

[MISE EN SERVICE D'UN SERVEUR LAMP VIRTUALISÉ] [Index en fin de document](#)

OS : GNU/LINUX => UBUNTU NOBLE version 24 (desktop)

SERVICES => APACHE, MariaDB Server (MySQL), PHP 8.3, PHPMYADMIN

PRÉREQUIS :

Installation d'Oracle Virtual Box VM

Archive préinstallée de l'OS Ubuntu Desktop v.24.04 pour Virtual Box.

➔ FICHIER : [**Ubuntu_Desktop_24.04.ova**]

- Récupérer / transférer ce fichier depuis le NAS , dossier RESSOURCES ➔ VM
- Installer la VM sur VirtualBox (en double cliquant dessus).
- Configurer le **RÉSEAU VIRTUEL** d'Ubuntu24 par : « Accès par pont » (bridged).
- Démarrer la machine virtuelle et activer votre session **STORMSHIELD** du GRETA.
- Le mot de passe administrateur de la VM est : **ubuntu**

1/ Services LAMP : installation des applications

- Lancer le terminal Oubuntu depuis SSH (**PuTTY**) cf. document : *SSH.pdf*
- Mise à jour des registres système :
➔ `sudo apt update`
- Installer le serveur web **APACHE** :
➔ `sudo apt -y install apache2`
- Installer le SGBDR (serveur de base des données) **MySQL** :
➔ **FAIRE UNE SAUVEGARDE DE LA VM (SNAPSHOT)**
➔ `sudo apt -y install mariadb-server`
- Finir l'installation du SGBDR, avec la commande :
➔ `sudo mysql_secure_installation`

Sur la question « Enter current password (..) » saisir ENTRER (rien)

Sur la question « Switch to Unix Socket (..) » saisir **yes**Sur la question « Change root password (..) » saisir : **yes**Saisir le mot de passe 2 fois **ICI PAR SÉCURITÉ IL NE S'AFFICHE PAS** : `Root@dev123`

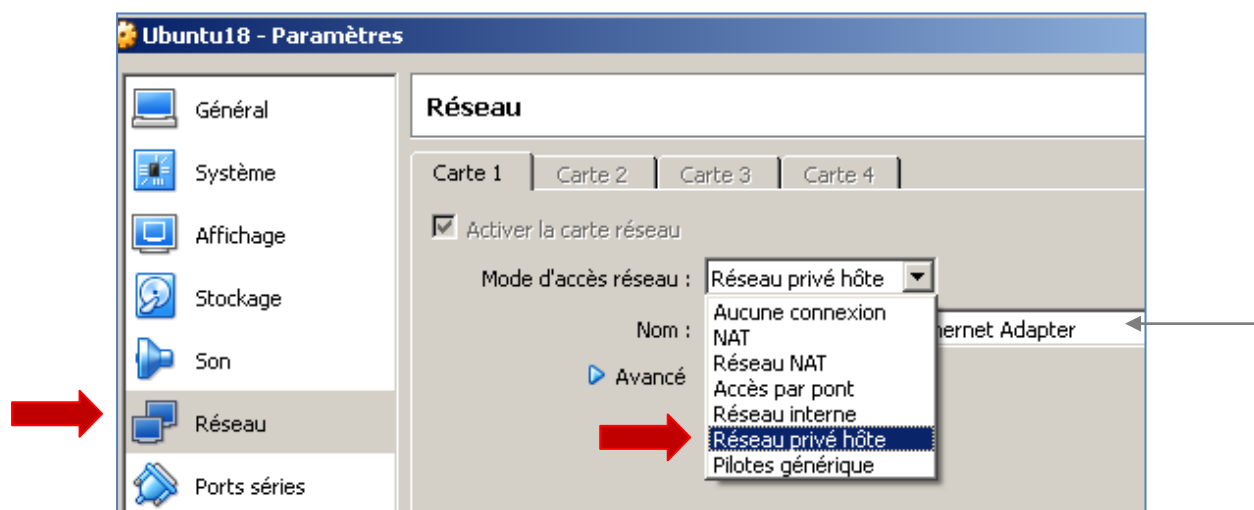
Pour chaque question posée, saisir les réponses proposées ci-dessous :

yes , **no** , **yes** , **yes** `All done!`

- Installer les binaires de **PHP** + les modules indispensables pour **APACHE** et **MySQL** :
➔ `sudo apt-get -y install php libapache2-mod-php php-mysql`
- Installer le client web MySQL : **PhpMyAdmin**
➔ `sudo apt-get -y install phpmyadmin`
-> Sélectionner « **☑ APACHE2** » avec **la barre d'espace**, puis valider avec la touche **ENTRER**.
-> Confirmer la mise à jour de la base « `dbconfig_common` » : **< oui >**
-> Ajouter le mot de passe deux fois : `Root@udev123`
➔ `sudo phpenmod mbstring && sudo service apache2 restart`
- Créer une sauvegarde de la VM d'Ubuntu en effectuant une **SnapShot** sur Virtual Box
Eteindre la VM d'Ubuntu par la commande :
➔ `sudo poweroff`

2/ Communication avec l'ordinateur Hôte - Premiers essais

- Pour communiquer avec *Windows*, nous devons changer les **paramètres** de l'interface réseau sur VirtualBox :



- Ici le **Réseau Privé Hôte** permet d'établir une connexion avec l'ordinateur hôte (ici *Windows*) par une **IP fixe** attribuée par VirtualBox.
- Si le nom de l'interface réseau n'apparaît pas, lancer la commande :
 - {repertoire d'install de VBOX} VBoxManage hostonlyif create
- Démarrer la machine virtuelle **ubuntu24**
- Lancer le terminal
- Pour obtenir l'**IP** du serveur ubuntu, lancer la commande → **ifconfig**
- Noter** l'**IP**, comme sur l'exemple ci-dessous :

```

ubuntu@ubuntu-VirtualBox: ~
Fichier Édition Affichage Rechercher Terminal Aide
ubuntu@ubuntu-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::5d4:28bd:c8b:ee64 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:01:70:a9 txqueuelen 1000 (Ethernet)
    RX packets 1110 bytes 107624 (107.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 129 bytes 35516 (35.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 78 bytes 6475 (6.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 78 bytes 6475 (6.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ubuntu@ubuntu-VirtualBox:~$
  
```



- Sur cet exemple, l'interface réseau détectée est nommée : **enp0s3**, l'IP v4 est nommée : **inet**. Une autre interface réseau : **lo** pour **local** est ici installée par défaut. Son IP sera toujours : **127.0.0.1**

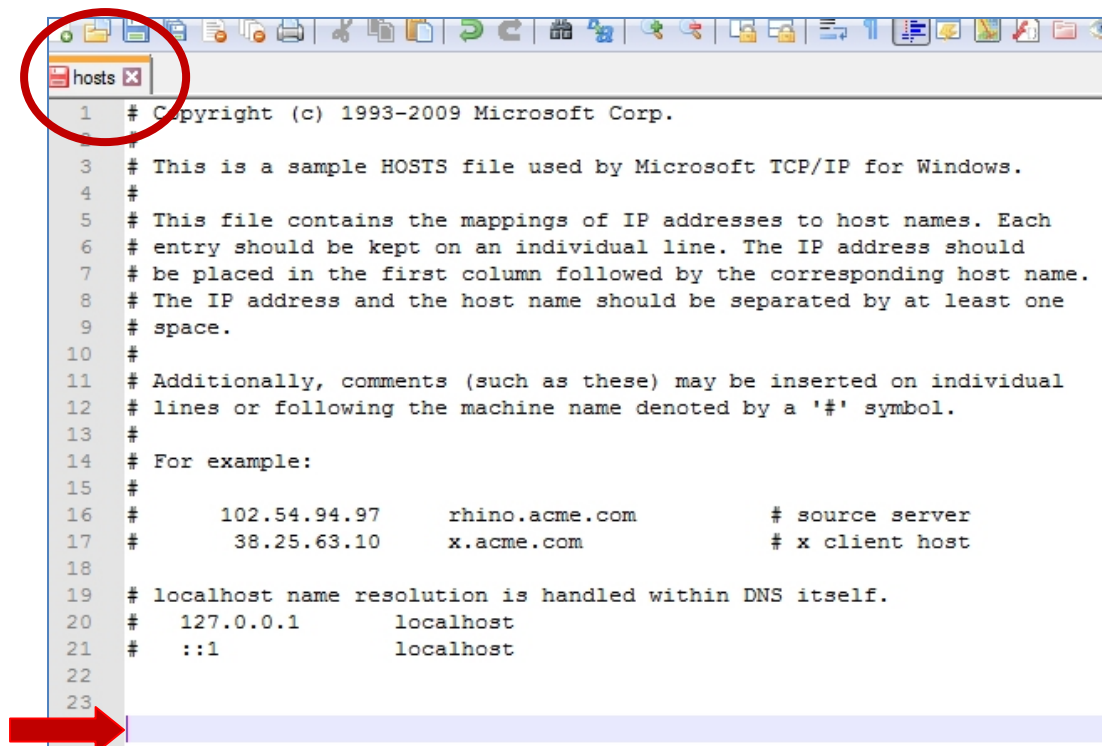
- Configuration du fichier **hosts** : **les noms d'hôtes**

Le fichier **hosts** de Windows permet de créer un « *nom de domaine* » local (ou "**nom d'hôte**"), et de cibler celui-ci sur une **IP** (généralement une IP locale). Celui-ci se trouve dans le dossier :

C:\WINDOWS\system32\drivers\etc\hosts

Il peut être modifié avec un éditeur de texte comme **notepad++** :

Important : Ce fichier doit s'ouvrir avec **des droits d'administrateurs**



The screenshot shows the Windows hosts file in Notepad++. A red circle highlights the 'hosts' tab in the title bar. A red arrow points to the end of the file content, indicating where to add new entries. The file content is as follows:

```
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 #
19 # localhost name resolution is handled within DNS itself.
20 # 127.0.0.1       localhost
21 # ::1            localhost
22
23
```

Exemple du fichier hosts par défaut

- Modifier ce fichier en ajoutant cette formule à la fin du texte :

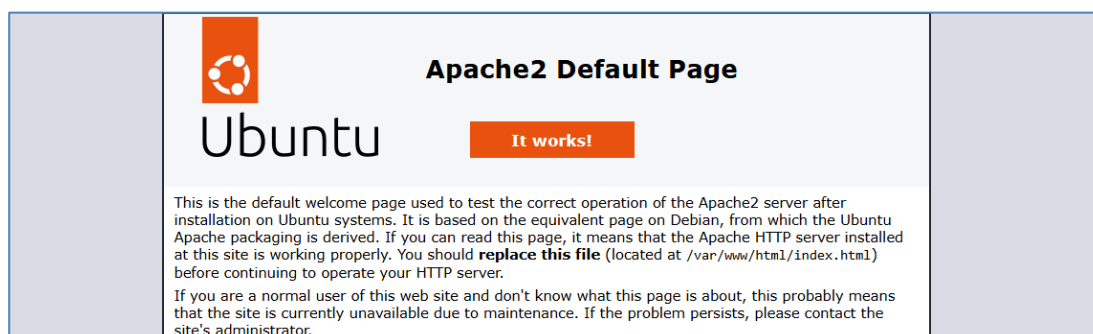
IP du serveur (noté précédemment) + **Un espace ou une tabulation** + **Un nom d'hôte**

Par exemple :

192.168.56.101

kercode.vm

- Enregistrer le fichier **hosts**
- Sur Windows, ouvrir un navigateur (*Chrome, Firefox, ...*) et saisir votre nom d'hôte :
➔ <http://kercode.vm>



Si cette page apparaît, vous avez correctement réussi à installer le service web Apache !
Bien lire le document qui s'affiche, notamment la ligne :

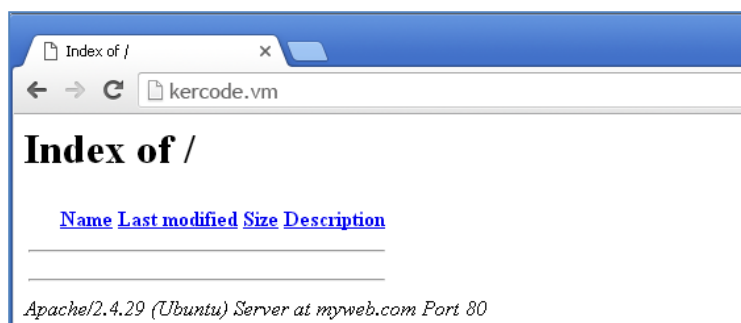
« **You should replace this file located at /var/www/html/index.html** (...) »

Pour commencer à alimenter le serveur web, il faut donc supprimer la page web par défaut :

- Sur la VM d'ubuntu, lancer le terminal.
- Se déplacer dans le répertoire web par défaut avec la commande :
→ `cd /var/www/html`
- Supprimer le fichier index par défaut :
→ `sudo rm index.html`
- Sur Windows, recharger la page :
→ <http://kercode.vm>



- Que peut-on constater ?
- Quel est l'intérêt de nommer une page web : **index.html** sur notre serveur ?



3/ Création d'une nouvelle page web par défaut

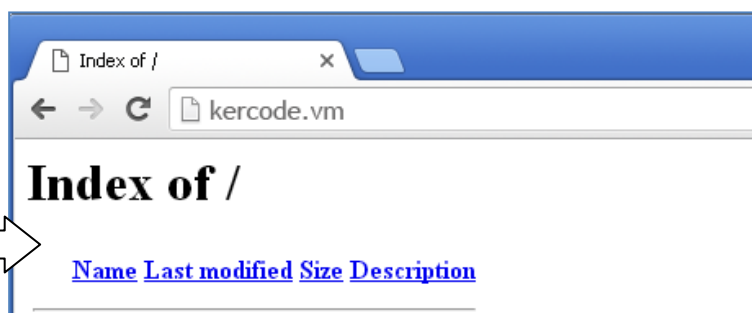
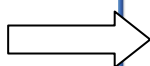
- Sur le **terminal** Oubuntu
- Créer un nouveau document avec l'utilitaire de texte GNU nommé : **nano**
→ `sudo nano index.html` [modifiez les parties surlignées ci-dessous] :

```
<!doctype html>
<html>
<head>
<meta charset="UTF-8" />
<title>un titre ?</title>
</head>

<body>
<h1>Ici quelque chose ?</h1>
<hr>
<i>Mon serveur web fonctionne ?</i>
</body>
</html>
```

- Sauvegarder le fichier par la commande : **CTRL+O**, appuyer sur **ENTREER** pour confirmer.
- Quitter nano avec **CTRL+X**
- Sur Windows, recharger la page :
→ <http://kercode.vm>

Ici le contenu
personnalisé doit
se mettre à jour!

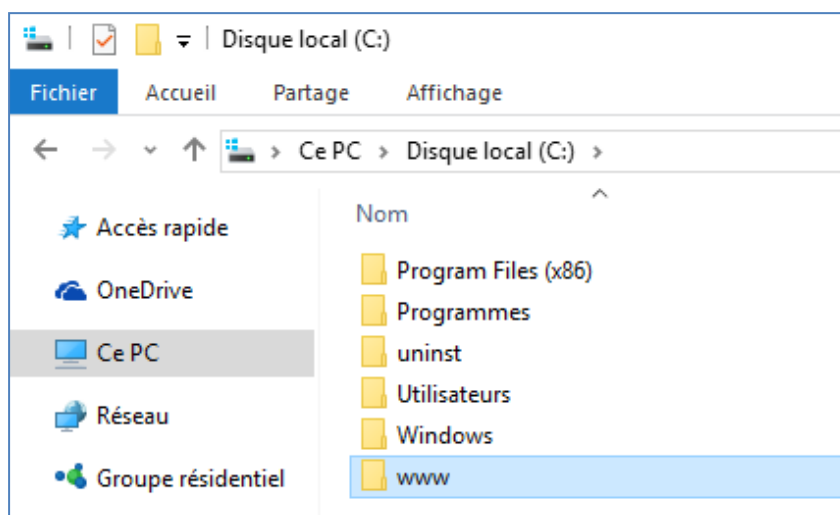


4/ Introduire un environnement de développement sur Windows : le partage WEB

Sur cette partie nous allons choisir un dossier *Windows* comme répertoire par défaut de notre serveur **LAMP**. Ceci va simplifier et alimenter le serveur web directement depuis l'hôte Windows !

a) Partage sur Windows

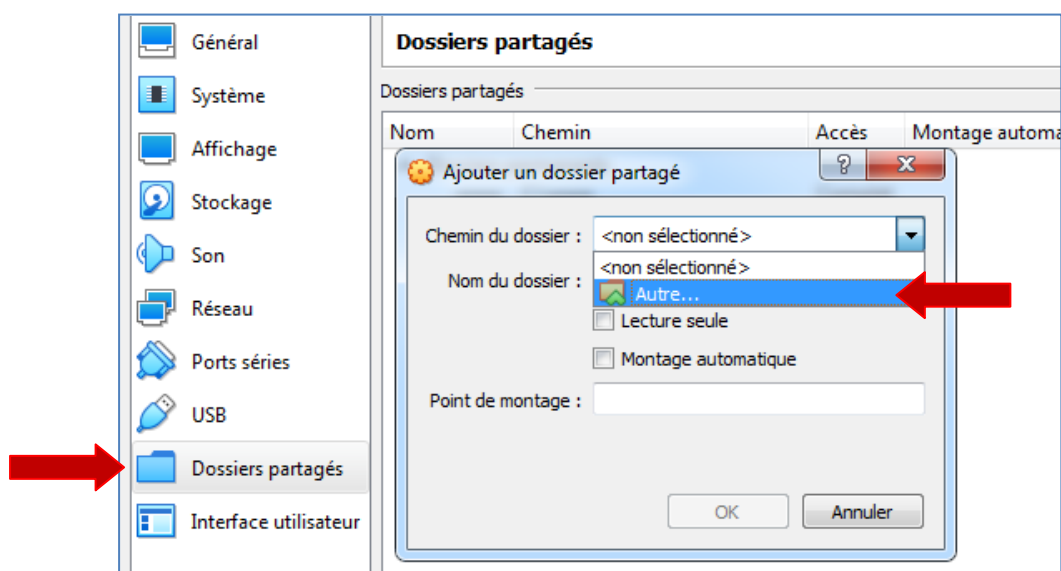
- Sur l'explorateur *Windows*, à la racine de votre répertoire « **C :** » créer un nouveau dossier nommé : « **www** », soit : **C:\www**

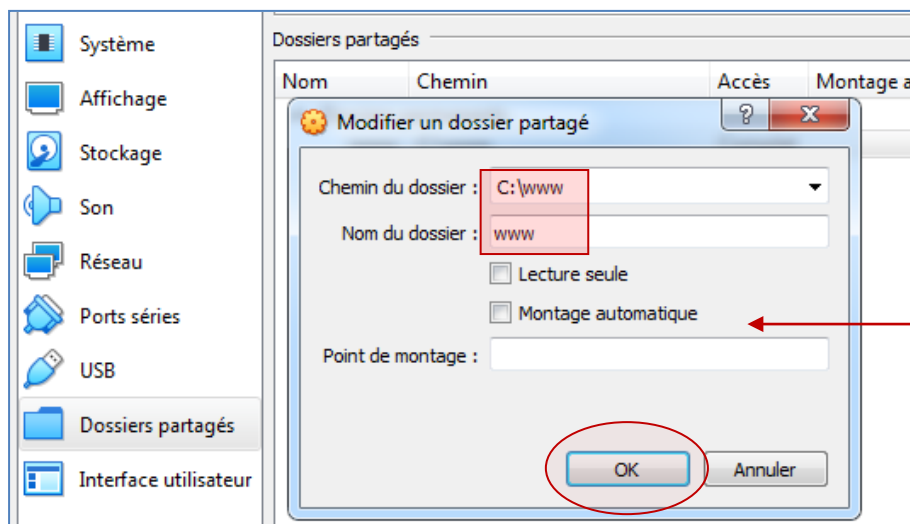


Note : Ce dossier sera utilisé comme répertoire Web par défaut.

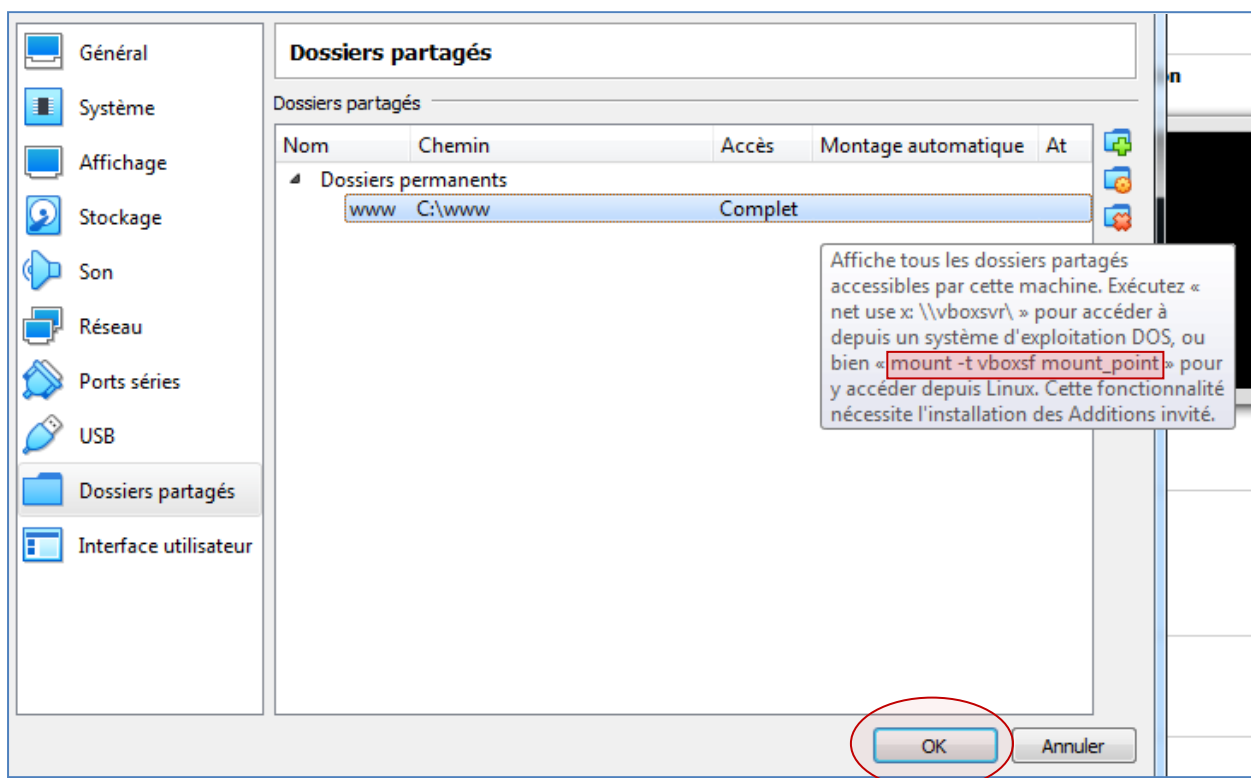
b) Partage sur Virtual Box

- Ouvrir les paramètres de la VM **ubuntu24**, sélectionner le menu « **Dossiers partagés** », recherchez le dossier « **www** » crée précédemment :





NOTE : Si l'option « *Configuration permanente* » est présente, cocher cette option : ☒ Configuration permanente



c) Activation du partage sur la VM

- Sur le terminal Ubuntu , lancer la commande :
→ `sudo mount -t vboxsf www /var/www/html`
- Sur Windows, déposez quelques fichiers (images, documents HTML, pdf,...) dans le répertoire : **C:\www**
- Saisir la commande : → `ls /var/www/html`
L'ensemble des documents présents dans : **c:\www** doivent s'afficher !
- Sur votre navigateur, recharger la page :
→ <http://kercode.vm>

A ce stade, il est indispensable de **créer une Snapshot** de votre VM d'**Ubuntu**.

5/ Automatisation du partage par ligne de commande simplifiée

Le but est de pouvoir activer / désactiver le partage sur Windows dans **C:\www**, ceci par une simple ligne de commande : « **share on** » ou « **share off** » :

- Récupérer le fichier *Shell Script* nommé « **share.sh** » depuis le NAS, dossier **RESSOURCES**.
- Placer ce fichier dans : **C:\www**
- Sur le terminal Oubuntu, lancer la commande → **cd** pour se placer dans le répertoire personnel de l'utilisateur « *ubuntu* ».
- Copier le fichier « **share.sh** » par la commande :
→ **cp /var/www/html/share.sh ./**
- Rendre ce fichier exécutable par la commande :
→ **chmod +x share.sh**
- Modifier le fichier système « **.bashrc** », exécuter ces deux commandes :
→ **sudo echo 'echo "Pour activer le partage sur www, lancer la commande : [share on]"' >> .bashrc**
→ **sudo echo "alias share=/home/ubuntu/share.sh" >> .bashrc**
- On recharge le fichier système « **.bashrc** » :
→ **source .bashrc**
- On teste la commande :
→ **share on**
- Sur votre navigateur, recharger la page :
→ <http://kercode.vm>
- On teste la commande :
→ **share off**
- Sur votre navigateur, recharger la page :
→ <http://kercode.vm>

Par défaut, réactiver le partage sur Windows → **share on**

Si vous avez rencontré des problèmes sur cette partie, la commande :

→ **sudo mount -t vboxsf www /var/www/html**

sera toujours possible pour activer le partage sur *Windows* dans le dossier partagé : **C:\www**

6/ Premières pages PHP

Avec l'éditeur de texte de votre choix (*NotePad++*, *VScode*, *etc...*) créer un fichier nommé : « **index.php** » dans le répertoire : **C:\www**

- Insérer le code suivant :

```
<?php
/* Diagnostic complet du service PHP */
phpinfo();
?>
```

- Sur votre navigateur, recharger la page :
→ <http://kercode.vm>

PHP Version 8.3.6



System	Linux ubuntu-VirtualBox 6.8.0-52-generic #53-Ubuntu SMP PREEMPT_DYNAMIC Sat Jan 11 00:06:25 UTC 2025 x86_64
Build Date	Dec 2 2024 12:36:18
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/15-xml.ini, /etc/php/8.3/apache2/conf.d/20-

Le fichier **php.ini** permet de configurer des nombreuses options pour le service PHP.

Autre exemple avec PHP afin de générer une page HTML supportant plusieurs images :

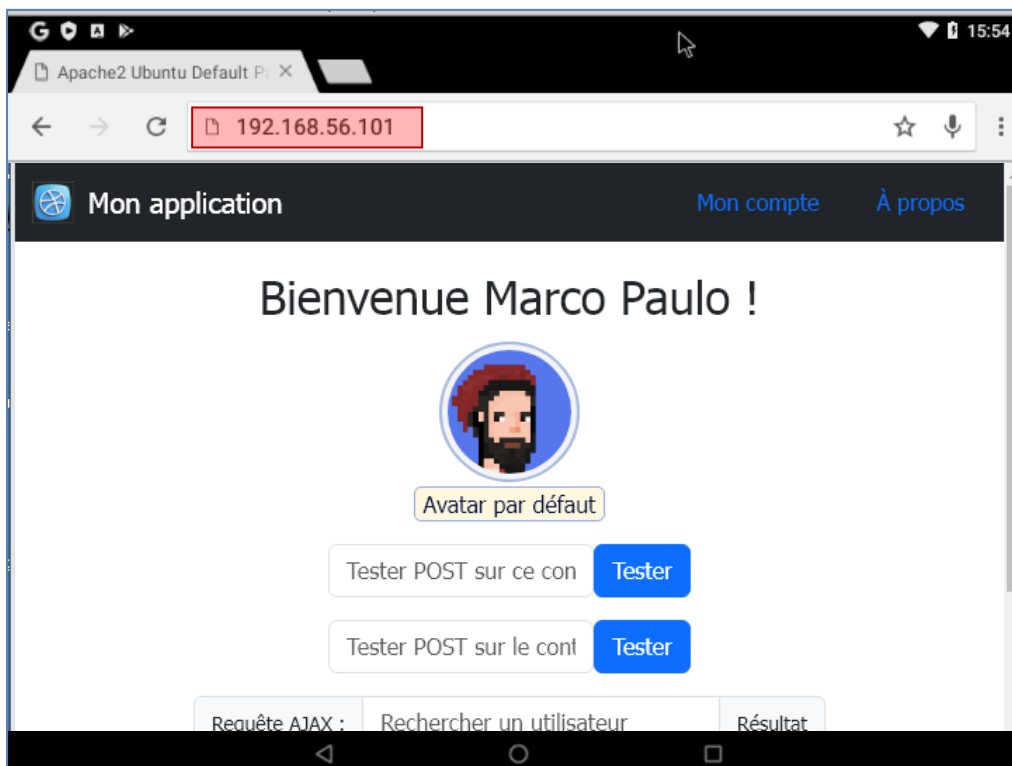
- Dans le répertoire : **C:\www** , créer un nouveau dossier nommé : « **test** ».
- Dans le dossier « **test** », créer un dossier nommé : « **images** ».
- Déposer dans ce dossier « **images** » quelques images de votre choix au format JPEG ou PNG.
- Dans le dossier « **test** », créer un nouveau fichier « **index.php** ». Ajouter ce code :

```
<html>
<head>
<style>
img {
    max-width:50%;
    height:auto;
    padding: 5px;
    border: 2px solid darkgrey;
}
</style>
</head>
<body>
<div style="text-align:center;">
    <h1><u>IMAGES</u></h1>
    <?php
    $glob_dir = "./images/";
    $files = glob($glob_dir . "*");
    foreach($files as $image) {
        $i++;
        echo "
        <p>
        <li><h2>Image ".$i." :</h2></li>
        <img src='".$image.'" /><br />
        <a href='".$image.">".basename($image)."</a>
        </p>
        ";
    }
    if(!$files)echo "Pas d'images.";
    ?>
</div>
</body>
</html>
```

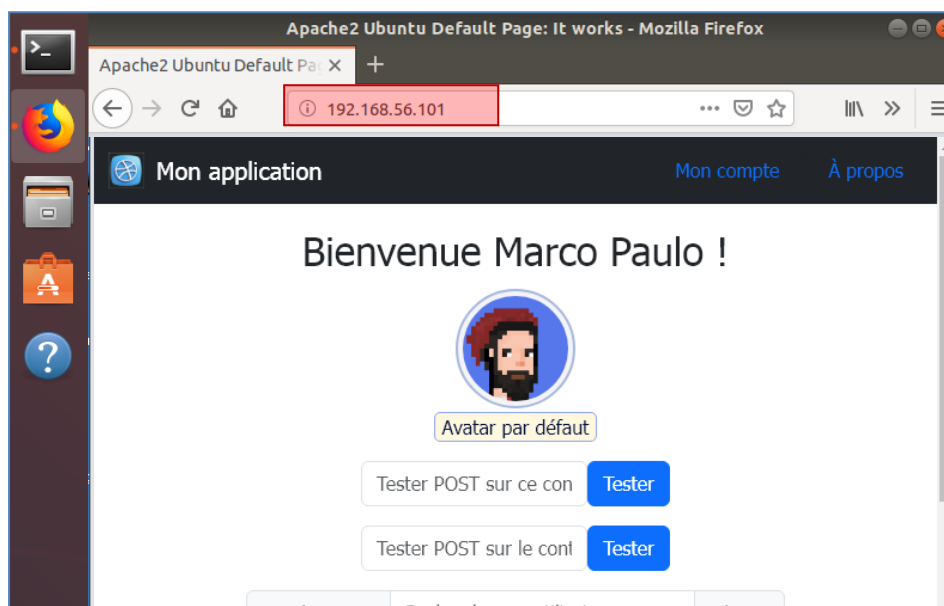

- Sur votre navigateur, tester le script HTML/CSS/PHP :
→ <http://kercode.vm/test>

7/ Clients et serveur : tester le serveur LAMP sur une autre VM

- Archiver le dossier : **c:\www** , ou vider tout son contenu.
- Copier / coller votre projet en cours (HTML/JS/CSS) dans le dossier : **c:\www**
- Sur Virtual Box, démarrer la machine virtuelle **Android 8** (nougat).
- Lancer l'application « *Chrome Android* » et saisir l'**IP** du serveur (Cf. page n°2) :



- Sur la VM d'**ubuntu24**, lancer l'application « *Firefox Linux* ». Saisir à nouveau l'**IP** du serveur :



8/ Base de données et clients MySQL

a) Pour commencer, nous allons nous connecter sur le client Shell **MySQL**.

- Sur la VM ubuntu24, lancer le terminal.
- On souhaite accéder au SGBDR (MySQL). Saisir la commande :
→ **sudo mysql -uroot -pRoot@dev123**

b) Création d'une nouvelle base, d'une table, et découverte du langage SQL

- Sur MySQL, créer une nouvelle base de données :

→ **create database test ;**

- On sélectionne la base créée :

→ **use test ;**

- Puis on crée une nouvelle table :

→ **create table ma_table(colonne1 int, colonne2 text) ;**

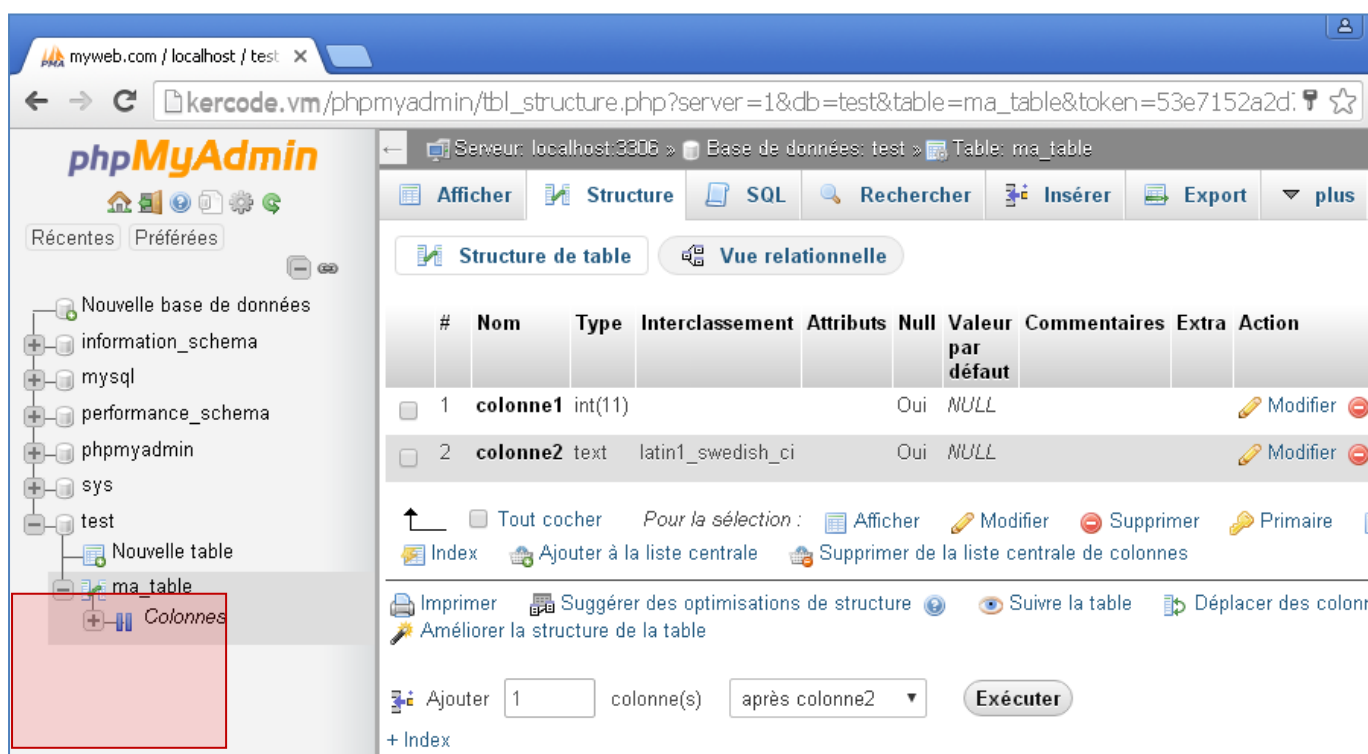
- Puis on quitte le client Shell MySQL :

→ **exit**

- Tester le client phpMyAdmin à l'adresse : <http://kercode.vm/phpmyadmin>

=> Utilisateur : **root**

=> Mot de passe : **Root@dev123**



c) Tester l'accès à une base et ses tables depuis un script PHP

- Récupérer sur le NAS => dossier **RESSOURCES** le fichier « **dbtest.php** ».
- Déposer le script dans : « **c:\www** ».
- Modifier les paramètres de connexion, comme sur l'exemple ci-dessous :

```
<?php

# $ php -f dbtest.php

$PARAM_nom_bd = 'test'; // nom de la base de données
$PARAM_utilisateur = 'root'; // nom d'utilisateur
$PARAM_mot_passe = 'Root@dev123'; // mot de passe de l'utilisateur
$PARAM_hote = 'localhost'; // serveur local (redirige vers MYSQL via le port 3306 par défaut)
```

- Tester le script PHP depuis l'adresse :
➔ <http://kercode.vm/dbtest.php>



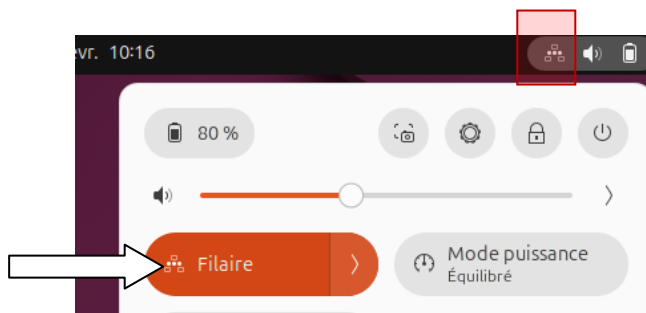
Ce type de connexion à une base de données utilise le module **PDO** de PHP.

Dans un contexte professionnel, on utilisera le PHP object associé à la méthode **DAO** (*Direct Access Object*) pour contrôler - avec le langage SQL une base de données distante...

9/ Versions de PHP

Certain projet au besoin de faire fonctionner d'autres versions de PHP. Ici nous allons installer la version **8.4** et configuration cette version par défaut pour le serveur Aapche.

- Changer la connexion réseau sur les configurations VBOX : « **NAT** »
- Eteindre puis réactiver la connexion filaire sur Ubuntu :



Lancer la commande :

➔ **ifconfig** (*l'IP v4 doit être changée*)

Vérifier que l'IP **10.0.2.15** s'affiche

```
ubuntu@ubuntu-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>
    inet 10.0.2.15 netmask 255.255.255.0
    inet6 fe80::a00:27ff:fec8:b3db prefixlen 64
    inet6 fd00::3562:5911:386a:500a prefixlen 64
```

La liste des commandes ci-dessous **ne doit pas renvoyer d'erreurs**, une connexion à Internet avec les **autorisations du STORMSHIELD** doivent être active :

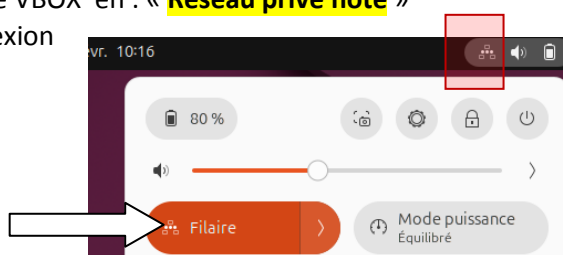
- Ajout du PPA (ou liste de paquets non officiels) de son auteur [ondrej](#), installation de PHP 8.4, le module pour le serveur Apache et quelques extensions :

➔ `sudo add-apt-repository ppa:ondrej/php` *Sur la question saisir : ENTREE*
 ➔ `sudo apt update`

- Attention, la commande ci-dessous **sur une seule ligne** :

➔ `sudo apt install -y php8.3-fpm php8.4 php8.4-common php8.4-fpm php8.4-mysql php8.4-curl php8.4-gd php8.4-gettext php8.4-cgi php8.4-json php8.4-imap php8.4-intl php8.4-mbstring php8.4-opcache php8.4-soap php8.4-zip php8.4-xml php8.4-xmlrpc && sudo a2dismod php8.3 && sudo a2enmod proxy_fcgi setenvif && sudo a2enconf php8.3-fpm && sudo service apache2 restart`

- Changer la connexion réseau de VBOX en : « Réseau privé hôte »
- Eteindre puis réactiver la connexion filaire sur Ubuntu :



Lancer la commande :

➔ `ifconfig` (l'IP v4 doit être changée)

Modifier la configuration par défaut d'Apache. La partie en bleu est à ajouter :

➔ `sudo nano /etc/apache2/sites-available/000-default.conf`

```
<VirtualHost *:80>
<FilesMatch \.php$>
    SetHandler "proxy:unix:/var/run/php/php7.4-fpm.sock|fcgi://localhost"
</FilesMatch>
```

On relance le serveur Apache:

➔ `sudo service apache2 restart`

- Tester le script PHP depuis l'adresse (voir la partie [phpinfo\(\)](#) vue précédemment) :
 ➔ <http://kercode.vm>

PHP Version 8.4.1	
	
System	Linux ubuntu 6.8.0-45-generic #45-Ubuntu SMP PREEMPT_DYNAMIC Fri Aug 30 12:02:04 UTC 2024 x86_64
Build Date	Nov 21 2024 14:54:00
Build System	Linux
Server API	Apache 2 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.4/apache2
Loaded Configuration File	/etc/php/8.4/apache2/php.ini

Le serveur LAMP virtualisé est maintenant pleinement fonctionnel !

Libre à vous d'ajouter bien d'autres fonctionnalités ou solutions, comme *GIT*, *NODE*, *NPM*, *REACT*, *VUE*, *COMPOSER*, *LARAVEL*, *SYMPHONY*, des CMS comme *WORDPRESS*, *DUPRAL*, *SPIP*, les protocoles de sécurité *TLS/SSL*, des fonctionnalités de messagerie comme *SMTP* avec *POSTFIX*, *FTP*, etc...

[MISE EN SERVICE D'UN SERVEUR LAMP VIRTUALISÉ]

INDEX DES PARTIES ABORDÉES

- [Services LAMP : installation des applications](#)
- [Communication avec l'ordinateur Hôte - Premiers essais](#)
- [Création d'une nouvelle page web par défaut](#)
- [Introduire un environnement de développement sur Windows : le partage WEB](#)
- [Automatisation du partage par ligne de commande simplifiée](#)
- [Premières pages PHP](#)
- [Clients et serveur : tester le serveur sur une autre VM](#)
- [Base de données et clients MYSQL](#)
- [Migration vers PHP 8.4](#)