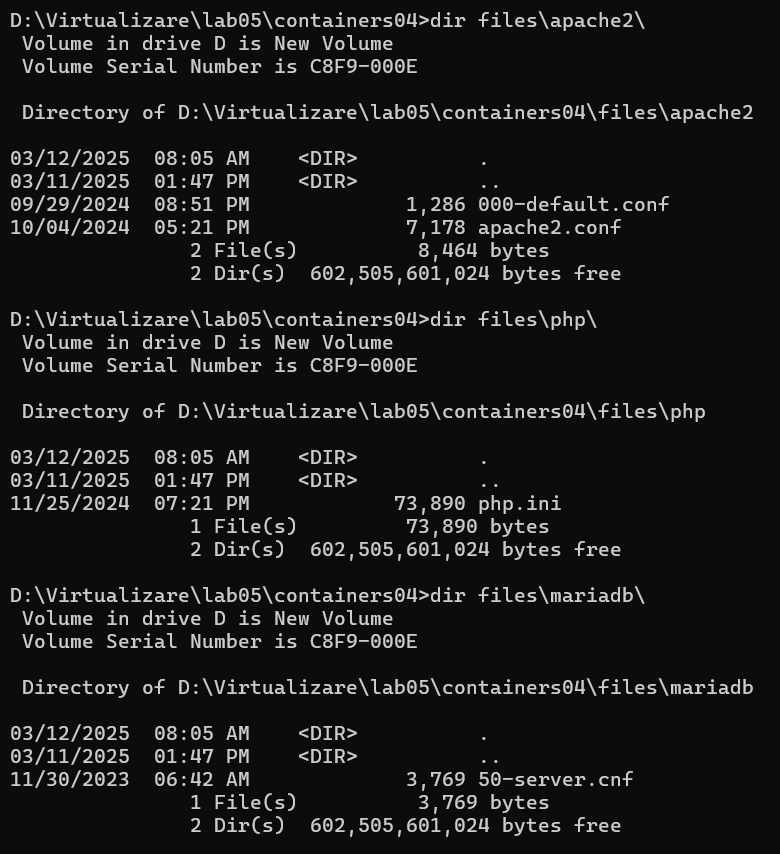
 **apache2-php-mariadb** – Numele imaginii Docker utilizate pentru container.

 **bash** – Specifică rularea unui shell Bash la pornirea containerului

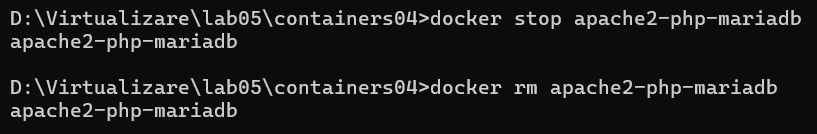
3. Copiați din container fișierele de configurare apache2, php, mariadb în directorul files/ de pe computer. Pentru a face acest lucru, în contextul proiectului, executați comenzile:



4. După executarea comenzilor, în directorul files/ ar trebui să apară fișierele de configurare apache2, php, mariadb. Verificați dacă acestea există.



Opriți și ștergeți containerul apache2-php-mariadb.



### 5. Configurarea

**Fișierul de configurare apache2**

Deschideți fișierul files/apache2/000-default.conf, găsiți linia #ServerName www.example.com și înlocuiți-o cu ServerName localhost.

Găsiți linia ServerAdmin webmaster@localhost și înlocuiți adresa de e-mail cu a dvs.

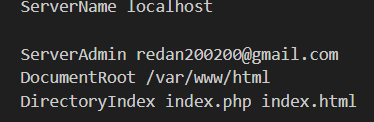
După linia DocumentRoot /var/www/html adăugați următoarea linie:

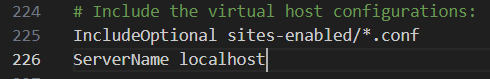
DirectoryIndex index.php index.html

Salvați fișierul și închideți-l.

La sfârșitul fișierului files/apache2/apache2.conf adăugați următoarea linie:

ServerName localhost





#### Fișierul de configurare php

Deschideți fișierul files/php/php.ini, găsiți linia ;error\_log = php\_errors.log și înlocuiți-o cu error\_log = /var/log/php\_errors.log.

Setați parametrii memory\_limit, upload\_max\_filesize, post\_max\_size și max\_execution\_time astfel:

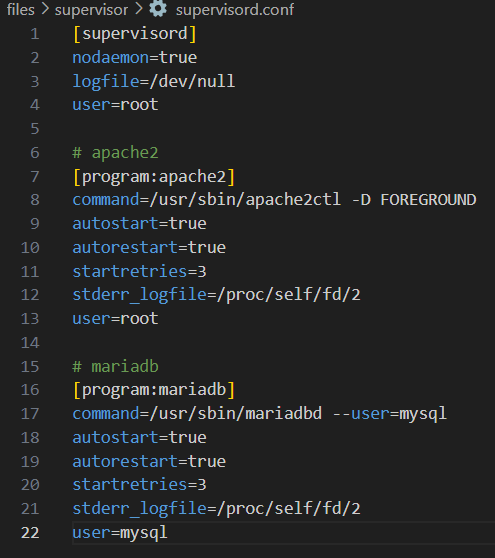


Deschideți fișierul files/mariadb/50-server.cnf, găsiți linia #log\_error = /var/log/mysql/error.log și eliminați # din fața ei.



### Crearea scriptului de pornire

In directorul files creati directorul supervisor si fisierul supervisord.conf cu urmatorul continut:



### Crearea Dockerfile

Deschiți fișierul Dockerfile și adăugați următoarele linii:

* după instrucția FROM ... adăugați montarea volumelor:

# mount volume for mysql data

VOLUME /var/lib/mysql

# mount volume for logs

VOLUME /var/log

* în instrucția RUN ... adăugați instalarea pachetului supervisor.
* după instrucția RUN ... adăugați copierea și dezarhivarea site-ului WordPress:

# add wordpress files to /var/www/html

ADD https://wordpress.org/latest.tar.gz /var/www/html/

* după copierea fișierelor WordPress adăugați copierea fișierelor de configurare apache2, php, mariadb, așa cum și a scriptului de pornire:

# copy the configuration file for apache2 from files/ directory

COPY files/apache2/000-default.conf /etc/apache2/sites-available/000-default.conf

COPY files/apache2/apache2.conf /etc/apache2/apache2.conf

# copy the configuration file for php from files/ directory

COPY files/php/php.ini /etc/php/8.2/apache2/php.ini

# copy the configuration file for mysql from files/ directory

COPY files/mariadb/50-server.cnf /etc/mysql/mariadb.conf.d/50-server.cnf

# copy the supervisor configuration file

COPY files/supervisor/supervisord.conf /etc/supervisor/supervisord.conf

* pentru functionarea mariadb creati directorul /var/run/mysqld si setati permisiile pe el:

# create mysql socket directory

RUN mkdir /var/run/mysqld && chown mysql:mysql /var/run/mysqld

* deschideti portul 80.
* adăugați comanda de pornire supervisord:

# start supervisor

CMD ["/usr/bin/supervisord", "-n", "-c", "/etc/supervisor/conf.d/supervisord.conf"]

Creati imaginea containerului cu numele apache2-php-mariadb și porniți containerul apache2-php-mariadb din imaginea apache2-php-mariadb. Verificați dacă site-ul WordPress este disponibil la adresa http://localhost/. Verificați dacă in directorul /var/www/html/ există fișierele site-ului WordPress. Verificați dacă fișierele de configurare apache2, php, mariadb sunt modificate.

