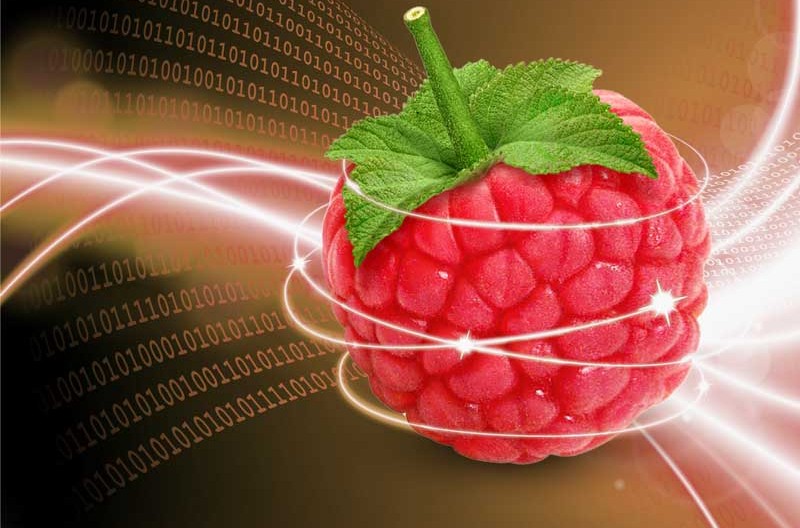
****

**Installer un serveur web sur votre Raspberry (Apache + PHP + MySQL)**

Mis à jour le 9 avril 2018 - 214 Commentaires - [Apache](https://raspbian-france.fr/tag/apache/), [Installation](https://raspbian-france.fr/tag/installation/), [Ligne de commande](https://raspbian-france.fr/tag/ligne-de-commande/), [Linux](https://raspbian-france.fr/tag/linux/), [MySQL](https://raspbian-france.fr/tag/mysql/), [PHP](https://raspbian-france.fr/tag/php/), [Planet-libre](https://raspbian-france.fr/tag/planet-libre/), [Raspbian](https://raspbian-france.fr/tag/raspbian/), [Serveur](https://raspbian-france.fr/tag/serveur/), [tutoriels](https://raspbian-france.fr/tag/tutoriels/)

Après avoir créé votre carte SD, et après avoir démarré pour la première fois votre Raspberry Pi, il y a de fortes chances que vous souhaitiez l’utiliser comme un serveur web.

**La Raspberry Pi comme serveur web, pourquoi ?**

Mais **pourquoi utiliser la Raspberry comme un serveur Web**, plutôt que de faire appel à des prestataires de services spécialisés dans l’hébergement web ?

Tout d’abord, d’un point de vue économique, il faut savoir que **les services d’hébergement web ne sont pas gratuits** et qu’il faut sortir le porte-monnaie tous les mois/ans. Contrairement à la Raspberry qui **demande juste une connexion.**  
De plus, en choisissant la Raspberry, **vous avez la possibilité de modifier vos services à souhait** (exemples : la taille du disque, l’hébergem­ent de Database, etc.), ce qui n’est en général **pas le cas chez les hébergeurs spécialisés**, qui vendent souvent des hébergements mutualisés avec **une faible capacité de configuration.**  
Cependant **pour supporter plus d’utilisateurs**, vous avez intérêt à utiliser [une Raspberry Pi 3](https://geni.us/rflraspberrypi3), soit la Raspberry Pi avec **1 Go de RAM**, plutôt que la [Raspberry de type B+](https://geni.us/rflraspberrypibplus) (512 Mo de RAM)

La question qui se pose maintenant est, **comment faire de la Raspeberry un serveur web** ?

**Installation du serveur Apache avec Raspbian**

**Apache, c’est quoi ?**

Tout d’abord, nous allons installer Apache, qui est **le serveur web en tant que tel**.  
Quand on parle de serveur web, on pense souvent à la machine, mais **ce terme désigne aussi le logiciel** qui permet à la machine d’analyser les requêtes d’un utilisateur (sous forme http), et de retourner le fichier correspondant à la requête (ou une erreur si le fichier n’est pas trouvé, ou la requête mal formulée).  
Dans le cadre d’Apache, **c’est donc du logiciel que l’on parle.**

À l’heure actuelle, **Apache est le serveur web le plus utilisé**, avec environ **60 % de parts de marché**. Apache possède même **sa propre licence**, utilisée par de nombreux autres projets. De plus, l’utilisation massive d’Apache (devenu le standard des serveurs web), couplée à sa forte popularité, à amener à une formidable abondance [de documentation, de cours, et autres livres traitant de son utilisation, depuis l’installation jusqu’à la sécurisation.](http://www.amazon.fr/gp/product/2746018195/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&camp=1642&creative=6746&creativeASIN=2746018195&linkCode=as2&tag=raspbianfranc-21)

Que ce soit pour la Raspberry Pi et Raspbian, ou pour une machine plus généraliste, **Apache est donc un choix sûr**, et les compétences que vous pourrez acquérir sur le sujet **vous seront toujours utiles.**

**Installation d’Apache**

Avant d’installer le serveur, assurons nous d’avoir une machine bien à jour. Pour ce faire nous devons **posséder les droits administrateur**, soit en étant connecté en root, soit via la commande sudo.

sudo apt update

sudo apt upgrade

sudo apt update

Une fois la Raspberry Pi à jour, nous allons installer le serveur Apache.

sudo apt install apache2

Au passage, nous allons en profiter pour donner des droits au dossier d’apache qui vous permettra de facilement administrer les sites. Pour cela, lancez les commandes suivantes :

sudo chown -R pi:www-data /var/www/html/

sudo chmod -R 770 /var/www/html/

**Vérifier qu’Apache fonctionne**

Une fois l’installation terminée, nous pouvons **tester qu’Apache fonctionne correctement** en nous rendant sur l’adresse de la Raspberry.  
Pour cela, il faut tenter d’accéder à la Raspberry depuis le port 80 (ce port n’étant pas encore ouvert depuis l’extérieur, **il faudra le faire depuis la Raspberry elle-même**). Ne vous inquiétez pas, c’est très facile. Il vous suffit d’ouvrir le navigateur web de la Raspberry (par défaut Midori sous Raspbian), et d’aller à l’adresse « http://127.0.0.1 ». Vous devriez alors obtenir une page avec un message du genre « *It works!* » et plein d’autre texte.  
Si jamais **vous ne possédez pas d’interface graphique sur votre Raspbian**, ou que vous utilisez le SSH pour vous connecter à votre Raspberry, vous pouvez utiliser la commande suivante :

wget -O verif\_apache.html http://127.0.0.1

Cette commande va **enregistrer le code HTML de la page dans le fichier « verif\_apache.html »** dans le répertoire courant.  
Vous n’avez donc plus qu’à lire le fichier avec la commande

cat ./verif\_apache.html

Si vous voyez marqué à un endroit dans le code « *It works!* », c’est qu’Apache fonctionne.

**Apache utilise le répertoire /var/www/html comme racine pour votre site.** Cela signifie que quand vous appelez votre Raspberry sur le port 80 (http), Apache cherche le fichier dans /var/www/html.

Par exemple, si vous appelez l’adresse http://127.0.0.1/example, Apache cherchera le fichier example dans le répertoire /var/www/html.  
Pour ajouter de nouveaux fichiers, sites, etc., vous devrez donc **les ajouter dans ce répertoire.**

Vous pouvez **dès à présent** utiliser votre Raspberry pour faire un site **en HTML,  CSS et JavaScript pur**, en interne.  
Cependant, vous voudrez sans doute rapidement **permettre des interactions entre le site et l’utilisateur**. Par exemple permettre à l’utilisateur de s’inscrire, etc. Pour cela, vous allez avoir besoin de PHP.

**Installation de PHP sur la Raspberry**

**Qu’est-ce que PHP ?**

Tout d’abord, il faut savoir que le **PHP est un langage interprété**. Et comme dans le cas des serveurs, l’acronyme PHP **peut avoir plusieurs sens**. En fait, quand l’on parle de PHP, on peut parler **soit du langage, soit de l’interpréteur.**  
Ici, quand nous parlons d’installer PHP, cela signifie que **nous allons installer l’interpréteur**, afin d’utiliser le langage.

PHP (le langage cette fois) est **principalement** utilisé pour rendre un site dynamique, c’est-à-dire que l’utilisateur envoie des informations au serveur qui lui renvoie **les résultats modifiés en fonction de ces infos**. A contrario, un site statique **ne s’adapte pas aux informations fournies par un utilisateur.** Il est enregistré sous forme de fichier **une fois pour toute,** et livrera toujours le même contenu.

**PHP est libre**, et maintenu par la fondation PHP, ainsi que l’entreprise Zend, et diverses autres entreprises (il est à noter que Zend est aussi l’auteur [du célèbre framework Zend PHP, très utilisé et reconnu dans le monde de l’entreprise](https://geni.us/rfllivrezendframework)).

C’est **un des langages de programmation les plus utilisés**, et c’est même **le plus utilisé pour la programmation destinée au web**, avec environ **79 % de parts de marché.**

Là encore, l’ensemble des compétences que vous pourrez acquérir, sur le langage ou sur l’installation et la configuration de l’interpréteur, **vous seront toujours utiles.** Nous ne pouvons donc que vous conseiller [d’apprendre le PHP](https://geni.us/rfllivrephp55), qui est vraiment un langage merveilleux et trop souvent sous-estimé.

**Installer PHP**

On va encore une fois faire appel à **l’administrateur** pour installer PHP avec la ligne de commande.

sudo apt install php php-mbstring

**Vérifier que PHP fonctionne**

Pour savoir si PHP fonctionne correctement, ce n’est pas très compliqué, et **la méthode est relativement proche de celle employée pour Apache.**

Vous allez en premier lieu **supprimer le fichier « index.html »** dans le répertoire « /var/www/html ».

sudo rm /var/www/html/index.html

Puis **créez un fichier « index.php »** dans ce répertoire, avec cette ligne de commande

echo "<?php phpinfo(); ?>" > /var/www/html/index.php

À partir de là, le fonctionnement **est le même que pour la vérification d’Apache.** Vous tentez d’accéder à votre page, et vous devriez avoir un résultat proche de cette image (si vous n’avez pas d’interface, utilisez la même méthode que précédemment, et cherchez les mots « PHP Version »).

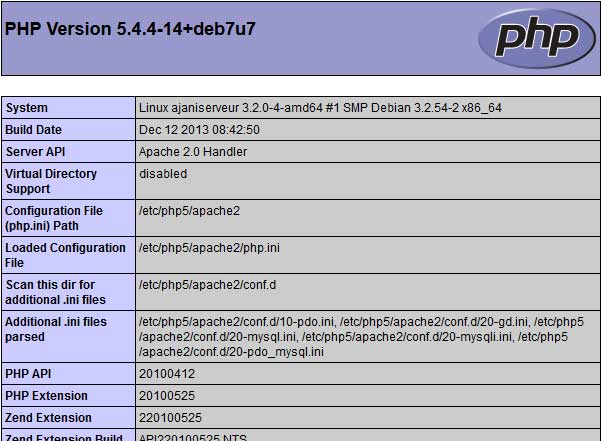


Tableau généré par la commande phpinfo sur une raspberry.

**Une base de données MySQL pour votre serveur**

**Un SGBD c’est quoi ? Pourquoi MySQL ?**

Maintenant que **nous avons mis en place PHP**, vous allez sans doute **vouloir stocker des informations** pour les utiliser dans vos sites. Pour cela, on utilise le plus souvent des bases de données.  
Nous allons donc mettre en place un SGBD (Système de Gestion de Bases de Données), à savoir MySQL.

**MySQL est un SGBD libre**, puissant, massivement utilisé (**environ 56 % de parts de marché des SGBD libres**). Là encore, MySQL est tellement un incontournable du développement, quelle qu’en soit la branche, que vous devez absolument [l’apprendre et le maitriser.](https://geni.us/rfllivremysql)

**Installer MySQL**

Pour ce faire, nous allons installer mysql-server **et** php-mysql (qui servira de lien entre php et mysql)

sudo apt install mysql-server php-mysql

**Vérifier que MySQL marche correctement**

Pour vérifier le fonctionnement de MySQL, cette fois **nous utiliserons uniquement la ligne de commande.** Pour cela, nous allons simplement nous connecter via la commande :

sudo mysql --user=root

Nous allons maintenant **supprimer l’utilisateur root 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.

Pour cela, une fois connecté à MySQL, lancez les commandes suivantes (remplacez password par le mot de passe de votre choix) :

DROP USER 'root'@'localhost';

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

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost';

Vous avez donc maintenant un **serveur web, relié à PHP et à MySQL**. Soit tout ce qu’il faut. **Vous pouvez donc vous arrêter ici.**

(Lors de vos prochaine connections, vous pourrez donc utilisez la commande mysql --user=root --password=votremotdepasse).

**PHPMyAdmin, gérer facilement ses bases de données**

Toutefois, vous pourriez vouloir **une interface un peu plus simple** pour administrer vos bases de données qu’une simple console MySQL. Pour cela, vous pouvez installer PHPMyAdmin.

**PHPMyAdmin, c’est quoi ?**

PHPMyAdmin est une application développée en PHP, et qui vise à fournir **une interface simplifiée pour MySQL.**  
Elle vous permet par exemple de voir rapidement et de façon lisible le contenu de votre base de données, ou de la manipuler **sans avoir besoin de faire vous-mêmes vos requêtes MySQL.**

**Installer PHPMyAdmin**

L’installation de PHPMyAdmin n’est **pas du tout obligatoire**. Nous ferons ici une installation **sans paramètres de sécurité particuliers.**

L’installation de PHPMyAdmin se fait très simplement, via le gestionnaire de paquets, en utilisant la commande suivante :

sudo apt install phpmyadmin

PHPMyAdmin vous posera plusieurs questions concernant ses réglages.

Comme nous avons déjà configurez la base de données, choisissez no à la question concernant l’utilisation de dbconfig-common. Choisissez d’utiliser PHPMyAdmin pour **un serveur Apache**. Pour le mot de passe root, il s’agit de **celui que vous aviez utilisé pour MySQL.**

**Vérifier l’installation de PHPMyAdmin**

Pour vérifier le bon fonctionnement de PHPMyAdmin, vous allez simple tenter d’y accéder, en utilisant l’adresse de votre Raspberry suivi de /phpmyadmin. Par exemple, en local ce sera « http://127.0.0.1/phpmyadmin ».

Si jamais vous avez une erreur, cela peut venir du fait que PHPMyAdmin se soit installé dans un autre dossier. Dans ce cas, essayez la commande

sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin

**Rendre votre raspberry pi accessible depuis le web**

Votre serveur web est donc prêt. Cependant, **vous ne pouvez sans doute pas y accéder depuis internet.** En effet, il faudrait pour cela que votre box (livebox, freebox, etc.) redirige les requêtes vers votre Raspberry, sur les bons ports, etc.  
Pour mettre ces redirections en place, et même obtenir une adresse URL, allez voir [le tuto sur le port fowarding et le dydns](http://raspbian-france.fr/mettre-en-ligne-serveur-web-raspbian-dydns-port-forwarding/).