Installation de Docker et mise en place d'un enthronement LAMP (Linux, Apache, MySQL, PHP)

Docker est une plateforme qui permet de créer, déployer et exécuter des applications dans des conteneurs légers et isolés. Ces conteneurs embarquent tout le nécessaire pour faire fonctionner une application (code, bibliothèques, configurations), ce qui garantit un environnement cohérent, quelle que soit la machine. Utiliser Docker pour mettre en place un environnement **LAMP** (Linux, Apache, MySQL, PHP) est très utile, car cela simplifie l'installation, évite les conflits de versions, facilite le déploiement sur d'autres systèmes, et permet de reproduire rapidement un environnement de développement ou de production. En résumé, Docker permet de gérer plus efficacement des projets web complexes tout en assurant portabilité et cohérence.

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

```
ant@127:~$ sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
compose-plugin
```

```
ant@127:~$  sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker-ce-cli is already the newest version (5:28.2.2-1~debian.12~bookworm).
containerd.io is already the newest version (1.7.27-1).
docker-buildx-plugin is already the newest version (0.24.0-1~debian.12~bookworm).
The following additional packages will be installed:
  docker-ce-rootless-extras libslirp0 pigz slirp4netns
Suggested packages:
  cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  docker-ce docker-ce-rootless-extras docker-compose-plugin libslirp0 pigz slirp4netns
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 39.9 MB/40.1 MB of archives.
After this operation, 180 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
sudo docker run hello-world
```

Terminal

```
ant@127:~$  sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:0b6a027b5cf322f09f6706c754e086a232ec1ddba835c8a15c6cb74ef0d43c29
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Docker Compose est un outil qui sert à **définir, configurer et exécuter plusieurs conteneurs Docker à la fois** à partir d'un simple fichier texte appelé `docker-compose.yml`.

Docker compose

Dans beaucoup de projets, une application ne se compose pas d'un seul conteneur. Par exemple, un environnement **LAMP** complet nécessite :

- un conteneur pour **Apache + PHP**,
- un conteneur pour **MySQL**,
- éventuellement un conteneur pour **phpMyAdmin**,
- et parfois d'autres services (Redis, mail, etc.).

Gérer tous ces conteneurs à la main avec `docker run` devient vite compliqué.

C'est là que **Docker Compose** devient très utile :

Il permet de **définir tous les services nécessaires dans un seul fichier**.

Il simplifie la gestion avec des **commandes courtes** : `docker-compose up` pour tout lancer, `docker-compose down` pour tout arrêter.

Il permet de **lier facilement les services entre eux** (ex. : PHP communique avec MySQL via le nom du service).

Il facilite la **reproductibilité** de l'environnement sur d'autres machines ou serveurs.

```
ant@127:~$ docker compose version
Docker Compose version v2.36.2
```

```
mkdir lamp-docker && cd lamp-docker
sudo nano docker-compose.yml
```

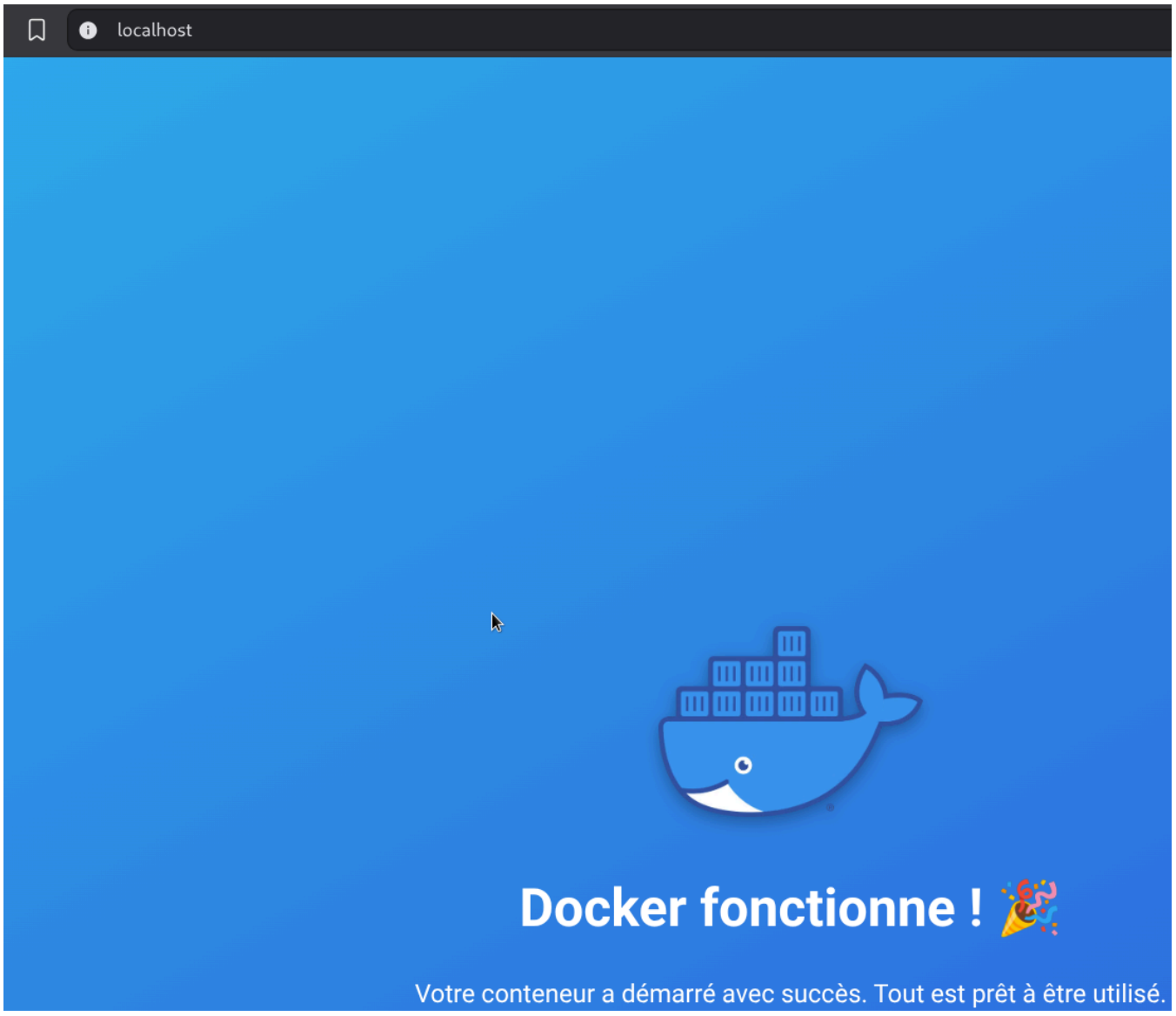
```
version: '3.8'    #check if there is a new docker compose version
services:
  web:
    image: php:8.2-apache    #check the php version you need for your project
    ports:
      - "80:80"    #this line maps your pc port to the container port
    depends_on:
      - db    #this line links this container to the db container
    volumes:
      - ./html:/var/www/html    #this line maps the content of ./html in your pc to the
/var/www/html of the container
  db:
    image: mysql:8.1.0    #check the mysql version you need for your project
    environment:
      MYSQL_ROOT_PASSWORD: root_password    #you can change the mysql root password here
      MYSQL_DATABASE: lamp_db    #you can change the database name here
    volumes:
      - ./mysql_data:/var/lib/mysql    #this line maps the content of ./mysql_data in your
pc to the /var/lib/mysql of the container
```

```
  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    ports:
      - "8080:80"    #this line maps your pc port to the container port
    depends_on:
      - db    #this line links this container to the db container
    environment:
      PMA_HOST: db
```

```
mkdir html
```

```
sudo usermod -aG docker $USER
newgrp docker
```

```
ant@127:~/oralsio/docker/lamp-docker$ sudo usermod -aG docker $USER
newgrp docker
ant@127:~/oralsio/docker/lamp-docker$
```




```
sudo tar -czvf lamp-docker.tar.gz lamp-docker
```

```
lamp-docker/mysql_data/#innodb_redo/#ib_redo16_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo39_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo15_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo24_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo30_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo29_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo37_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo33_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo13_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo20_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo27_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo22_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo9
lamp-docker/mysql_data/#innodb_redo/#ib_redo12_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo31_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo21_tmp
lamp-docker/mysql_data/#innodb_redo/#ib_redo10
lamp-docker/mysql_data/#innodb_redo/#ib_redo28_tmp
lamp-docker/mysql_data/client-key.pem
lamp-docker/mysql_data/ca-key.pem
lamp-docker/mysql_data/undo_002
lamp-docker/mysql_data/mysql.sock
lamp-docker/mysql_data/mysql/
lamp-docker/mysql_data/mysql/slow_log.CSV
lamp-docker/mysql_data/mysql/general_log.CSM
lamp-docker/mysql_data/mysql/slow_log_216.sdi
lamp-docker/mysql_data/mysql/general_log_215.sdi
lamp-docker/mysql_data/mysql/general_log.CSV
lamp-docker/mysql_data/mysql/slow_log.CSM
lamp-docker/readme.txt
ant@127:~/oralsio/docker$ ls
lamp-docker  lamp-docker.tar.gz
```