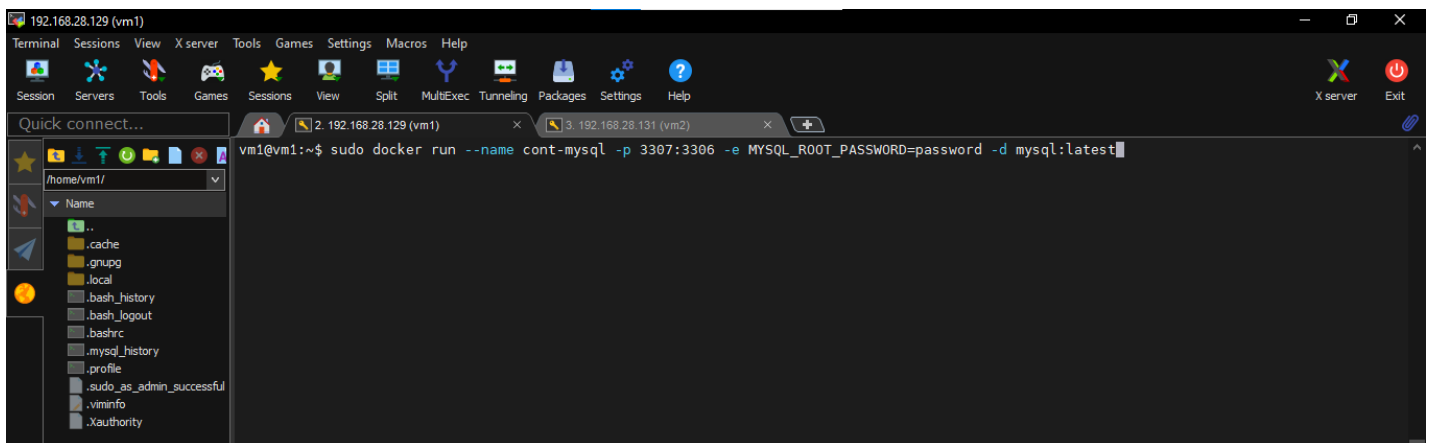


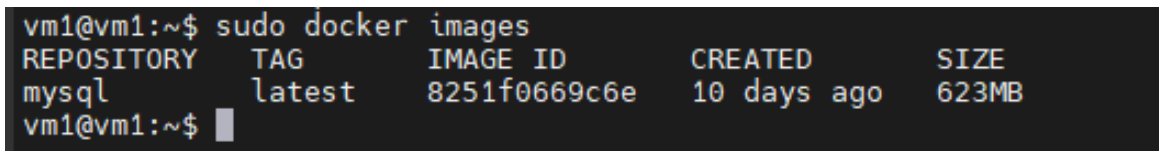
➤ Configuration machine virtuel « vm1 » pour le conteneur MySQL avec address IP :192.168.26.129

- 1) Creation d'un conteneur avec l'image MySQL avec configuration de mysql_root_password=password
 - On a utilisé le port cote user **3307**
 - Le port cote conteneur par défaut de MySQL **3306**



```
vm1@vm1:~$ sudo docker run --name cont-mysql -p 3307:3306 -e MYSQL_ROOT_PASSWORD=password -d mysql:latest
```

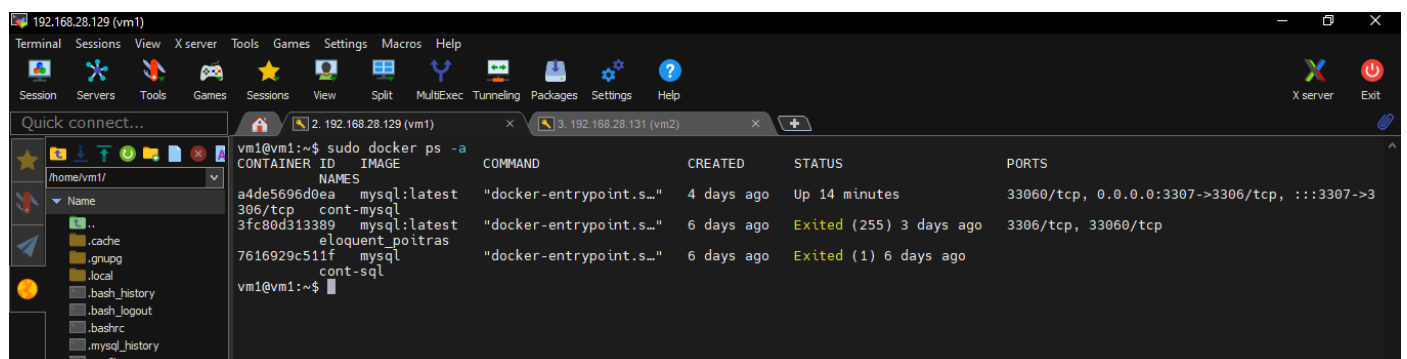
- 2) Affichage de l'image créer



```
vm1@vm1:~$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mysql	latest	8251f0669c6e	10 days ago	623MB

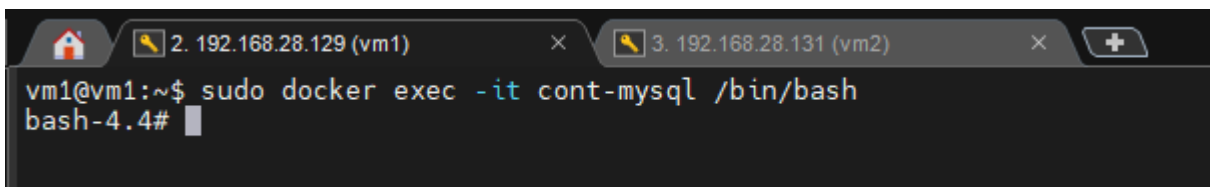
- 3) Afficher les detaillés du conteneur l'ID du conteneur respectif, l'image sous-jacente, la commande exécutée lorsque le conteneur a été démarré, l'heure à laquelle le conteneur respectif a été démarré et l'état actuel.



```
vm1@vm1:~$ sudo docker ps -a
```

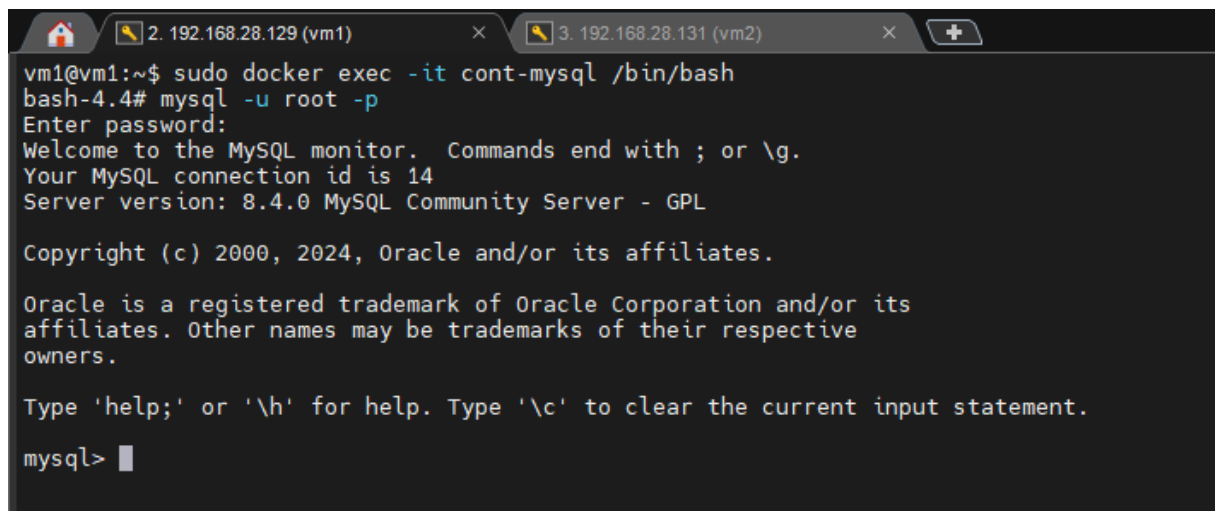
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
a4de5696d0ea	mysql:latest	"docker-entrypoint.s..."	4 days ago	Up 14 minutes	33060/tcp, 0.0.0.0:3307->3306/tcp, :::3307->3
306/tcp	cont-mysql	"docker-entrypoint.s..."	6 days ago	Exited (255) 3 days ago	3306/tcp, 33060/tcp
3fc80d313389	mysql:latest	"docker-entrypoint.s..."	6 days ago	Exited (255) 3 days ago	
eloquent_poitras	mysql	"docker-entrypoint.s..."	6 days ago	Exited (1) 6 days ago	
7616929c511f	mysql	"docker-entrypoint.s..."	6 days ago	Exited (1) 6 days ago	
cont-sql	cont-sql	"docker-entrypoint.s..."	6 days ago	Exited (1) 6 days ago	

- 4) Nom conteneur :cont-mysql
- -it : on veut un terminal et être interactif avec lui
 - /bin/bash : on lance « bash »
 - Bash : est un interpréteur de commandes (shell) pour les systèmes d'exploitation Linux



```
vm1@vm1:~$ sudo docker exec -it cont-mysql /bin/bash
bash-4.4#
```

- 5) Entrer dans mysql qui est installer dans un conteneur



```
vm1@vm1:~$ sudo docker exec -it cont-mysql /bin/bash
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.4.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, 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>
```

- 6) Voir la liste des base de données existant et les tous user pour le serveur MySQL :

```
2. 192.168.28.129 (vm1) x 3. 192.168.28.131 (vm2) x +
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.4.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, 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> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| projet |
| sys |
+-----+
5 rows in set (0.16 sec)

mysql> select User from mysql.user;
+-----+
| User |
+-----+
| mariam |
| root |
| mysql.infoschema |
| mysql.session |
| mysql.sys |
| root |
+-----+
6 rows in set (0.04 sec)

mysql> █
```

- 7) Création d'une base de données pour notre projet « ecommerce » et pour vérifier la création de la base de données avec succès :

```
mysql> create database ecommerce;
Query OK, 1 row affected (0.17 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| ecommerce |
| information_schema |
| mysql |
| performance_schema |
| projet |
| sys |
+-----+
6 rows in set (0.10 sec)

mysql> █
```

Création d'un utilisateur pour la base de données « ecommerce » qui a tous les droits d'accès et peut accéder de n'importe quelle adresse IP :

```

mysql> create user 'hassoun'@'%' identified by 'test';
Query OK, 0 rows affected (0.47 sec)

mysql> grant all privileges on ecommerce.* TO 'hassoun'@'%;
Query OK, 0 rows affected (0.07 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.02 sec)

mysql> select User from mysql.user;
+-----+
| User |
+-----+
| hassoun |
| mariam |
| root |
| mysql.infoschema |
| mysql.session |
| mysql.sys |
| root |
+-----+
7 rows in set (0.02 sec)

mysql> █

```

8) Creation des tables dans la base de donnes « ecommerce »

```

mysql> use ecommerce
Database changed
mysql> show tables;
Empty set (0.01 sec)

mysql> CREATE TABLE `categorie` (
  ->   `id` int(11) NOT NULL,
  ->   `libelle` varchar(100) NOT NULL,
  ->   `description` varchar(255) NOT NULL,
  ->   `date_creation` datetime NOT NULL DEFAULT current_timestamp(),
  ->   `icone` varchar(200) NOT NULL
  -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;█

```

9) Configuration de la base de donnes avec les tables nécessaires :

```
-> MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=19;
Query OK, 0 rows affected, 1 warning (0.28 sec)
Records: 0 Duplicates: 0 Warnings: 1

mysql> ALTER TABLE `produit`
-> MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=14;
Query OK, 0 rows affected, 1 warning (0.47 sec)
Records: 0 Duplicates: 0 Warnings: 1

mysql> ALTER TABLE `utilisateur`
-> MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;
Query OK, 0 rows affected, 1 warning (0.27 sec)
Records: 0 Duplicates: 0 Warnings: 1

mysql> ALTER TABLE `ligne_commande`
-> ADD CONSTRAINT `ligne_commande_ibfk_1` FOREIGN KEY (`id_produit`) REFERENCES `produit` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
-> ADD CONSTRAINT `ligne_commande_ibfk_2` FOREIGN KEY (`id_commande`) REFERENCES `commande` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;
Query OK, 0 rows affected (0.36 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_ecommerce |
+-----+
| categorie            |
| commande            |
| ligne_commande      |
| produit             |
| utilisateur         |
+-----+
5 rows in set (0.08 sec)

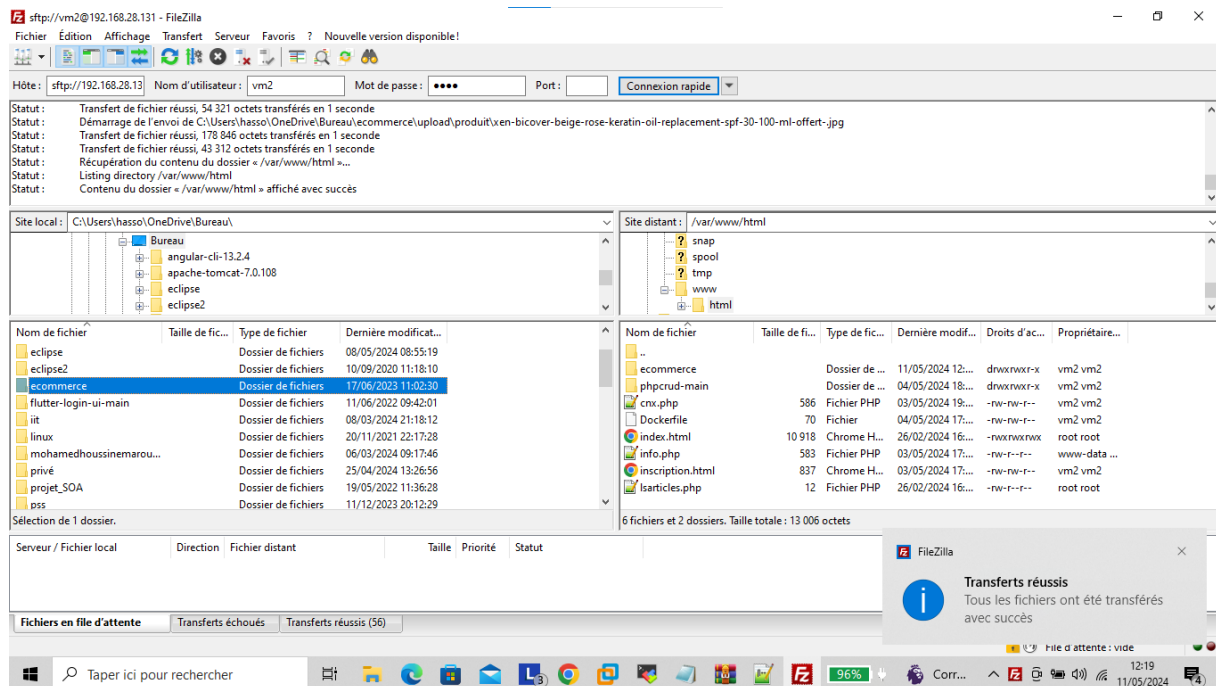
mysql>
```

➤ Configuration machine virtuel « vm2 » pour le conteneur apache avec adresse IP :192.168.26.131

1) Configuration de notre connexion avec la base de données située dans le conteneur dans le projet avec les données nécessaires :

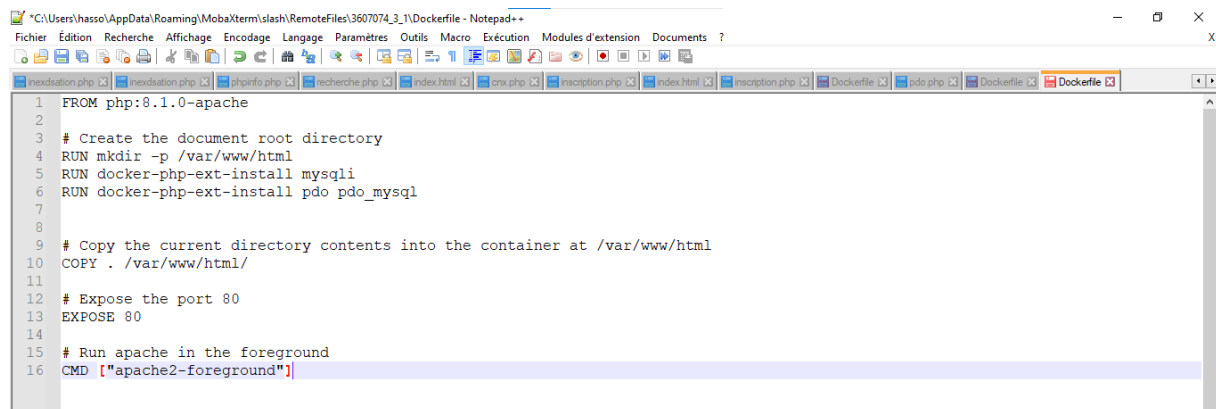
```
1 <?php
2
3
4 try{
5     $pdo = new PDO("mysql:host=192.168.28.129;dbname=ecommerce", 'hassoun', 'test');
6 }
7 catch(PDOException $e){
8     print "connection failed !: " . $e->getMessage() ;
9 }
10
11
12
13
14
```

2) On a utiliser filezilla pour transformer notre projet de notre machine physique vers la machine virtuel

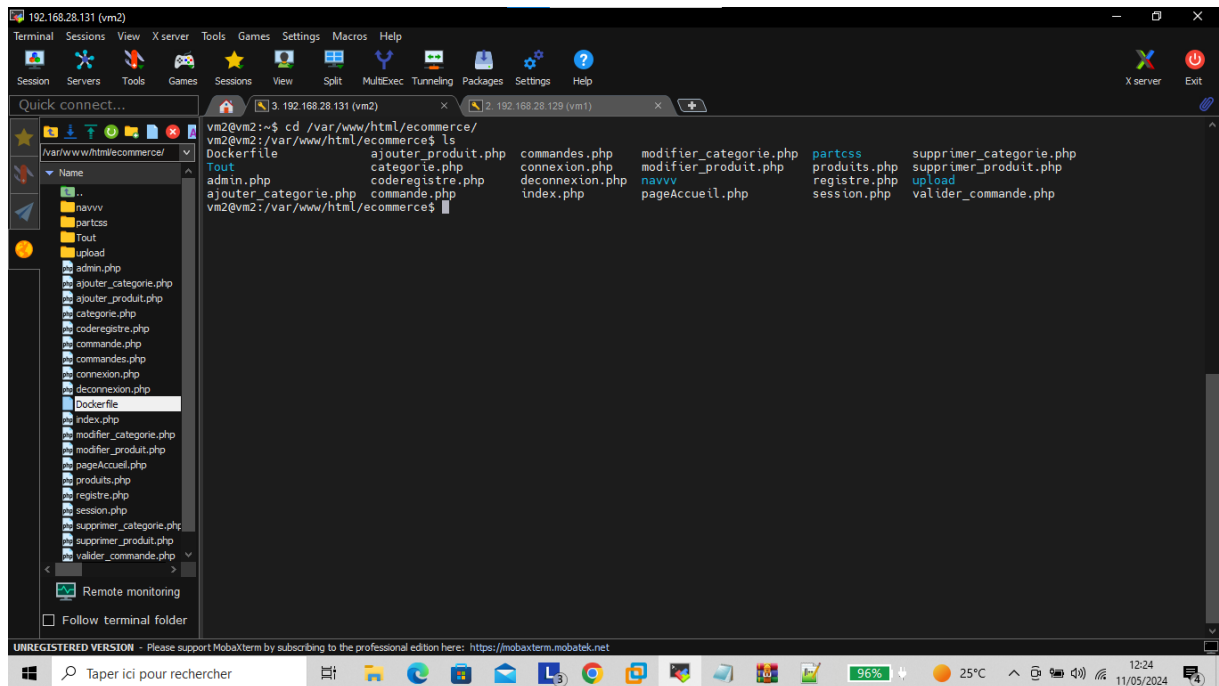


3) Dans notre projet on va faire la création de notre dockerfile :

- Dockerfile : est un fichier, «sans extension», qui contient une série d'instructions utilisées par Docker pour construire une image de conteneur.



4) Le chemin d'existence de notre dockerfile



5) Maintenant on va faire la creation de l'image nomme « imageapp » à partir du fichier Dockerfile (y compris le point à la fin)

- **(.)** a la fin indique que le dockerfile existe dans l'emplacement courant
- Image : un fichier qui contient toutes les instructions nécessaire pour exécuter une application (code source, bib, dépendances, variables d'environnement, etc).

The screenshot shows a MobaXterm terminal window with a file explorer on the left and a terminal output on the right. The file explorer shows a directory structure for a web application, including files like `admin.php`, `ajouter_categorie.php`, `ajouter_produit.php`, `categorie.php`, `coderegistre.php`, `commande.php`, `commandes.php`, `connexion.php`, `deconnexion.php`, `Dockerfile`, `index.php`, `modifier_categorie.php`, `modifier_produit.php`, `pageAccueil.php`, `produits.php`, `registre.php`, `session.php`, `supprimer_categorie.php`, `supprimer_produit.php`, and `valider_commande.php`. The terminal output shows the following steps:

```
during execution
- add LIBDIR to the 'LD_RUN_PATH' environment variable
during linking
- use the '-Wl,--rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
-----

Build complete.
Don't forget to run 'make test'.

+ strip --strip-all modules/pdo_mysql.so
Installing shared extensions: /usr/local/lib/php/extensions/no-debug-non-zts-20210902/
find . -name '*.gcno' -o -name '*.gcda' | xargs rm -f
find . -name '*.lo' -o -name '*.o' -o -name '*.dep' | xargs rm -f
find . -name '*.la' -o -name '*.a' | xargs rm -f
find . -name '*.so' | xargs rm -f
find . -name '*.libs' -a -type d | xargs rm -rf
rm -f libphp.la modules/* libs/*
rm -f ext/opcache/jit/zend_jit_x86.c
rm -f ext/opcache/jit/zend_jit_arm64.c
Removing intermediate container 82e12da1921b
--> c0cfb79dea48
Step 5/7 : COPY . /var/www/html/
--> d0afa8158232
Step 6/7 : EXPOSE 80
--> Running in 696f93fdffdcd
Removing intermediate container 696f93fdffdcd
--> 9841f2ee859f
Step 7/7 : CMD ["apache2-foreground"]
--> Running in 09a3cab6213a
Removing intermediate container 09a3cab6213a
--> d0ae1dc23f44
Successfully built d0ae1dc23f44
Successfully tagged imageapp:latest
vm2@vm2: /var/www/html/ecommerce$
```

6) Creation avec succes

The screenshot shows a MobaXterm terminal window with a file explorer on the left and a terminal output on the right. The file explorer shows a directory structure for a web application, including files like `admin.php`, `ajouter_categorie.php`, `ajouter_produit.php`, `categorie.php`, `coderegistre.php`, `commande.php`, `commandes.php`, `connexion.php`, `deconnexion.php`, `Dockerfile`, `index.php`, `modifier_categorie.php`, `modifier_produit.php`, `pageAccueil.php`, `produits.php`, `registre.php`, `session.php`, `supprimer_categorie.php`, `supprimer_produit.php`, and `valider_commande.php`. The terminal output shows the following steps:

```
during execution
- add LIBDIR to the 'LD_RUN_PATH' environment variable
during linking
- use the '-Wl,--rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to '/etc/ld.so.conf'

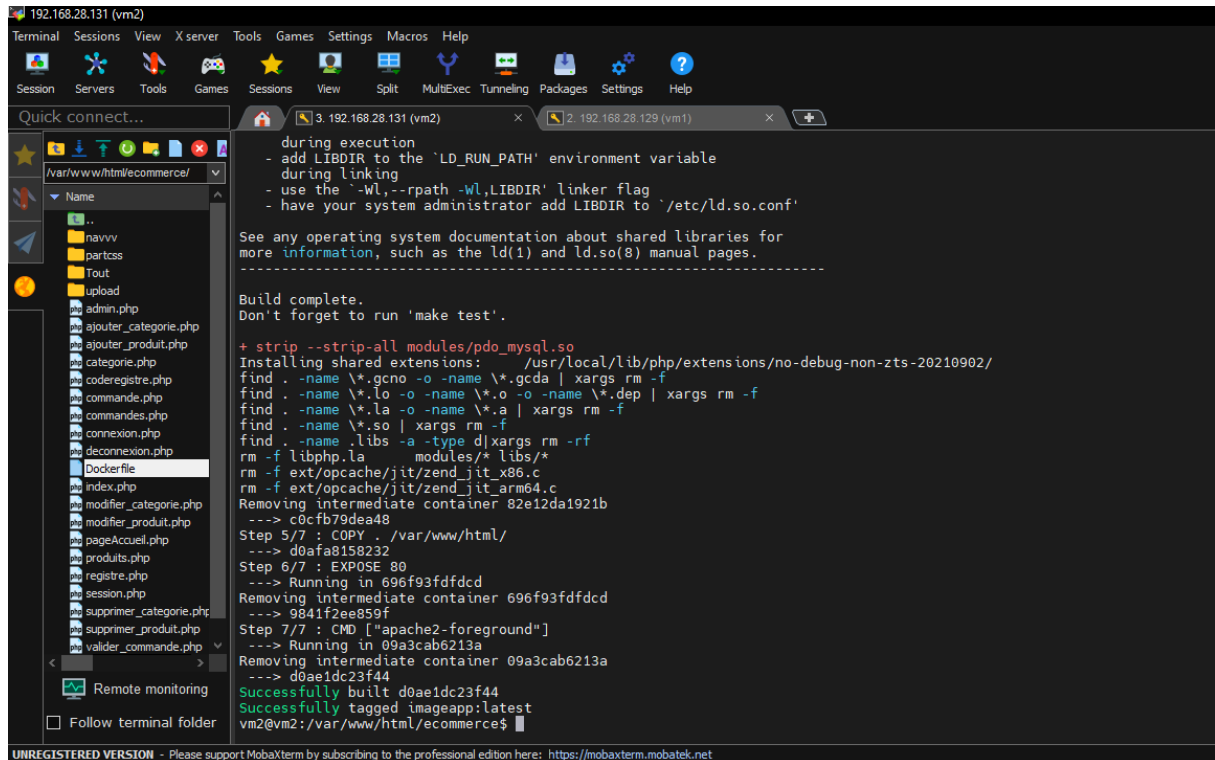
See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
-----

Build complete.
Don't forget to run 'make test'.

+ strip --strip-all modules/pdo_mysql.so
Installing shared extensions: /usr/local/lib/php/extensions/no-debug-non-zts-20210902/
find . -name '*.gcno' -o -name '*.gcda' | xargs rm -f
find . -name '*.lo' -o -name '*.o' -o -name '*.dep' | xargs rm -f
find . -name '*.la' -o -name '*.a' | xargs rm -f
find . -name '*.so' | xargs rm -f
find . -name '*.libs' -a -type d | xargs rm -rf
rm -f libphp.la modules/* libs/*
rm -f ext/opcache/jit/zend_jit_x86.c
rm -f ext/opcache/jit/zend_jit_arm64.c
Removing intermediate container 82e12da1921b
--> c0cfb79dea48
Step 5/7 : COPY . /var/www/html/
--> d0afa8158232
Step 6/7 : EXPOSE 80
--> Running in 696f93fdffdcd
Removing intermediate container 696f93fdffdcd
--> 9841f2ee859f
Step 7/7 : CMD ["apache2-foreground"]
--> Running in 09a3cab6213a
Removing intermediate container 09a3cab6213a
--> d0ae1dc23f44
Successfully built d0ae1dc23f44
Successfully tagged imageapp:latest
vm2@vm2: /var/www/html/ecommerce$
```


7) Ona créer le conteneur a partir de l'image avec le port

- 8090 cote user
- 80 cote conteneur par default



The screenshot shows a MobaXterm terminal window with a file explorer on the left and a terminal output on the right. The file explorer shows a directory structure for a web application, including files like `admin.php`, `ajouter_categorie.php`, `ajouter_produit.php`, `categorie.php`, `coderegistre.php`, `commande.php`, `commandes.php`, `connexion.php`, `deconnexion.php`, `Dockerfile`, `index.php`, `modifier_categorie.php`, `modifier_produit.php`, `pageAccueil.php`, `produits.php`, `registre.php`, `session.php`, `supprimer_categorie.php`, `supprimer_produit.php`, and `valider_commande.php`. The terminal output shows the following commands and their results:

```
during execution
- add LIBDIR to the 'LD_RUN_PATH' environment variable
during linking
- use the '-Wl,-rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
-----

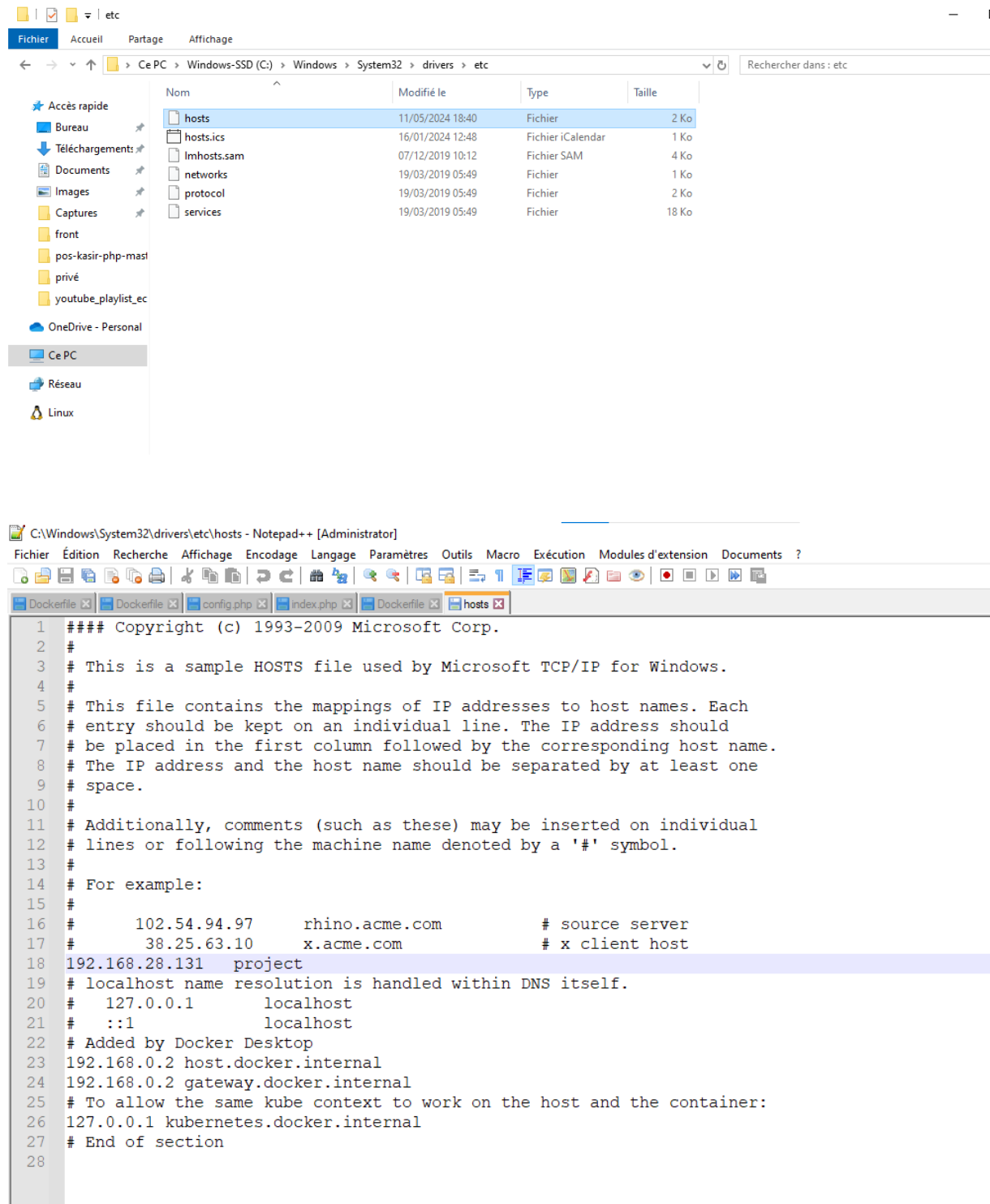
Build complete.
Don't forget to run 'make test'.

+ strip --strip-all modules/pdo_mysql.so
Installing shared extensions: /usr/local/lib/php/extensions/no-debug-non-zts-20210902/
find . -name \*.gcno -o -name \*.gcda | xargs rm -f
find . -name \*.lo -o -name \*.o -o -name \*.dep | xargs rm -f
find . -name \*.la -o -name \*.a | xargs rm -f
find . -name \*.so | xargs rm -f
find . -name \*.libs -a -type d | xargs rm -rf
rm -f libphp.la modules/* libs/*
rm -f ext/opcache/jit/zend_jit_x86.c
rm -f ext/opcache/jit/zend_jit_arm64.c
Removing intermediate container 82e12da1921b
--> c0c6fb79dea48
Step 5/7 : COPY . /var/www/html/
--> d0afa8158232
Step 6/7 : EXPOSE 80
--> Running in 696f93fdcdcd
Removing intermediate container 696f93fdcdcd
--> 9841f2ee859f
Step 7/7 : CMD ["apache2-foreground"]
--> Running in 09a3cab6213a
Removing intermediate container 09a3cab6213a
--> d0ae1dc23f44
Successfully built d0ae1dc23f44
Successfully tagged imageapp:latest
vm2@vm2:/var/www/html/ecommerce$
```

At the bottom of the terminal window, there is a message: "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>".

8) Maintenant on va tester notre application

9) Pour modifier l'adresse IP par nom de domaine on va



The screenshot shows a Windows File Explorer window with the path `C:\Windows\System32\drivers\etc`. The file `hosts` is selected. Below it, a Notepad++ window is open, displaying the contents of the `hosts` file. The file contains a list of IP addresses and hostnames, including `192.168.28.131 project`, `192.168.0.2 host.docker.internal`, and `192.168.0.2 gateway.docker.internal`.

Nom	Modifié le	Type	Taille
hosts	11/05/2024 18:40	Fichier	2 Ko
hosts.ics	16/01/2024 12:48	Fichier iCalendar	1 Ko
lmhosts.sam	07/12/2019 10:12	Fichier SAM	4 Ko
networks	19/03/2019 05:49	Fichier	1 Ko
protocol	19/03/2019 05:49	Fichier	2 Ko
services	19/03/2019 05:49	Fichier	18 Ko

```
1 ##### Copyright (c) 1993-2009 Microsoft Corp.
2 #
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4 #
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #         102.54.94.97       rhino.acme.com          # source server
17 #         38.25.63.10       x.acme.com              # x client host
18 192.168.28.131 project
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1       localhost
21 #   ::1            localhost
22 # Added by Docker Desktop
23 192.168.0.2 host.docker.internal
24 192.168.0.2 gateway.docker.internal
25 # To allow the same kube context to work on the host and the container:
26 127.0.0.1 kubernetes.docker.internal
27 # End of section
28
```