

Sommaire :

Introduction.....	1
Partie I : Généralités.....	3
a) Recherche d'information sur les technologies :.....	3
a) Recherche d'information sur les tests de performances :	3
b) Limites et Modalités de nos tests :.....	4
Partie II : Les tests de performance.....	5
a) Les logiciels de tests :	5
b) Les Résultats :.....	6
c) Synthèse :	8
d) Conclusion :	10
ANNEXES : Nos installations	12
a) Machine n°1 : Apache 2 et PHP 7.0.....	13
b) Machine n°2 : Apache 2 et HHVM.....	15
c) Machine n°3 : Nginx et PHP 7.0.....	19
b) Machine n°4 : Nginx et HHVM.....	25

Introduction

Objectif

Notre objectif est de comparer les performances des deux serveurs web Apache 2 et Nginx sur un même système d'exploitation, Debian 8.5. Nous comparerons également les technologies PHP 7.0 et HHVM sur ces mêmes serveurs.

Contexte

Nous avons choisi les serveurs Apache et Nginx car nous avons toujours utilisé Apache et ne l'avons jamais comparé à un "concurrent". De plus, Apache est la solution Web la plus utilisée dans le monde, il est donc intéressant de voir si ce statut fait d'elle la meilleure ou si une autre solution peut potentiellement venir la détrôner.

Pour les technologies Web, nous utiliserons également PHP 7.0, car nous sommes habitués à PHP 5.6 depuis le début de nos études. Il serait intéressant de passer à la version la plus récente. La technologie HHVM semble avoir été conçue pour la virtualisation, et comme une amélioration de PHP. HHVM serait donc la technologie qui remplacerait PHP dans le futur. Il est donc utile de comparer leurs performances.

L'application utilisée pour les tests de performance, c'est-à-dire WordPress, a été choisie car nous l'avons déjà utilisé lors de nos stages en entreprise, et que sa mise en place est simple. Cette application permet également l'ajout de plugin qui nous permettra de réaliser nos tests de performance de manière graphique.

Indicateurs

- Citer leurs principales différences.
- Tester WordPress (requête/seconde).
- Monter les serveurs en charge.
- Tester les ressources mémoires et CPU qu'ils consomment.

Méthode

Nous allons installer les différents serveurs avec leurs configurations respectives, puis grâce à des plugins installés directement sur WordPress, nous pourrons graphiquement tester nos machines.

Mise en place

Les procédures de nos installations sont disponibles en annexes de ce document. Nous avons choisi les plugins « *Server Status* », « *WP Performance Tester* » et « *WP Server Stats* ». Leur installation est détaillée dans la partie II.

Résultats

Les résultats sont notés dans la partie II de ce document.

Partie I : Généralités

a) Recherche d'information sur les technologies :

- **Nginx (Engine X) :** Nginx est un logiciel libre développé en 2002 par Igor Sysoev. De plus en plus utilisé aujourd'hui, il s'agit d'un serveur HTTP qui se met petit à petit en concurrence avec Apache2.
- **Apache2 :** Apache2 est sûrement le serveur HTTP le plus connu et le plus utilisé dans la domaine du World Wide Web aujourd'hui, il a été développé en 1995 par la fondation Apache, le logiciel Apache perd de plus en plus de parts de marchés : entre 1999 à 2008 il y avait une hausse des parts de marchés, arrivant à 72,45% de part de marchés en 2008, puis ensuite une baisse des parts de marchés jusqu'à aujourd'hui. Apache2 fonctionne principalement sur les OS UNIX.
- **HipHop for PHP :** HipHop for PHP est un ensemble de logiciels créés par le réseau social Facebook, c'est un logiciel qui permet d'optimiser le développement, ainsi que l'exécution du code PHP, il a été publié en logiciel libre le 2 février 2010.
- **HHVM :** Plus tard, une nouvelle fonctionnalité a permis au logiciel libre HH for PHP d'avoir un peu plus d'impact en se lançant dans un projet de virtualisation qui sera appelé HHVM, afin d'optimiser encore plus les performances

a) Recherche d'information sur les tests de performances :

- **Test de charge :** il s'agit d'un test au cours duquel on va simuler un nombre d'utilisateurs virtuels prédéfinis. Il permet de mesurer le dimensionnement des serveurs, de la bande passante nécessaire sur le réseau, etc.
- **Test de performance :** il s'agit d'un test au cours duquel on va mesurer les performances de l'application soumise à une charge d'utilisateurs. Les informations recueillies concernent les temps de réponse utilisateurs, les temps de réponse réseau et les temps de traitement d'une requête sur le(s) serveur(s).
- **Test de robustesse, d'endurance, de fiabilité :** il s'agit de tests au cours desquels on va simuler une charge importante d'utilisateurs sur une durée relativement longue, pour voir si le système testé est capable de supporter une activité intense sur une longue période sans dégradations des performances et des ressources applicatives ou système.
- **Test de capacité, test de montée en charge :** il s'agit d'un test au cours duquel on va simuler un nombre d'utilisateurs sans cesse croissant de manière à déterminer quelle charge limite le système est capable de supporter.

- **Test aux limites** : il s'agit d'un test au cours duquel on va simuler en général une activité bien supérieure à l'activité normale, pour voir comment le système réagit aux limites du modèle d'usage de l'application.

b) Limites et Modalités de nos tests :

Système	Debian 8.5		Debian 8.5	
Serveur HTTP	Apache 2		Nginx	
Techno Web	PHP 7.0	HHVM	PHP 7.0	HHVM
CMS	Wordpress		Wordpress	

Pour comparer toutes ces technologies, nous allons utiliser les critères suivants :

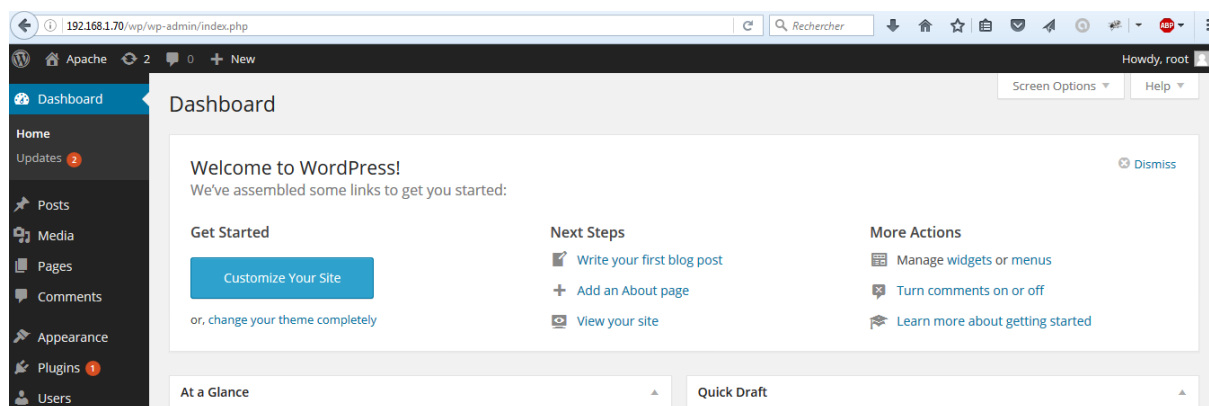
- Pourcentage du CPU utilisé.
- Utilisation de l'espace disque par la base de données.
- Utilisation de la mémoire vive par PHP/HHVM.
- Nombres maximum de connexions possibles.
- Le temps d'accès à la base de données.
- Requêtes par secondes exécutées par le WordPress.

Nous serons sûrement limités par le temps, car nous ne pourrions pas tester les services sur plusieurs heures ou sur plusieurs jours par exemple. C'est pourquoi nous avons exclu les indicateurs tels que la durée de bon fonctionnement du service lors d'une période de forte affluence, par exemple.

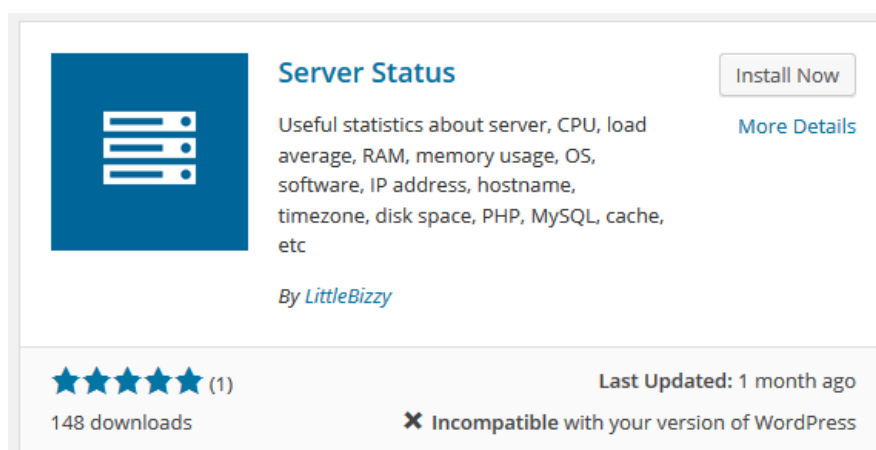
Partie II : Les tests de performance

a) Les logiciels de tests :

Pour réaliser nos tests de performance, nous allons trois plugins directement via l'outil plugin de WordPress. Pour cela il faut, sur le panneau d'administration de nos sites WordPress, accéder à l'onglet « **Plugins** » :



Puis, dans l'onglet « **Add New** », nous recherchons les plugins « **Server Status** », « **WP Performance Tester** » et « **WP Server Stats** » qui vont nous permettre d'évaluer la vitesse d'exécution de nos installations, de définir combien de mémoire ils utilisent, etc. Nous cliquons donc sur « **Install Now** » pour chaque plugin :



Puis, après installation, cliquez sur « **Activate Plugin** » :

Installing Plugin: Server Status 1.2

Downloading install package from <https://downloads.wordpress.org/plugin/server-status-littlebizzy.1.2.zip...>

Unpacking the package...

Installing the plugin...

Successfully installed the plugin **Server Status 1.2**.

[Activate Plugin](#) | [Return to Plugin Installer](#)

Une fois ceci fait pour les trois plugins, sur les quatre serveurs, il ne reste plus qu'à lire les résultats proposés.

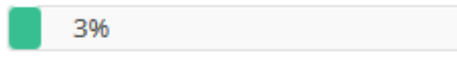
b) Les Résultats :

Après avoir lancé les tests, nous obtenons des résultats comme celui-ci :

Server Performance Benchmarks	
Test	Execution Time (seconds)
Math	0.234
String Manipulation	0.355
Loops	0.019
Conditionals	0.041
Mysql Connect	0.000
Mysql Select Database	0.000
Mysql Query Version	0.000
Mysql Query Benchmark	9.004
Total Time (seconds)	9.653

WordPress Performance Benchmark	
Execution Time (seconds)	Queries Per Second
138.440	7.2233458537995

Real Time CPU Load:



Database Software : (Debian)

Database Version : 5.5.53

Maximum No. of Connections : 151

Maximum Packet Size : 16 MB

Database Disk Usage : 1 MB

Index Disk Usage : 384 KB

PHP Version : 7.0.13-1-dotdeb+8.1

PHP Max Upload Size : 2 MB

PHP Max Post Size : 8 MB

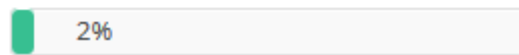
PHP Max Execution Time : 120 sec

PHP Safe Mode : Off

PHP Short Tag : Off

PHP Memory Limit : 256 MB

Real Time PHP Memory Usage : 2.02 MB



Nous avons donc capturé tous les résultats de nos installations, et voici les résultats.

Apache 2 et PHP 7.0 :

Pourcentage du CPU utilisé : **3%**

Utilisation de l'espace disque par la base de données : **1 MB**

Utilisation de la mémoire vive par PHP/HHVM : **2,02 MB sur 256 MB**

Nombres maximum de connexions possibles : **151 connexions**

Le temps d'accès à la base de données : **9,653 secondes**

Requêtes par secondes exécutées par le WordPress : **7,22 requêtes par secondes**

Apache 2 et HHVM :

Pourcentage du CPU utilisé : **2%**

Utilisation de l'espace disque par la base de données : **1 MB**

Utilisation de la mémoire vive par PHP/HHVM : **1,56 MB sur 256 MB**

Nombres maximum de connexions possibles : **151 connexions**

Le temps d'accès à la base de données : **7,542 secondes**

Requêtes par secondes exécutées par le WordPress : **9,81 requêtes par secondes**

Nginx et PHP 7.0 :

Pourcentage du CPU utilisé : **1%**

Utilisation de l'espace disque par la base de données : **1 MB**

Utilisation de la mémoire vive par PHP/HHVM : **2,01 MB sur 256 MB**

Nombres maximum de connexions possibles : **243 connexions**

Le temps d'accès à la base de données : **7,128 secondes**

Requêtes par secondes exécutées par le WordPress : **14,10 requêtes par secondes**

Nginx et HHVM :

Pourcentage du CPU utilisé : **1%**

Utilisation de l'espace disque par la base de données : **1 MB**

Utilisation de la mémoire vive par PHP/HHVM : **0,98 MB sur 256 MB**

Nombres maximum de connexions possibles : **243 connexions**

Le temps d'accès à la base de données : **5,942 secondes**

Requêtes par secondes exécutées par le WordPress : **21,79 requêtes par secondes**

c) Synthèse :

Nous avons répertorié tous les résultats dans un tableau, en appliquant un code couleur en fonction de leurs performances par rapport à leurs « concurrents ». Ainsi, nous pouvons classer les quatre configurations entre elle d'un rapide coup d'œil :

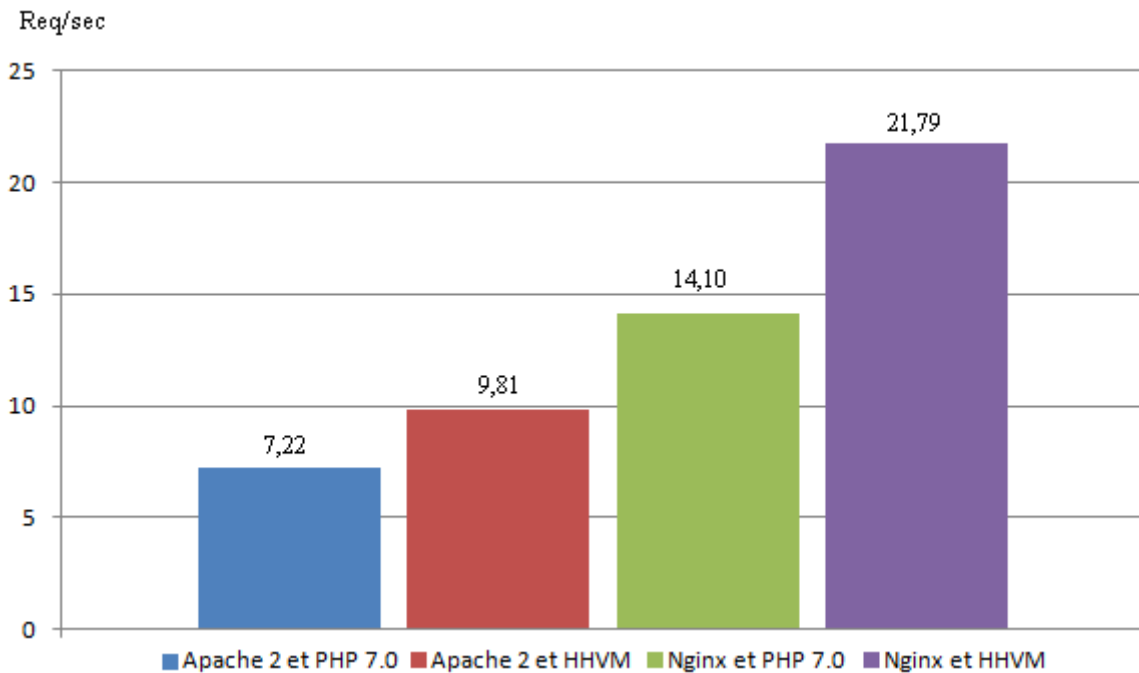
Vert : Meilleure performance

Jaune : Performance moyenne

Rouge : Pire performance

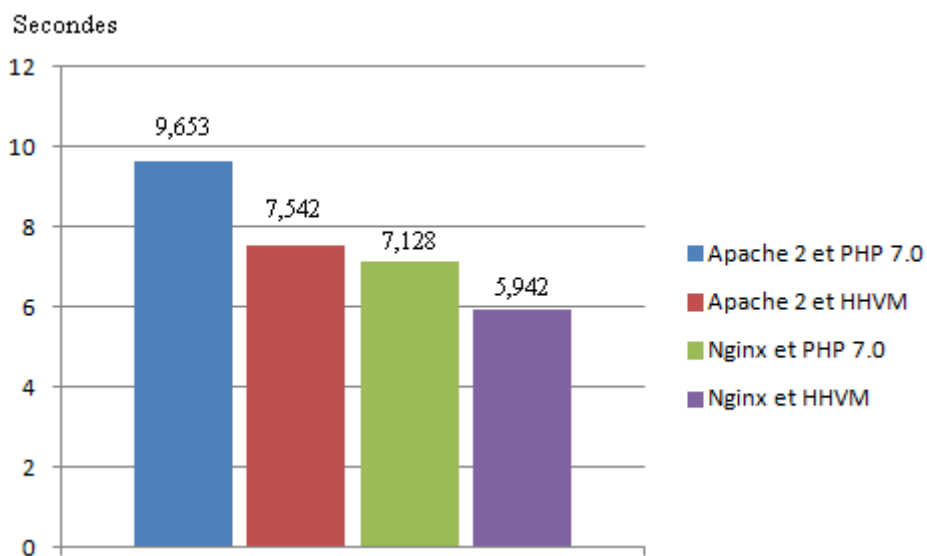
Configuration	Apache 2 et PHP 7.0	Apache 2 et HHVM	Nginx et PHP 7.0	Nginx et HHVM
CPU utilisé	3 %	2 %	1 %	1 %
Espace disque utilisée par la BDD	1 MB	1 MB	1 MB	1 MB
Mémoire vive utilisée par PHP/HHVM	2,02 MB	1,56 MB	2,01 MB	0,98 MB
Nombres de connexions maximum	151 connexions	151 connexions	243 connexions	243 connexions
Temps d'accès à la BDD	9,653 secondes	7,542 secondes	7,128 secondes	5,942 secondes
Requêtes par secondes exécutées par le WordPress	7,22 req/secs	9,81 req/secs	14,10 req/secs	21,79 req/secs

Requêtes par secondes exécutées par le WordPress



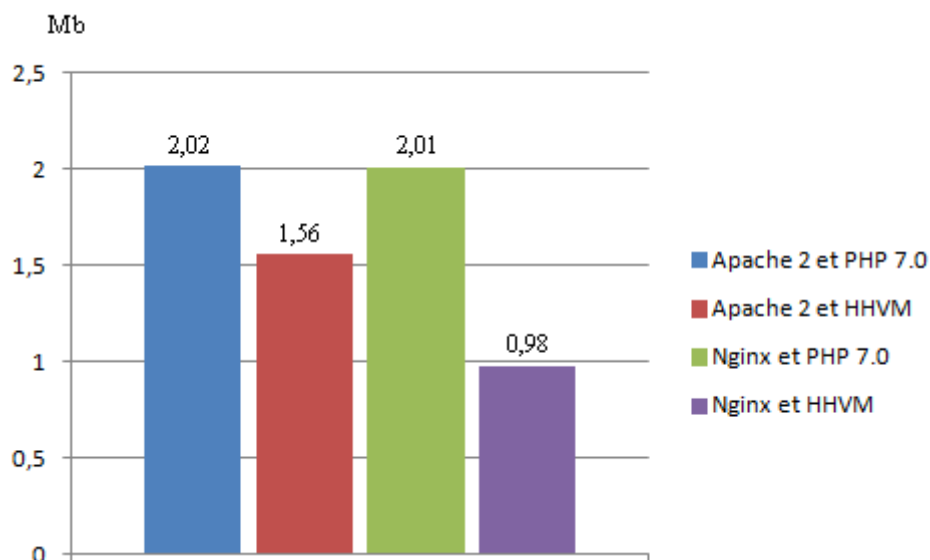
Plus la valeur est haute, plus la solution est performante.

Temps d'accès à la base de données en secondes



Plus la valeur est basse, plus la solution est performante.

Mémoire vive utilisée par PHP/HHVM



Plus la valeur est basse, plus la solution est performante.

d) Conclusion :

Nous pouvons conclure que :

- Nginx est plus économique qu'Apache en capacité de CPU utilisée.
- HHVM est également moins gourmand que PHP pour l'utilisation de la mémoire vive.
- La connexion à la base de données lors du lancement de WordPress est plus rapide sur Nginx que sur Apache.
- La connexion à la base de données lors du lancement de WordPress est plus rapide avec HHVM qu'avec PHP.
- Nginx supporte plus de connexions simultanées qu'Apache.
- WordPress peut exécuter plus de requêtes par secondes sur Nginx que sur Apache.
- WordPress peut exécuter plus de requêtes par secondes avec HHVM qu'avec PHP.

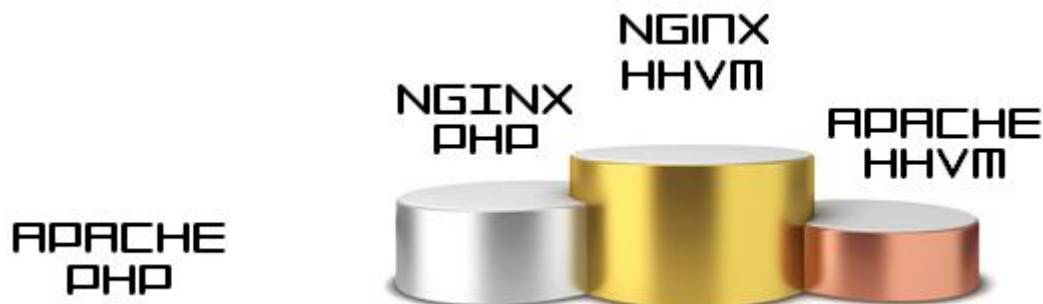
Malgré que la solution Web la plus utilisée dans le monde soit Apache, il apparaît dans nos tests que Nginx est plus performant. En effet, il accède plus rapidement au serveur de base de données et permet donc aux sites Web qu'il héberge d'effectuer un plus grand nombre de requêtes par secondes.

Quant à la différence entre PHP et HHVM, il est clair que HHVM est plus économique. Il utilise moins de ressources que PHP, notamment de mémoire vive, pour fonctionner. Ce résultat est normal, sachant que HHVM a été conçu spécialement pour la virtualisation, et qu'il a été construit après des améliorations successives de la technologie PHP par Facebook.

Pour conclure, le combo Nginx et HHVM est plus économique en ressources ainsi que plus performant que les technologies Apache 2 et PHP (5 ou 7) que nous utilisons tout au long de l'année en cours. Néanmoins, pour les besoins que nous avons, il est normal d'utiliser les solutions les plus répandues, car ce sont elles que nous serons susceptibles de rencontrer la plupart du temps.

Les seules limites de Nginx et de HHVM peuvent se trouver dans la complexité de leurs installations, un peu moins simple qu'Apache pour Nginx, et dans des problèmes de compatibilités qui peuvent survenir, pour HHVM notamment, étant donné que cette technologie est récente.

Finalement, nos tests de performance ont permis de réaliser le classement suivant :



ANNEXES : Nos installations

Pré-requis :

Pour préparer le téléchargement de tous les paquets requis et les mises à jour, il faut ajouter sur toutes nos machines les dépôts suivant dans */etc/apt/sources.list* :

```
deb http://ftp.fr.debian.org/debian/ jessie main  
deb http://ftp.fr.debian.org/debian/ jessie-updates main
```

```
GNU nano 2.2.6      Fichier : /etc/apt/sources.list
#
# deb cdrom:[Debian GNU/Linux 8.1.0 _Jessie_ - Official amd64 CD Binary-1 20150603]
deb cdrom:[Debian GNU/Linux 8.1.0 _Jessie_ - Official amd64 CD Binary-1 20150603]
deb http://ftp.fr.debian.org/debian/ jessie main
deb-src http://ftp.fr.debian.org/debian/ jessie main
deb http://security.debian.org/ jessie/updates main
deb-src http://security.debian.org/ jessie/updates main
# jessie-updates, previously known as 'volatile'
deb http://ftp.fr.debian.org/debian/ jessie-updates main
deb-src http://ftp.fr.debian.org/debian/ jessie-updates main

[ Lecture de 15 lignes ]
^G Aide      ^O Écrire    ^R Lire fich.^Y Page préc.^K Couper     ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher  ^V Page suiv.^U Coller    ^T Orthograp.
```

Et effectuer un *apt-get update*.

a) Machine n°1 : Apache 2 et PHP 7.0

Installation d'Apache 2 :

Pour installer Apache 2, il suffit d'utiliser la commande suivante : ***apt-get install apache2***
Et pour tester si le service est fonctionnel : ***service apache2 status***

Installation de PHP 7.0 :

On ajoute d'abord le dépôt suivant dans ***/etc/apt/sources.list*** :

deb http://packages.dotdeb.org jessie all

Puis, on utilise les commandes :

apt-get install php7.0
apt-get install php7.0-fpm

Il suffit ensuite tester le tout avec un fichier info.php.

Installation de WordPress :

Pour installer WordPress sur une machine Apache, nous avons besoin d'installer les applications suivantes : ***apt-get install wordpress curl apache2 mysql-server php-mysql***

Nous devons ensuite créer un site dans ***/etc/apache2/sites-available*** nommé ***wp.conf*** :

```
Alias /wp/wp-content /var/lib/wordpress/wp-content
Alias /wp /usr/share/wordpress

<Directory /usr/share/wordpress>
    Options FollowSymLinks
    AllowOverride Limit Options FileInfo
    DirectoryIndex index.php
    Require all granted
</Directory>

<Directory /var/lib/wordpress/wp-content>
    Options FollowSymLinks
    Require all granted
</Directory>
```

Puis l'activer avec ***a2ensite wp*** et redémarrer Apache. Nous devons ensuite créer un script pour permettre à notre site WordPress de se connecter sur une base de données. Nous le nommons ***config-NOTRE_IP.php*** :

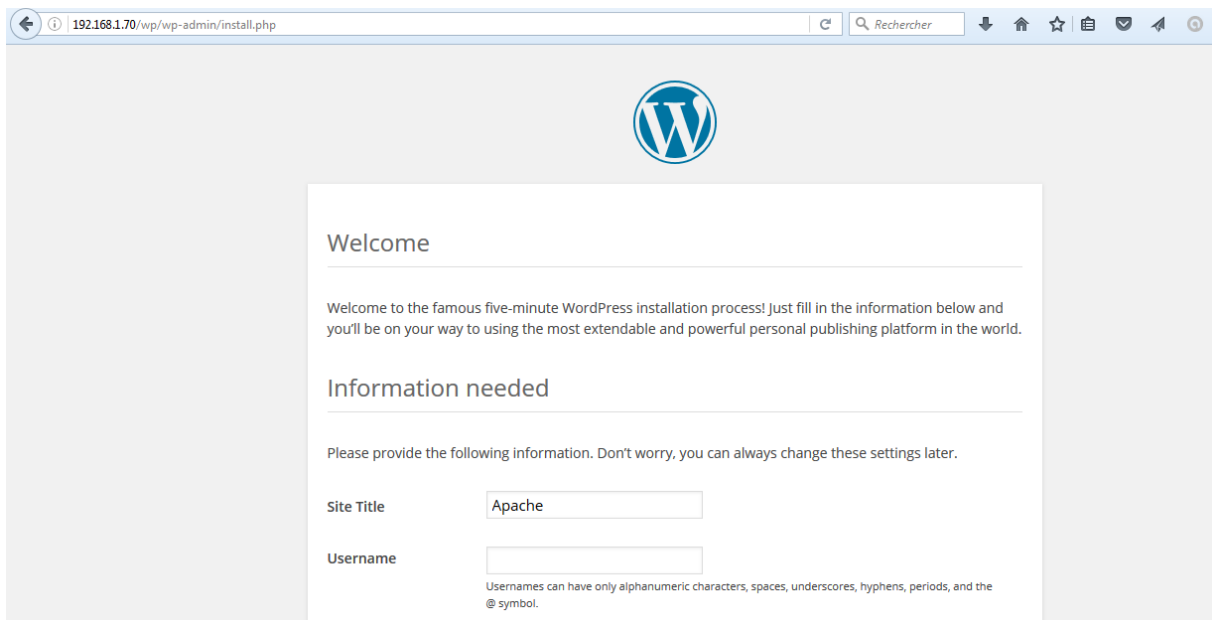
```
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'password');
define('DB_HOST', 'localhost');
define('WP_CONTENT_DIR', '/var/lib/wordpress/wp-content');
?>_
```

Nous créons ensuite le script qui va permettre la création de la base de données, **wp.sql** :

```
CREATE DATABASE wordpress;
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER
ON wordpress.*
TO wordpress@localhost
IDENTIFIED BY 'password';
FLUSH PRIVILEGES;_
```

Et nous l'exécutons avec un **cat ~/wp.sql | mysql --defaults-extra-file=/etc/mysql/debian.cnf**.

Nous pouvons maintenant accéder à l'installation graphique de WordPress en entrant notre adresse IP sur un navigateur suivie de **/wp** :



Il ne reste plus qu'à suivre l'installation guidée.

b) Machine n°2 : Apache 2 et HHVM

Installation d'Apache 2 :

Pour installer Apache 2, il suffit d'utiliser la commande suivante : ***apt-get install apache2***
Et pour tester si le service est fonctionnel : ***service apache2 status***

Installation de HHVM :

On ajoute d'abord le dépôt suivant dans ***/etc/apt/sources.list*** :

deb http://dl.hhvm.com/debian jessie main

Puis, on utilise la commande : ***apt-get install hhvm***

!\ ATTENTION : L'installation de HHVM est très longue. Il ne faut pas oublier de démarrer HHVM.

Après avoir installé Apache et HHVM indépendamment, nous devons exécuter le script suivant pour lancer la configuration automatique des deux services :

/usr/share/hhvm/install_fastcgi.sh

Ensuite, la commande suivante pour permettre à HHVM de fonctionner avec le langage PHP :

/usr/bin/update-alternatives --install /usr/bin/php php /usr/bin/hhvm 60

Et enfin, il faut modifier le fichier ***/etc/apache2/mods-available/hhvm_proxy_fcgi.conf*** comme ci-dessous afin de localiser nos pages HTML :

```
GNU nano 2.2.6      Fichier : hhvm_proxy_fcgi.conf
ProxyPassMatch ^/(.+\.hh|php)(/.*?)$ fcgi://127.0.0.1:9000/var/www/html/$1
```

Nous pouvons maintenant tester la configuration avec un fichier *info.php* :

Version	
Version	3.15.0
Version ID	31500
Debug	
Compiler ID	tags/HHVM-3.15.0-0-g92a682ebaa3c85b84857852d8621f528607fe27d
Repo Schema	225d4323575bbc8a498dc809a1c41354f6bca830
PHP Version	5.6.99-hhvm
Zend Version	2.4.99
uname	Linux apachehhvm 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt25-2+deb8u3 (2016-07-02) x86_64

Installation de WordPress :

Pour installer WordPress sur une machine Apache, nous avons besoin d'installer les applications suivantes : ***apt-get install wordpress curl apache2 mysql-server php-mysql***

Nous devons ensuite créer un site dans ***/etc/apache2/sites-available*** nommé ***wp.conf*** :


```
Alias /wp/wp-content /var/lib/wordpress/wp-content
Alias /wp /usr/share/wordpress

<Directory /usr/share/wordpress>
    Options FollowSymLinks
    AllowOverride Limit Options FileInfo
    DirectoryIndex index.php
    Require all granted
</Directory>

<Directory /var/lib/wordpress/wp-content>
    Options FollowSymLinks
    Require all granted
</Directory>
```

Puis l'activer avec **a2ensite wp** et redémarrer Apache. Nous devons ensuite créer un script pour permettre à notre site WordPress de se connecter sur une base de données. Nous le nommons **config-NOTRE_IP.php** :

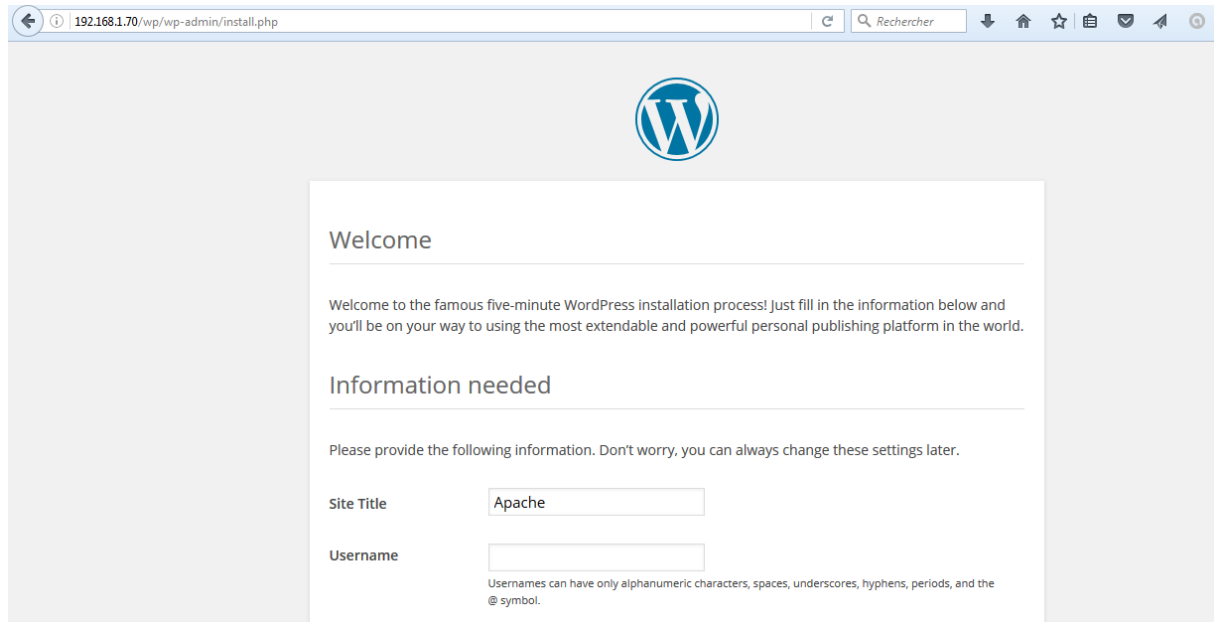
```
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'password');
define('DB_HOST', 'localhost');
define('WP_CONTENT_DIR', '/var/lib/wordpress/wp-content');
?>_
```

Nous créons ensuite le script qui va permettre la création de la base de données, **wp.sql** :

```
CREATE DATABASE wordpress;
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER
ON wordpress.*
TO wordpress@localhost
IDENTIFIED BY 'password';
FLUSH PRIVILEGES;_
```

Et nous l'exécutons avec un **cat ~/wp.sql | mysql --defaults-extra-file=/etc/mysql/debian.cnf**.

Nous pouvons maintenant accéder à l'installation graphique de WordPress en entrant notre adresse IP sur un navigateur suivie de **/wp** :



Il ne reste plus qu'à suivre l'installation guidée.

c) Machine n°3 : Nginx et PHP 7.0Installation de Nginx :

Pour installer Nginx, il suffit d'utiliser la commande suivante : **apt-get install nginx**
Et pour tester si le service est fonctionnel : **service nginx status**

Installation de PHP 7.0 :

On ajoute d'abord le dépôt suivant dans **/etc/apt/sources.list** :

deb http://packages.dotdeb.org jessie all

Puis, on utilise les commandes :

apt-get install php7.0
apt-get install php7.0-fpm

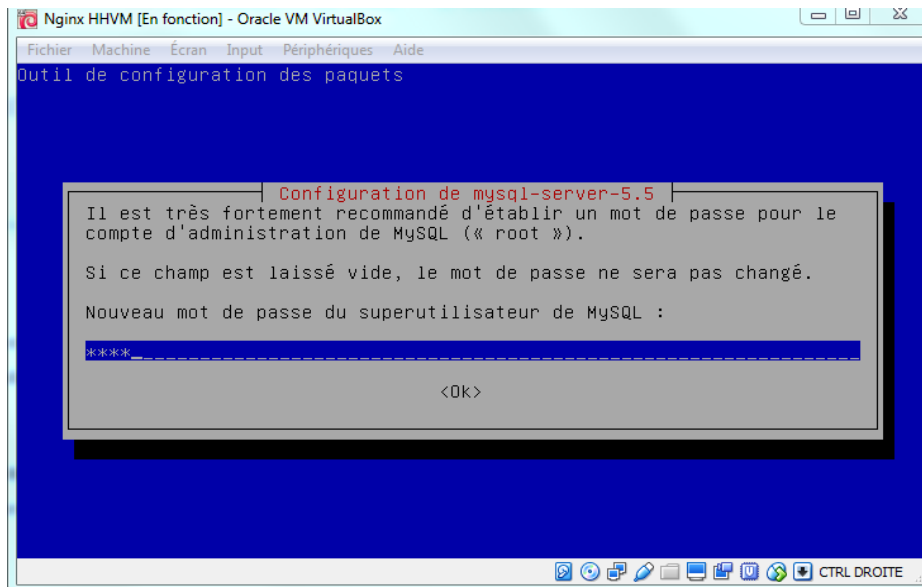
/\! ATTENTION : Cette commande installe également Apache 2 sur la machine. Si le service Web utilisé est Nginx, il faut penser à désactiver Apache 2 avec un **service apache2 stop**.

Une fois l'installation faite, vous allez modifier un fichier de configuration afin de pouvoir afficher correctement les pages contenant du PHP, présent dans le répertoire **/etc/nginx/sites-available** et modifiez le fichier de configuration **default** de façon à avoir ces lignes ci-dessous :

```
# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
#
location ~ /\.php$ {
    try_files $uri =404;
    fastcgi_index info.php;
    fastcgi_pass php7.0-fpm-sock;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include /etc/nginx/fastcgi_params;
    include snippets/fastcgi-php.conf;
#
```

Le socket que vous allez ajouter devra aussi apparaître dans un fichier appelé **php7.0-fpm.conf** présent dans le répertoire **/etc/nginx/conf.d** :

```
GNU nano 2.2.6      Fichier : php7.0-fpm.conf
upstream php7.0-fpm-sock {
    server unix:/var/run/php/php7.0-fpm.sock;
}
```

Nous avons choisi le mot de passe **root**.

Une fois fait, vous allez créer une base de données dédiée à WordPress :

```
mysql> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.00 sec)
```

Créez ensuite un utilisateur :

```
mysql> CREATE USER wordpressuser@localhost IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)
```

Donnez maintenant les privilèges pour la base de donnée à l'utilisateur :

```
mysql> GRANT ALL PRIVILEGES ON wordpress.* TO wordpressuser@localhost;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

Procédez maintenant à l'installation de WordPress, en le téléchargeant :

```
root@nginxhhvm:~# wget http://wordpress.org/latest.tar.gz
```

Décompressez le dans le dossier /var/www/html :

```
root@nginxhhvm:~# tar xzvf latest.tar.gz
```

Nous allons maintenant configurer WordPress. Allez dans votre dossier WordPress et copier le fichier suivant que vous allez configurer :

```
root@nginxhhvm:~/wordpress# cp wp-config-sample.php wp-config.php
```

Une fois fait, vous allez changer les 3 premières lignes concernant la base de données, l'utilisateur ainsi que le mot de passe (en jaune) dans le fichier *wp-config-sample.php* :

```
// ** MySQL settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define('DB_NAME', 'wordpress');  
  
/** MySQL database username */  
define('DB_USER', 'wordpressuser');  
  
/** MySQL database password */  
define('DB_PASSWORD', 'password');  
  
/** MySQL hostname */  
define('DB_HOST', 'localhost');
```

Vous allez maintenant configurer WordPress dans votre dossier Nginx, les fichiers devront être déplacés de façon à ce que ce soit Nginx qui les récupères :

Vous allez diriger tout les fichiers WordPress dans le */var/www/html* :

```
root@nginxhhvm:/var/www/html# rsync -avP wordpress/ /var/www/html
```

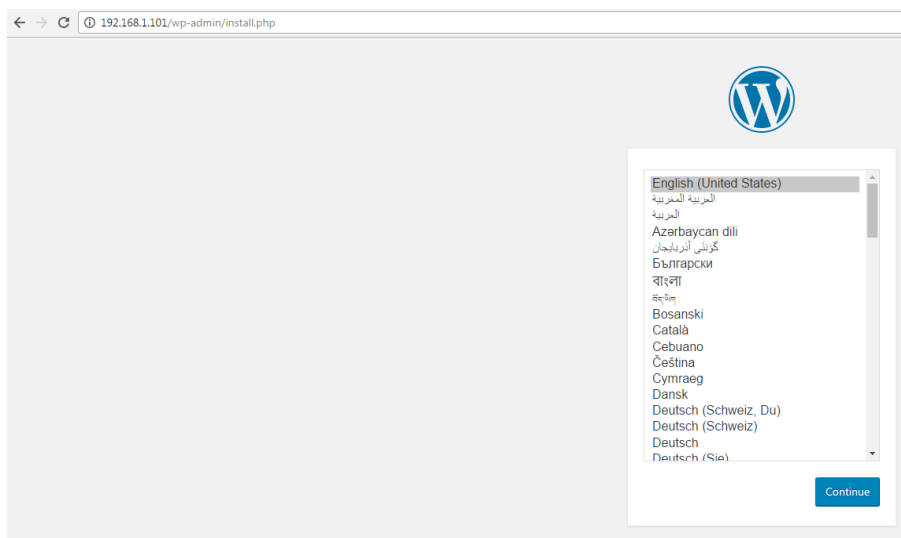
Nous allons maintenant accorder des droits :

```
root@nginxhhvm:/var/www/html# chown -R :www-data /var/www/html/*  
root@nginxhhvm:/var/www/html#
```

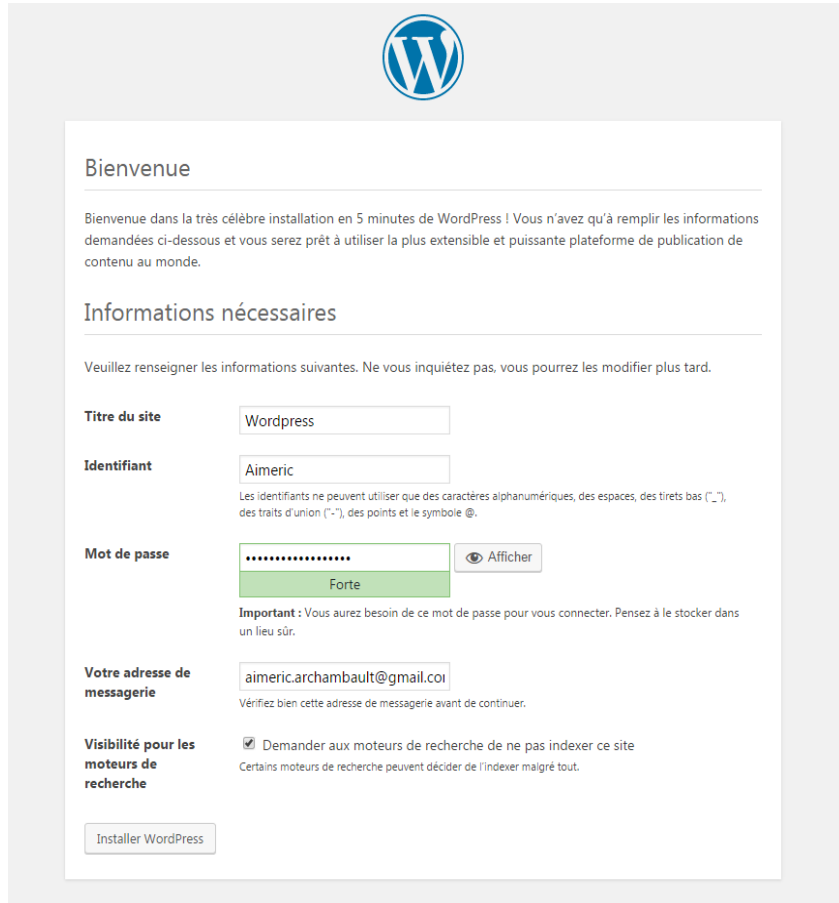
Pour pouvoir accéder à WordPress via un navigateur, nous allons modifier le fichier *default* présent dans le répertoire */etc/nginx/sites-available* :

```
root /var/www/html;  
  
# Add index.php to the list if you are using PHP  
index index.php index.htm index.nginx-debian.html;  
  
server_name 192.168.1.101;  
  
location / {  
    # First attempt to serve request as file, then  
    # as directory, then fall back to displaying a 404.  
    # try_early_files $uri $uri/ =404;  
    try_files $uri $uri/ /index.php?q=$uri&$args;  
}
```

Voilà les lignes qui ont été modifiées (en jaune). Une fois fait, vous allez mettre dans votre navigateur de recherche ce que vous avez insérer sur la ligne **server_name** qui est notre adresse IP :



Nous arrivons sur le fichier d'installation :




The image shows the WordPress installation 'Welcome' screen. At the top center is the WordPress logo. Below it, the heading 'Bienvenue' is followed by a paragraph: 'Bienvenue dans la très célèbre installation en 5 minutes de WordPress ! Vous n'avez qu'à remplir les informations demandées ci-dessous et vous serez prêt à utiliser la plus extensible et puissante plateforme de publication de contenu au monde.' The next section is 'Informations nécessaires', with a sub-heading 'Veillez renseigner les informations suivantes. Ne vous inquiétez pas, vous pourrez les modifier plus tard.' The form contains several fields: 'Titre du site' with the value 'Wordpress'; 'Identifiant' with the value 'Aimeric', accompanied by a note: 'Les identifiants ne peuvent utiliser que des caractères alphanumériques, des espaces, des tirets bas ("_"), des traits d'union ("-"), des points et le symbole @.'; 'Mot de passe' with a masked password '.....', a strength indicator 'Forte' in a green box, and an 'Afficher' button; 'Votre adresse de messagerie' with the value 'aimeric.archambault@gmail.coi' and a note: 'Vérifiez bien cette adresse de messagerie avant de continuer.'; and a checkbox for 'Visibilité pour les moteurs de recherche' which is checked, with the text 'Demander aux moteurs de recherche de ne pas indexer ce site' and a note: 'Certains moteurs de recherche peuvent décider de l'indexer malgré tout.' At the bottom left is an 'Installer WordPress' button.

Wordpress
Un site utilisant WordPress

Bonjour tout le monde !

19 octobre 2016
Un commentaire

Bienvenue dans WordPress. Ceci est votre premier article. Modifiez-le ou supprimez-le, puis lancez-vous !

Recherche... 

ARTICLES RÉCENTS

- [Bonjour tout le monde !](#)

COMMENTAIRES RÉCENTS

- Un commentateur WordPress dans [Bonjour tout le monde !](#)

ARCHIVES

- [octobre 2016](#)

CATÉGORIES

- [Non classé](#)
-

b) Machine n°4 : Nginx et HHVM

Installation de Nginx :

Pour installer Nginx, il suffit d'utiliser la commande suivante : ***apt-get install nginx***
Et pour tester si le service est fonctionnel : ***service nginx status***

Installation de HHVM :

On ajoute d'abord le dépôt suivant dans ***/etc/apt/sources.list*** :

deb http://dl.hhvm.com/debian jessie main

Puis, on utilise la commande : ***apt-get install hhvm***

!/ ATTENTION : L'installation de HHVM est très longue. Il ne faut pas oublier de démarrer HHVM.

Une fois l'installation faite, vous allez modifier un fichier de configuration afin de pouvoir afficher correctement les pages contenant du PHP, présent dans le répertoire ***/etc/nginx/sites-available*** et modifiez le fichier de configuration ***default*** de façon à avoir ces lignes ci-dessous:

```
# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
#
location ~ \.(hh|php)$ {
    fastcgi_keep_conn on;
    fastcgi_pass    127.0.0.1:9000;
    fastcgi_index   info.php;
    fastcgi_param   SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include         fastcgi_params;
}
```

Maintenant, essayez d'afficher votre page PHP que vous avez configuré dans le répertoire ***/var/www/html*** et que vous avez nommé ***info.php*** :

192.168.1.101/info.php

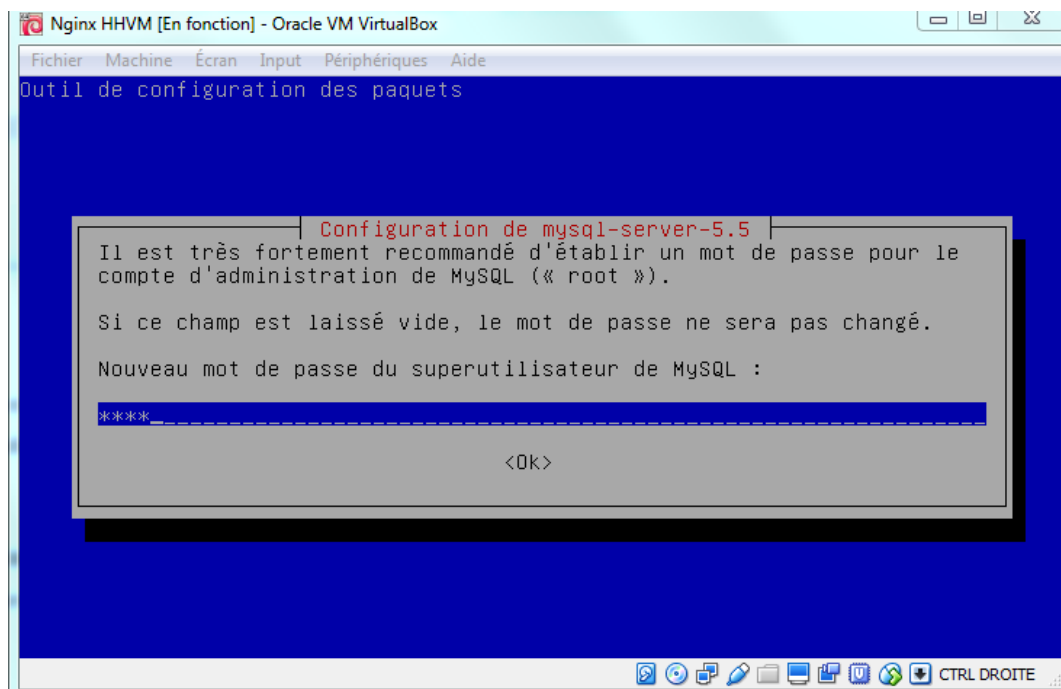
HHVM Version 3.15.0

Version

Version	3.15.0
Version ID	31500
Debug	
Compiler ID	tags/HHVM-3.15.0-0-g92a682ebaa3c85b84857852d8621f528607fe27d
Repo Schema	225d4323575bbc8a498dc809a1c41354f6bc830
PHP Version	5.6.99-hhvm
Zend Version	2.4.99
osname	Linux nginxhhvm 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt25-2+deb8u3 (2016-07-02) x86_64

Installation de WordPress :

Pour pouvoir utiliser WordPress, nous devons préalablement avoir un serveur de bases de données :
apt install mysql-server



Nous avons choisi le mot de passe **root**.

Une fois fait, vous allez créer une base de données dédiée à WordPress :

```
mysql> CREATE DATABASE wordpress;  
Query OK, 1 row affected (0.00 sec)
```

Créez ensuite un utilisateur :

```
mysql> CREATE USER wordpressuser@localhost IDENTIFIED BY 'password';  
Query OK, 0 rows affected (0.00 sec)
```

Donnez maintenant les privilèges pour la base de donnée à l'utilisateur :

```
mysql> GRANT ALL PRIVILEGES ON wordpress.* TO wordpressuser@localhost;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.00 sec)
```

Procédez maintenant à l'installation de WordPress, en le téléchargeant :

```
root@nginxhhvm:~# wget http://wordpress.org/latest.tar.gz
```

Décompressez le dans le dossier /var/www/html :

```
root@nginxhhvm:~# tar xzvf latest.tar.gz
```

Nous allons maintenant configurer WordPress. Allez dans votre dossier WordPress et copier le fichier suivant que vous allez configurer :

```
root@nginxhhvm:~/wordpress# cp wp-config-sample.php wp-config.php
```

Une fois fait, vous allez changer les 3 premières lignes concernant la base de données, l'utilisateur ainsi que le mot de passe (en jaune) dans le fichier **wp-config-sample.php** :

```
// ** MySQL settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define('DB_NAME', 'wordpress');  
  
/** MySQL database username */  
define('DB_USER', 'wordpressuser');  
  
/** MySQL database password */  
define('DB_PASSWORD', 'password');  
  
/** MySQL hostname */  
define('DB_HOST', 'localhost');
```

Vous allez maintenant configurer WordPress dans votre dossier Nginx, les fichiers devront être déplacés de façon à ce que ce soit Nginx qui les récupères :

Vous allez diriger tout les fichiers WordPress dans le `/var/www/html` :

```
root@enginxhvm:/var/www/html# rsync -avP wordpress/ /var/www/html
```

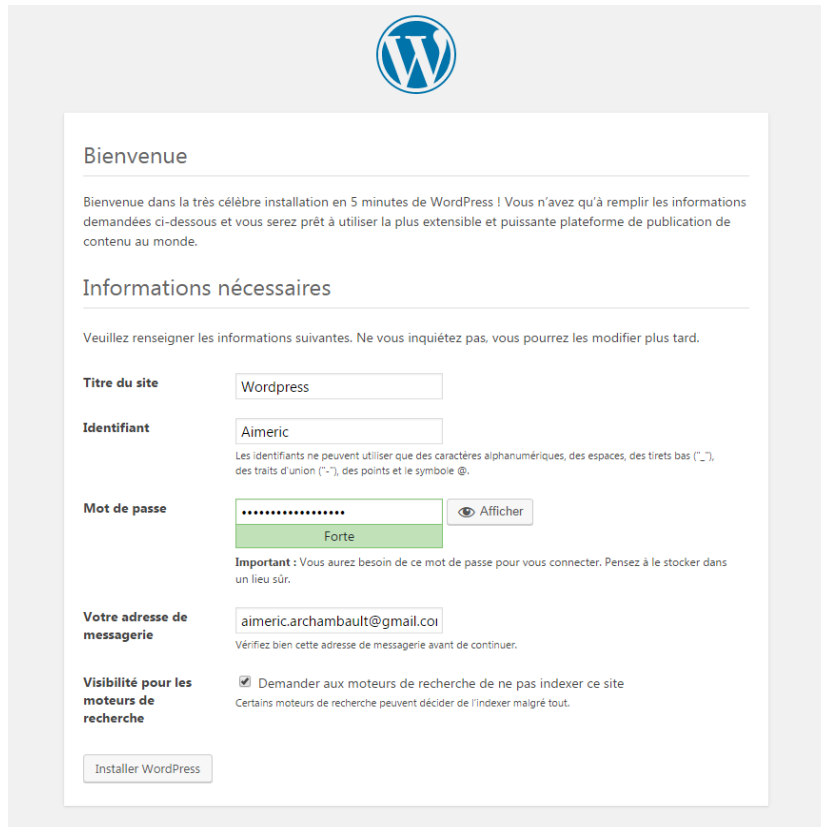
Nous allons maintenant accorder des droits :

```
root@enginxhvm:/var/www/html# chown -R :www-data /var/www/html/*  
root@enginxhvm:/var/www/html#
```

Pour pouvoir accéder à WordPress via un navigateur, nous allons modifier le fichier **default** présent dans le répertoire `/etc/nginx/sites-available` :

```
root /var/www/html;  
  
# Add index.php to the list if you are using PHP  
index index.php index.htm index.nginx-debian.html;  
  
server_name 192.168.1.101;  
  
location / {  
    # First attempt to serve request as file, then  
    # as directory, then fall back to displaying a 404.  
    # try_files $uri $uri/ =404;  
    try_files $uri $uri/ /index.php?q=$uri&$args;  
}
```

Voilà les lignes qui ont été modifiées (en jaune). Une fois fait, vous allez mettre dans votre navigateur de recherche ce que vous avez insérer sur la ligne **server_name** qui est notre adresse IP, et nous arrivons sur le fichier d'installation :



The image shows the WordPress installation 'Welcome' screen. At the top center is the WordPress logo. Below it, the heading 'Bienvenue' is followed by a paragraph: 'Bienvenue dans la très célèbre installation en 5 minutes de WordPress ! Vous n'avez qu'à remplir les informations demandées ci-dessous et vous serez prêt à utiliser la plus extensible et puissante plateforme de publication de contenu au monde.' Below this is the section 'Informations nécessaires'. A sub-heading reads: 'Veuillez renseigner les informations suivantes. Ne vous inquiétez pas, vous pourrez les modifier plus tard.' The form contains several fields: 'Titre du site' with the value 'Wordpress'; 'Identifiant' with the value 'Aimeric', accompanied by a note: 'Les identifiants ne peuvent utiliser que des caractères alphanumériques, des espaces, des tirets bas ("_"), des traits d'union ("-"), des points et le symbole @.'; 'Mot de passe' with a masked password '.....', a strength indicator 'Forte', and an 'Afficher' button; 'Votre adresse de messagerie' with the value 'aimerichambault@gmail.com' and a note: 'Vérifiez bien cette adresse de messagerie avant de continuer.'; and a checkbox 'Visibilité pour les moteurs de recherche' which is checked, with the text 'Demander aux moteurs de recherche de ne pas indexer ce site' and a note: 'Certains moteurs de recherche peuvent décider de l'indexer malgré tout.' At the bottom left of the form is an 'Installer WordPress' button.

Wordpress
Un site utilisant WordPress

Bonjour tout le monde !

19 octobre 2016
Un commentaire

Bienvenue dans WordPress. Ceci est votre premier article. Modifiez-le ou supprimez-le, puis lancez-vous !

Recherche...

ARTICLES RÉCENTS

- [Bonjour tout le monde !](#)

COMMENTAIRES RÉCENTS

- Un commentateur WordPress dans [Bonjour tout le monde !](#)

ARCHIVES

- [octobre 2016](#)

CATÉGORIES

- [Non classé](#)
-