

Rapport Installation serveur

Sommaire :

1 — Informations connexion SSH	(page 1)
2 — Installation apache	(page 2)
3 — Installation PHP	(page 3)
4 — Installation base de données	(page 4)
5 — Configuration base de données	(page 5)

INFORMATIONS CONNEXION SSH

Connexion sur le RASPBERRY PI :

ssh saepi@192.168.1.167

Mot de passe :

Ju47zxk57zpU

Changer de mot de passe :

passwd saepi

sudo apt update

(Mise à jour des packages de debian)

INSTALLATION APACHE

- Installation
 - `sudo install apache2 -y` (Installation du serveur Apache)
 - `systemctl start apache2` (Lancement du serveur)
 - `systemctl enable apache2` (Activation du serveur si non activé)
- Réglage du pare-feu
 - `firewall-cmd --permanent --zone=public --add-service=http`
 - `firewall-cmd --permanent --zone=public --add-service=https`
 - `firewall-cmd --reload` (rechargement du pare-feu)
- Retour sur l'installation / Commande utile
 - Administration
 - `systemctl status apache2` (Vérifier l'état du service Apache)
 - `apache2 -v` (Vérifier la version du service Apache)
 - `sudo chown -R pi:www-data /var/www/html` (Droit au dossier)
 - `sudo chmod -R 770 /var/www/html/` (Droit au dossier)

Commande	Signification
<code>systemctl stop apache2</code>	Arrêter le service Apache
<code>Systemctl start apache2</code>	Démarrer le service Apache
<code>Systemctl restart apache2</code>	Redémarrer le service Apache
<code>systemctl reload apache2</code>	Recharger le service Apache
<code>systemctl disable apache2</code>	Désactiver le service Apache au démarrage
<code>systemctl enable apache2</code>	Réactiver le service Apache au démarrage

- A savoir sur le serveur Apache

→ Les fichiers Apache se trouvent dans `/etc/httpd`
→ Fichier de configuration est `/etc/conf/httpd.conf`
→ Les fichiers de configuration se termine par `.conf` dans `/etc/httpd/conf.d` sont inclus dans le fichier de configuration principal d'Apache
→ Fichier de configuration pour les modules sont dans `/etc/httpd/conf.modules.d`
→ Fichier journaux d'Apache dans `/var/log/httpd`
→ Pour les Webroot (modules)/ racine du site dans `/var/www/html/` ou `/var/www/`

INSTALLATION PHP

- Installation
 - `sudo apt install php php-mbstring php-gd php-mysqlnd php-zip php-soap php-ldap php-curl`
(installation de php)
- Administration
 - `sudo rm /var/www/html/index.html` (Suppression du fichier index.html)
 - `touch /var/www/html/index.php` (Création d'un fichier)
 - `echo "<?php phpinfo(); ?>" > /var/www/html/index.php` (mettre dans le fichier la commande)

Tout est prêt pour la pile Lamp. On peut maintenant voir si php fonctionne correctement

→ 192.168.1.167 (dans un navigateur)

`phpinfo();` est une fonction de php qui permet d'afficher la configuration php de notre site. En tapant l'ip de la machine avec `/info.php`, on tombe sur cette page.



System	Linux ajaniserveur 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64
Build Date	Dec 12 2013 08:42:50
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-gd.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525 NTS

Capture d'écran de phpinfo

INSTALLATION BASE DE DONNÉES

- Installation
 - `sudo apt install mariadb mariadb-server` (Installation du serveur de base de données)
 - `sudo apt install php-mysql` (servira à faire le lien entre php et mysql)

`systemctl enable mariadb` (Activation du serveur de base de données)
`systemctl start mariadb` (Lancement du serveur de base de données)

`mysql_secure_installation` (Sécurisation de l'installation du serveur de base de données. On donne un nouveau mot de passe.)

`sudo mysql --user=root` (Connexion pour le vérifier le fonctionnement de Mysql)

Nous allons maintenant supprimer l'utilisateur root par défaut et créer un nouvel utilisateur root, car celui par défaut n'est utilisable que par le compte administrateur du système, et n'est donc pas accessible aux script PHP du serveur :

`DROP USER 'root'@'localhost';`

`CREATE USER 'root'@'localhost' IDENTIFIED BY 'password';`

*`GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;`*

CONFIGURATION DE LA BASE DE DONNÉE

Pour tester la connexion à la base de données, on exécute les commandes suivantes:
`mysql -u root -p;` on ajoute le mot de passe. (Connexion à la base de données)

Identifiant de la base de données: root

Mot de passe : q4ToTaeWRz3QHd3V

`create database <nom_database>` (Création d'une base de données quelconque)

`create user 'utilisateur@localhost' identified with mysql_native_password by 'password'`
(Création d'un utilisateur)

`grant all on <nom_user>* to utilisateur@localhost;` (Attribution des droits de l'utilisateur)

On peut maintenant quitter la base de données avec : 'quit', et se reconnecter avec l'utilisateur créé :
`mysql -u utilisateur -p;` on ajoute le mot de passe.

Nous avons finalement décidé d'installer PhpMyAdmin pour avoir une interface graphique plus simple à utiliser.

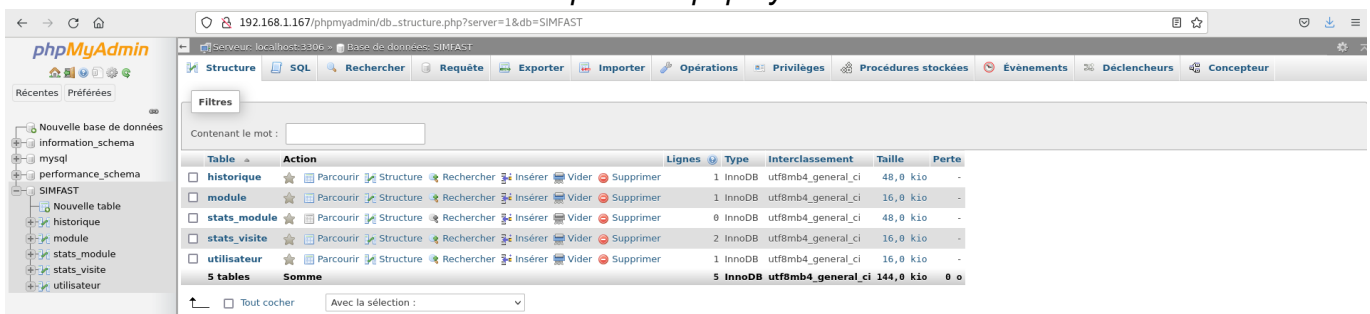
Pour cela, on active l'extension mysqli :

`sudo phpenmod mysqli`

`sudo /etc/init.d/apache2 restart`

Puis on installe phpmyadmin sur le serveur :

`sudo apt install phpmyadmin`



Capture d'écran de phpmyadmin

C'est un rapport sur l'installation du serveur (des services) et non pas la création de la base de données, les requêtes de la base de données seront dans un autre livrable.