

ACTIVITES DOCKER

Activité 1 :

1. Créer un conteneur ubuntu 18.04 avec apache, mysql, php et phpmyadmin en utilisant l'image : **mattrayner/lamp**

On veuillera à mapper les ports 80 et 3306 et créer un volume app.

Pour cela on saisit: `docker run -p 80:80 -p 3306:3306 -v ${PWD}/app:/app mattrayner/lamp:latest-1804`

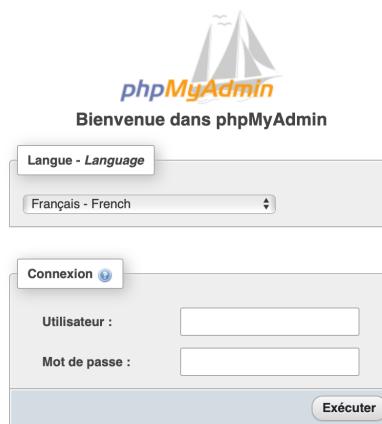
2. Lancer votre navigateur préféré en localhost : <http://localhost>

Index of /

Name	Last modified	Size	Description
<hr/>			
<i>Apache/2.4.29 (Ubuntu) Server at localhost Port 80</i>			

La page précédente s'affiche, cela permet de vérifier que le serveur Apache fonctionne. Nous n'avons pas encore de page web sur le serveur.

3. Saisissez l'url : <http://localhost/phpmyadmin/> dans votre navigateur.



Bienvenue dans phpMyAdmin

Langue - Language

Français - French

Connexion

Utilisateur :

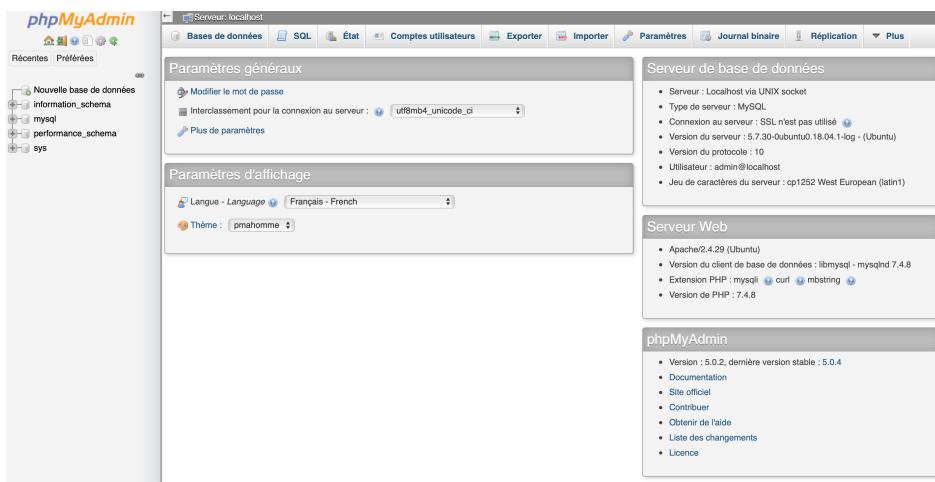
Mot de passe :

Exécuter

On constate que phpmyadmin est correctement installé.

4. Saisissez Utilisateur : admin et mot de passe : transmis lors de la saisie du docker run.

On obtient la page suivante :



The screenshot shows the phpMyAdmin configuration interface for a MySQL server. The top navigation bar includes 'Bases de données', 'SQL', 'État', 'Comptes utilisateurs', 'Exporter', 'Importer', 'Paramètres', 'Journal binaire', 'Réplication', and 'Plus'. The left sidebar lists databases: 'Nouvelle base de données', 'information_schema', 'mysql', 'performance_schema', and 'sys'. The main content area is divided into several sections:

- Paramètres généraux:** Includes 'Modifier le mot de passe' and 'Interclassement pour la connexion au serveur: utf8mb4_unicode_ci'.
- Paramètres d'affichage:** Shows 'Langue - Language: Français - French' and 'Thème: pmahomme'.
- Serveur de base de données:** Lists the server as 'localhost via UNIX socket', MySQL type, SSL connection status, server version (5.7.30-0ubuntu0.18.04.1-log), protocol version (10), user (admin@localhost), and character set (cp1252 West European (latin1)).
- Serveur Web:** Lists Apache 2.4.29 (Ubuntu), PHP version 7.4.8, and extensions mysqli, curl, mbstring.
- phpMyAdmin:** Lists version 5.0.2, documentation, official site, contributors, help, change logs, and license.

5. A l'aide de visual studio code, créer un fichier nommé test.php et enregistrez le dans votre répertoire \${PWD}/app. En effet, nous avons créé précédemment un volume app que l'on a lié avec le répertoire /app du conteneur en saisissant : `v ${PWD}/app:/app`

Contenu du fichier test.php :

```
<?php
Echo « page internet de Nom Prénom dans un conteneur »
?>
```

6. Tester ensuite le site internet en saisissant : <http://localhost/test.php>
7. Relever l'adresse ip de votre pc et faites une requête vers votre site depuis le PC de votre voisin. Que constatez-vous ? Cette IP est-elle celle de votre conteneur ? expliquer.

8. Dans une autre fenêtre terminale créer à nouveau un conteneur avec la commande :

```
docker run -p 80:80 -p 3306:3306 -v ${PWD}/app2:/app
mattrayner/lamp:latest-1804
```

9. Que constatez-vous ?

10. Modifier la commande précédente comme ci-dessous :

```
docker run -p 81:80 -p 3307:3306 -v ${PWD}/app2:/app
mattrayner/lamp:latest-1804
```

11. Tester la connexion depuis le PC du voisin. Que constatez-vous ?

SCENARIO DU TP

The diagram illustrates the difference in launching a server between a traditional VM and a Docker container. On the left, a 'Machine Virtuelle (VM)' is shown as a stack of components: 'App LAMP', 'OS Complet (Lourd!)', 'Librairies', and 'Matériel Virtuel'. Below it, the text 'Démarrage... 2 min' (Startup... 2 min) is shown. On the right, a 'Conteneur Docker' is shown as a stack of components: 'App LAMP', 'Dépendances minimales', and 'Image'. Below it, a speech bubble says 'Salut Sam ! La VM, c'est comme démarrer un ordinateur entier. Moi, j'emmène le cœur de ta machine pour que tu fasses juste le processus nécessaire. C'est beaucoup plus léger !' (Hello Sam ! A VM is like starting a whole computer. I just bring the heart of your machine so you can do the necessary process. It's much lighter!). A laptop icon shows a blue arrow pointing to it with the text 'Démarrage : 2 secondes !' (Startup: 2 seconds!). A smiling person is shown on the right, indicating the faster startup time of Docker.

Objectif du TP



Ok Whale ! Aujourd’hui, le prof veut qu’on lance une pile LAMP complète pour développer une page web.

Et il faut que je puisse voir la page depuis mon PC !

Facile. On va utiliser une image pré-téléchargée pour l’emploi, ouvrir des ports et créer un volume pour tes fichiers.

Ce qu’on apprend

Objectif du TP : déployer un serveur web, une base de données, et rendre le tout accessible sur le réseau.

Planché 2 : Le Volume (Pont entre l'hôte et le conteneur)

Planche 2

Le volume, c'est le pont

C'est ça la magie du volume ! Tu utilises tes outils préférés sur ton PC (VS Code), et le conteneur "voit" le fichier instantanément.

Hôte

Conteneur (comme dans <IMAGE 0>)

Tester test.php

Page internet de Sam Ci.EL dans un conteneur

Ce qu'on apprend

On crée un fichier de code standard dans le dossier partagé sur la machine hôte.

Yesss ! Mon code s'exécute.

Yesss ! Mon code s'exécute.

Apache dans le conteneur a lu ton fichier via le volume et PHP l'a interprété.

Ce qu'on apprend

Le développement est immédiat : modification locale > rafraîchissement du navigateur > résultat visible.

Planché 4 : Le conflit de ports

Planche 4

On relance un 2e conteneur (la tentative)



Conteneur 1 (comme dans <IMAGE 0>)

Conteneur 2

Hôte

Port 80

CRASH!

Je veux un deuxième site pour un autre projet. Je relance la même commande !

```
> docker run -p 80:80 -p 3306:3306 -v $PWD:/app2:z app mattnayr/lamp:latest
1804
```

Ah... ça bloque ! (Le constat)



Mince, ça ne marche pas.

Erreur : Le port 80 est déjà utilisé sur cette machine.

Ce qu'on apprend

On crée un fichier de code standard dans le dossier partagé sur la machine hôte.

Logique ! Imagine le port 80 comme une place de parking. Le premier conteneur s'y est déjà garé. La place est prise.

Ce qu'on apprend

Conflit de ports. Un port spécifique sur la machine hôte (ici le 80) peut être déjà occupé par un autre service à la fois.

Planché 6 : Conclusion – Synthèse

Planche 6

Ce que Docker apporte (Le Top pour le Dev)

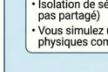


Checklist

- Démarrage éclair** (Secondes vs Minutes)
- Léger** (Pas d'OS complet)
- Reproductible** (La même image)
- Simple à exposer** (Gestion des ports)
- Idéal pour le Dév** (Volumes)

Pour les TP et le développement web, c'est vraiment plus fluide qu'une VM.

Quand préférer une VM ? (L'autre outil)



Utilise une VM si :

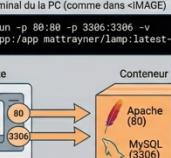
- Besoin d'un OS complet différent du vôtre
- Isolation de sécurité très forte (le noyau n'est pas partagé)
- Vous simulez un parc de machines physiques complètes

Je ne remplace pas totalement les VM. Nous sommes deux outils différents !

Activité 1 : Premier lancement et tests

Planche 1

Téléchargement & lancement



PC Hôte

Conteneur

Regarde bien : -p ouvre les portes (ports) et -v crée un point magique entre ton dossier local et le conteneur.

Ca marche ! Apache répond.

Oui, mais ton dossier 'app' est encore vide. C'est la page par défaut.

Test Apache

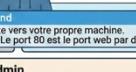


Index of /

Ce qu'on apprend

localhost pointe vers votre propre machine. Il est machine. Le port 80 est le port web par défaut, il est redirigé vers le conteneur.

Test phpMyAdmin



Utilisateur : admin
Mot de passe : fourni par l'enseignant

Et la base de données est là aussi !

Ce qu'on apprend

Un seul conteneur peut contenir plusieurs services (ici Apache + MySQL + PHP + phpMyAdmin) qui fonctionnent ensemble.

Planché 3 : Accès depuis le voisin : IP du PC vs IP du conteneur

Planche 3

Planche 5 : La solution (Changer les ports)

Planche 5

On corrige la commande

test.php

```
test.php
$ docker run -p 81:80 -p 3307:3306 -v $PWD/app:/app mattnayr/lamp:latest
1884
```

Voilà la solution : on dit à Docker : "Si quelqu'un appelle sur le port 81 de l'hôte, transfère-le sur le port 80 du deuxième conteneur."

Page internet de Sam...

Port 80

80

81

Conteneur 1 (comme dans l'IMAGE 0)

Hôte

Conteneur 2

Super ! localhost pour le premier site, localhost:81 pour le deuxième.

Ce qu'on apprend

Le port dans l'URL (ex : :81) permet de choisir à quel conteneur s'adresse sur la même machine.

Test depuis le voisin

Page internet de Sam...

Je peux accéder à tes deux environnements en même temps !

Ce qu'on apprend

Isolation et standardisation : Docker permet de multiplier les environnements de développement facilement sur un seul seul poste.