

Collection Nuxwin
Les Tutoriaux
UBUNTU
DAPPER DRAKE

Installation d'une solution



Table des matières

Avant Propos

Indispensable

Référence des logiciels qui seront installés via cette procédure

I. Vérification des dépôts actifs et re-chargement des informations relatives aux paquets

A. Activation des dépôts nécessaires

B. Re-chargement des informations relatives aux paquets :

II. Vérification de la manière dont sont traitées les dépendances des paquets recommandés

III. Installation du Serveur Web Apache2

IV. Installation du Serveur Mysql-5.0

V. Installation de PHP5 (avec modules pour Apache2 et Mysql-5.0)

VI. Pré-vérifications

A. Vérification du bon fonctionnement du Serveur Web Apache2

B. Vérification du bon fonctionnement de PHP5

a. Création du fichier phpinfo.php

b. Execution du fichier phpinfo.php

VII. Installation de quelques modules supplémentaires pour PHP5

VIII. Sécurisation du Serveur Mysql-5.0

IX. Installation de PhpMyAdmin (PMA)

X. Vérification du bon fonctionnement de PhpMyAdmin

POUR ALLER PLUS LOIN

Avant Propos

I. Les jeux de caractères (encodages) du Serveur Web Apache2

A. Laisser le navigateur Web choisir l'encodage approprié

B. Changer le jeu de caractères utilisé par défaut par le Serveur Web Apache2

II. Les Hôtes Virtuels (VirtualHost) – Méthodes

Introduction

1. Les Hôtes Virtuels basés sur l'adresse Ip

A. Association d'une deuxième adresse Ip à une carte Réseau (Ip aliasing)

B. Configuration des Hôtes Virtuels

2. Hôtes Virtuels basés sur le numéro de port

3. Hôtes Virtuels basés sur le nom

Quelques précisions concernant ce tutorial

L'auteur de ce tutorial

Avant Propos

Ce tutorial traite de la procédure à suivre pour installer une solution **LAMP** (*Linux, Apache, Mysql, Php/Perl*) sur un système (machine) disposant de la distribution **Ubuntu Dapper Drake (6.06 LTS)** en mode graphique.

Il s'adresse donc plus particulièrement aux nouveaux utilisateurs d'Ubuntu qui désirent rester en mode graphique à défaut de bien connaître l'administration d'un Serveur en lignes de commandes (shell).

Bien entendu, il s'adresse tout aussi bien aux développeurs professionnels désireux de travailler avec Ubuntu sur leur portable...

Ce Tutorial est proposé par Monsieur **Laurent DECLERCQ**, membre de la communauté Ubuntu.

Indispensable

La procédure traitée dans ce tutorial est à exploiter sur une nouvelle installation d'Ubuntu Dapper Drake ou sur un installation sur laquelle aucune solution **LAMP** n'a encore été installée.

Vous ne devez pas avoir choisi l'option **LAMP** à l'installation d'Ubuntu Dapper Drake.

Référence des logiciels qui seront installés via cette procédure

- **Apache2** (2.0.55-4ubuntu2) ;
- **Mysql-5.0** (5.0.21-3ubuntu1) ;
- **PHP5** (5.1.2-1ubuntu3) ;
- **PhpMyAdmin** (4:2.8.0.3-1) ;

et modules divers pour PHP5.

Etant précisé que la version des dits logiciels peut évoluer.

I. Vérification des dépôts actifs et re-chargement des informations relatives aux paquets

A. Activation des dépôts nécessaires :

Voici la liste des dépôts qui doivent être actifs pour l'installation de cette solution **LAMP** :

```
deb cdrom:[Ubuntu 6.06 _Dapper Drake_ - Alpha i386 (20060504)]/ dapper main restricted
deb http://fr.archive.ubuntu.com/ubuntu/ dapper main restricted
deb-src http://fr.archive.ubuntu.com/ubuntu/ dapper main restricted
deb http://fr.archive.ubuntu.com/ubuntu/ dapper universe
deb-src http://fr.archive.ubuntu.com/ubuntu/ dapper universe
```

Normalement, si vous n'avez pas modifié à la configuration des dépôts depuis que vous avez installé Ubuntu, seuls les deux derniers dépôts ci-dessus doivent être dé-commentés dans votre fichiers **sources.list**.

De même, il est vivement recommandé de désactiver tous les autres dépôts afin d'éviter tous problèmes de dépendances, notamment, si les deux dépôts suivants s'y trouvent, il est fortement conseillé de les commenter (désactiver) :

```
deb http://dotdeb.thefox.com.fr stable all
deb-src http://dotdeb.thefox.com.fr stable all
```

Le fichier **sources.list** se trouve dans le répertoire suivant :

```
/etc/apt
```

Pour dé-commenter (activer les dépôts) procédez comme suite :

```
gksudo gedit /etc/apt/sources.list
```

si vous utilisez l'éditeur de texte **gedit** (mode graphique) ou :

```
sudo nano /etc/apt/sources.list
```

si vous voulez le faire directement dans le terminal via l'éditeur de texte **nano**.

Note : Si vous utiliser **Kubuntu**, utiliser **kdesu** au lieu de **gksudo** et **kate** au lieu de **gedit**.

Une fois que le fichiers **sources.list** est édité, enlevez le signe **#** qui est situé devant les deux dépôts concernés et terminez en le sauvegardant.

Pour commenter (désactiver), ajoutez le signe **#** devant les dépôts concernés.

B. Re-chargement des informations relatives aux paquets :

Après avoir activé les dépôts nécessaires à l'installation de la solution **LAMP**, il faut que vous rechargez les informations relatives aux paquets disponibles.

Pour ce faire, il vous suffit de tapez la commande suivante dans un terminal :

```
sudo apt-get update
```

II. Vérification de la manière dont sont traitées les dépendances des paquets recommandés

Avant de commencer la procédure d'installation, il faut que vous vérifiez la manière dont sont traitées les dépendances des paquets recommandés :

Pour ce faire, ouvrez le gestionnaire de paquets **synaptic** (Système --> Administration --> Gestionnaire de paquets **synaptic**).

Une fois que le gestionnaire de paquets **synaptic** est ouvert, cliquez sur (**configuration --> préférences**) et dans la nouvelle fenêtre qui s'ouvre, dans l'onglet **général**, vérifiez que l'option : **traiter les paquets recommandés comme des dépendances** est bien activée.

Si ce n'est pas le cas, activez là et rechargez les informations relatives aux paquets comme vu précédemment ou via le gestionnaire de paquets **synaptic** directement.

III. Installation du Serveur Web Apache2

Pour installer le **Serveur Web Apache2**, il vous suffit de taper la commande suivante dans un terminal :

```
sudo apt-get install apache2 apache2-doc apache2-mpm-prefork
```

Ceci aura pour effet d'installer le **Serveur Web apache2** dans le répertoire **/etc/apache2** ainsi que sa documentation.

Ps : Il se peut que pendant l'installation, votre cd d'Ubuntu vous soit demandé. Pour éviter cela, ré-éditez le fichier **/etc/apt/sources.list** et commentez les lignes faisant références au cd-rom. Ceci étant effectué, sauvegardez le fichier et rechargez les informations relatives aux paquets disponibles en tapant la commande suivante dans un terminal :

```
sudo apt-get update
```

IV. Installation du Serveur Mysql-5.0

Pour installer le **Serveur Mysql-5.0**, il vous suffit de taper la commande suivante dans un terminal :

```
sudo apt-get install mysql-server-5.0
```

V. Installation de PHP5 (avec modules pour Apache2 et Mysql-5.0)

Pour installer **PHP5** et quelques modules supplémentaires pour le **Serveur Web Apache2**, il vous suffit de taper la commande suivante dans un terminal :

```
sudo apt-get install php5 libapache2-mod-php5 php5-mysql
```

Pendant l'installation, vous pouvez voir apparaître quelques erreurs, notamment lorsque le **Serveur Web Apache2** va être redémarré, vous allez sûrement rencontrer cette erreur :

```
apache2: Could not determine the server's fully qualified domain name...
```

Ceci n'est absolument pas gênant et peut être réglé très facilement. Il vous suffit de rajouter la directive **ServerName** dans le fichier **apache2.conf** qui se trouve dans le répertoire **/etc/apache2/**.

Exemple :

```
ServerName www.nuxwin.com
```

VI. Pré-vérifications

A ce stade de l'installation, il est conseillé de procéder à quelques vérifications, notamment :

Vérifier :

- Le bon fonctionnement du **Serveur Web Apache2** ;
- Le bon fonctionnement de **PHP5**.

Avant tout, vous devriez redémarrer le **Serveur Web Apache2** pour que les changements soient pris en compte.

Pour ce faire, il vous suffit de taper la commande suivante dans un terminal :

```
sudo /etc/init.d/apache2 force-reload
```

A. Vérification du bon fonctionnement du Serveur Web Apache2 :

Dans la barre de votre navigateur internet (firefox...), tapez l'url suivante :

```
http://127.0.0.1/
```

ou son équivalent :

```
http://localhost/
```

Si tout s'est bien passé, vous devriez voir une page Web dans laquelle l'index du répertoire Web apparaît ainsi que le répertoire **apache2-default**.

Ps : **localhost** est en réalité un alias de l'adresse Ip **127.0.0.1** qui représente votre machine.

B. Vérification du bon fonctionnement de PHP5 :

a. Création du fichier phpinfo.php :

Pour vérifier que **PHP5** a bien été installé et est fonctionnel, vous allez devoir créer un fichier d'usage, le fameux **phpinfo.php**.

Pour ce faire, créez un nouveau document dans le répertoire **/var/www** que vous nommerez **phpinfo.php** grâce à cette commande :

```
gksudo gedit /var/www/phpinfo.php
```

ou autre éditeur de texte (**nano**, **kedit**,...)

et insérez-y le code suivant :

```
<?php phpinfo(); ?>
```

Enfin, finissez en enregistrant le document.

b. Execution du fichier phpinfo.php :

Le fichier **phpinfo.php** étant créé, vous devez l'exécuter.

Pour exécuter le fichier **phpinfo.php**, il vous suffit de tapez l'url suivante dans la barre d'adresse de votre navigateur internet (firefox...) :

```
http://127.0.0.1/phpinfo.php
```

ou celle-ci :

```
http://localhost/phpinfo.php
```

si vous voyez apparaître ce message d'erreur :

```
Warning: Unknown: failed to open stream: Permission denied in Unknown on line 0
```

```
Warning: Unknown: Failed opening '/var/www/phpinfo.php' for inclusion  
(include_path='.:usr/share/php:usr/share/pear') in Unknown on line 0
```

C'est simplement dû aux droits actuels du fichier **phpinfo.php** que vous venez de créer.

Pour résoudre cette erreur, il suffit donc de changer les droits du fichier.

Actuellement le fichier **phpinfo.php** peut seulement être lu, modifié et exécuté par l'utilisateur **root**. Vous comprendrez donc qu'il ne peut être lu et exécuté par les autres utilisateurs.

Il faut donc que vous le rendiez accessible, en lecture et exécution par les utilisateurs autres que **root**. Pour ce faire, il vous suffit de lui appliquer un **chmod 605** en tapant la commande suivante dans un terminal :

```
sudo chmod 605 /var/www/phpinfo.php
```

Cette modification étant effectuée, retenez le point **b** du grand **B** de ce paragraphe.

Ps : Si votre navigateur internet vous demande de télécharger le fichier **phpinfo.php**, cela peut venir d'un problème de module mal chargé. Dans ce cas, vous pouvez tenter de résoudre cette "erreur" en tapant les commandes suivantes pour activer le module php5 :

1. On active le module **php5** :

```
sudo a2enmod php5
```

2. On re-démarre le **Serveur Web Apache2** pour que les changements soient pris en compte :

```
sudo /etc/init.d/apache2 force-reload
```

Si tout s'est bien passé, vous devriez voir apparaître une **page Web** dans laquelle se trouvent plusieurs informations, notamment celles liées à la version de **PHP** utilisée sur votre système.

Ps : Par la suite, lorsque vous allez créer des documents **HTML** et/ou des **scripts PHP**, pensez à appliquer les bons droits (permissions) sur les répertoires et fichiers de sorte qu'ils puissent être lus et exécutés. Ceci est une source d'erreurs très répandue chez les nouveaux utilisateurs Linux.

D'une manière générale, un **chmod 755** convient à la plupart des configurations.

VII. Installation de quelques modules supplémentaires pour PHP5

Dans la plupart des cas, des modules supplémentaires pour **PHP5** sont nécessaires au bon fonctionnement des applications Web l'utilisant.

Vous allez donc installer quelques modules qui sont souvent nécessaires.

Dans un terminal, tapez la commande suivante :

```
sudo apt-get install php-pear php5-cli php5-gd php5-sqlite php5-xsl php5-mcrypt
```

Ps : D'autres modules pour **PHP5** sont disponibles. Vous pouvez voir lesquels en exécutant le gestionnaire de paquets *synaptic* et en faisant une recherche sur **PHP5** ou en tapant la commande suivante dans un terminal :

```
apt-cache search php5
```

VIII. Sécurisation du Serveur Mysql-5.0

Pour l'instant, le **Serveur Mysql-5.0** est exécuté sous son utilisateur **root** sans mot de passe.

Il est donc nécessaire de sécuriser un tant soit peu l'installation du dit Serveur.

Dans un terminal, tapez la commande suivante :

```
sudo mysql_secure_installation
```

Une fois cette commande exécutée le script de sécurisation du **Serveur Mysql-5.0** démarre.

Ce script vous pose plusieurs questions :

```
Enter current password for root (enter for none):
```

Ici, le script vous demande d'entrer le mot de passe courant de l'utilisateur **root**. N'en ayant pas encore défini, vous devez simplement appuyer sur la touche **Enter**.

```
Setting the root password ensures that nobody can log into the MySQL  
root user without the proper authorisation.
```

```
Set root password? [Y/n]
```

Ici, il vous est demandé si vous voulez attribuer un mot de passe à l'utilisateur **root**. Vous devez donc taper la lettre **Y**.

```
New password:
```

Bien ici, c'est assez simple, vous devez taper le mot de passe que vous voulez attribuer à l'utilisateur **root** du **Serveur Mysql-5.0**. Attention à bien le choisir et de vous en rappeler. Il sera aussi utilisé pour la connexion à **PhpMyAdmin**.

```
By default, a MySQL installation has an anonymous user, allowing anyone  
to log into MySQL without having to have a user account created for  
them. This is intended only for testing, and to make the installation  
go a bit smoother. You should remove them before moving into a  
production environment.
```

```
Remove anonymous users? [Y/n]
```

Ici, il vous est indiqué qu'à l'installation du **Serveur Mysql-5.0**, un utilisateur anonyme a été créé. Ceci étant dangereux pour un environnement de production, tapez sur la lettre **Y** pour le supprimer.

```
Normally, root should only be allowed to connect from 'localhost'. This  
ensures that someone cannot guess at the root password from the network.
```

```
Disallow root login remotely? [Y/n]
```

Si vous n'avez pas besoin d'administrer votre base de données à distance, ce qui devrait être le cas de la plupart des utilisateurs, tapez sur la lettre **Y**. Ainsi, l'utilisateur **root** ne pourra se connecter qu'en local (la machine sur laquelle le **Serveur Mysql-5.0** est installé).

```
Remove test database and access to it? [Y/n]
```

Ici, contentez-vous d'appuyer sur la touche **Enter**.

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n]

Appuyez sur la lettre **Y** pour prendre en compte les changements que vous venez de faire et les appliquer immédiatement.

Voilà pour ce qui est de la sécurisation du **Serveur Mysql-5.0**.

IX. Installation de PhpMyAdmin (PMA)

AVERTISSEMENT

Il est clairement rappelé que pour pouvoir installer le logiciel **PhpMyAdmin**, il faut que les dépôts universe soient décommentés dans votre fichier sources.list.

Pour installer **PhpMyAdmin**, il vous suffit de taper la commande suivante dans un terminal :

```
sudo apt-get install phpmyadmin
```

Ps : Il est important d'installer **PhpMyAdmin** en dernier dans le cadre de ce tutoriel et il est aussi important d'avoir défini un mot de passe pour l'utilisateur **root** du **Serveur Mysql-5.0** avant de l'installer. A défaut, Vous risqueriez de rencontrer des problème avec **PhpMyAdmin**.

X. Vérification du bon fonctionnement de PhpMyAdmin

Pour vérifier le bon fonctionnement de **PhpMyAdmin**, tapez ceci dans la barre d'adresse de votre navigateur internet :

```
http://127.0.0.1/
```

ou ceci :

```
http://localhost/
```

Et ensuite, dans la page Web qui apparaît, cliquez sur le répertoire nommé **phpmyadmin**.

Vous pouvez aussi taper l'url de **PhpMyAdmin** directement dans la barre d'adresse de votre navigateur :

```
http://127.0.0.1/phpmyadmin/
```

ou ceci :

```
http://localhost/phpmyadmin/
```

ce qui donnera le même résultat, à savoir, l'affichage d'une nouvelle page Web qui n'est autre que la page d'authentification de **PhpMyAdmin**.

Pour vous connecter à **PhpMyAdmin**, faites comme ceci :

```
Nom d'utilisateur
```

le nom de l'utilisateur Mysql, il s'agit de l'utilisateur **root**.

Mot de passe :

Le mot de passe que vous avez défini plus haut pour l'utilisateur **root** du **Serveur Mysql-5.0**.

Si tout s'est bien passé, vous devriez être connecté à **PhpMyAdmin** et pouvoir créer/gérer vos bases de données.

Ps : Il existe déjà deux bases de données, n'y touchez pas !!!.

Note : Si votre navigateur vous propose de télécharger un **fichier PHTML** au lieu d'afficher **PhpMyAdmin**, cela vient sûrement d'un problème de module non chargé.

Dans ce cas, vous pouvez tenter de résoudre cette "erreur" en tapant les commandes suivantes dans un terminal :

1. On active le module php5 :

```
sudo a2enmod php5
```

2. On re-démarre le Serveur Web Apache2 :

```
sudo /etc/init.d/apache2 force-reload
```

Ps : N'oubliez pas de vider le cache de votre navigateur et de le fermer complètement avant de recommencer.

Fin du tutoriel

POUR ALLER PLUS LOIN

Avant Propos

Les sections suivantes pourraient être résumées en 5 ou 10 lignes tout au plus. Ce faisant, mon but n'est pas de donner des solutions exemptes de toutes explications. Mon but est de fournir des explications qui permettront à l'utilisateur final de bien comprendre le fonctionnement du **Serveur Web Apache2** afin qu'il puisse, de lui-même, répéter les méthodes sans recourir, à chaque ré-installation, à ce document.

I. Les jeux de caractères (encodages) du Serveur Web Apache2

Par défaut, lorsqu'on installe le **Serveur Web Apache2**, c'est le jeu de caractères **UTF-8** qui est utilisé. Ceci peut être vérifié en éditant le fichier **charset** qui se trouve dans le dossier **/etc/apache2/conf.d**.

Ce fichier contient la ligne suivante :

AddDefaultCharset UTF-8

Il en résulte qu'à défaut d'avoir formaté les documents Web avec le jeu de caractères **UTF-8**, les accents ne pourront s'afficher correctement lors du traitement des dits documents par le **Serveur Web Apache2**.

Pour exemple, le **é** sera encodé de cette manière : 

Pour résoudre, ce problème, plusieurs solutions s'offrent à vous,

RAPPEL

La directive **AddDefaultCharset** spécifie le nom du jeu de caractères qui sera ajouté à toutes les réponses qui n'ont aucun paramètre sur le type de contenu dans l'en-tête HTTP. Elle remplace le jeu de caractères spécifié dans le corps du document Web par l'inclusion du **marqueur META** (ex : `<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">`).

Pour exemple, si la directive **AddDefaultCharset** est présente dans les fichiers de configuration du **Serveur Web Apache2** avec l'argument **UTF-8**, vous aurez beau inclure le **marqueur META** (ex : `<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">`) dans le corps de vos documents Web, cela ne changera rien au problème d'encodage puisque cette directive supplétera ledit marqueur.

Note explicative :

Le contenu du fichier **charset** mentionné ci-dessus est inclut dans la configuration du **Serveur Web Apache2** grâce à la directive Include => **Include /etc/apache2/conf.d/[^#]** .

Cette directive et l'argument qui l'accompagne font en sorte que tout le contenu des fichiers qui se trouvent dans le répertoire **/etc/apache2/conf.d** soient inclus à la configuration du **Serveur Web Apache2**.

A. Laisser le navigateur Web choisir l'encodage approprié :

Comme nous l'avons vu, par défaut, il est demandé au **Serveur Web Apache2** d'utiliser le jeu de caractères **UTF-8** lors du traitement des documents Web. Nous pouvons aisément désactiver cette fonction ce qui aura pour effet de laisser le navigateur internet choisir le jeu de caractères à employer.

Pour désactiver cette fonction, deux solutions existent :

1. Vous commentez la directive d'inclusion **Include /etc/apache2/conf.d/[^#]** qui se trouve dans le fichier **/etc/apache2/apache2.conf**.
ou
2. Vous commentez la ligne **AddDefaultCharset UTF-8** qui se trouve dans le fichier **/etc/apache2/conf.d/charset** .

RAPPEL

Pour commenter il suffit simplement de rajouter le signe # devant le code. Pour dé-commenter, il suffit simplement de retirer le signe # qui se trouve devant le code.

ATTENTION

Si vous optez pour la première solution, tout le contenu des fichiers qui se trouvent dans le dossier `/etc/apache2/conf.d` ne sera plus pris en compte par le **Serveur Web Apache2**. Il est donc préférable de choisir la deuxième solution.

B. Changer le jeu de caractères utilisé par défaut par le Serveur Web Apache2 :

Au lieu de désactiver la fonction de définition du jeu de caractères utilisé par défaut par le **Serveur Web Apache2**, nous pouvons définir celui que nous désirons employer.

Là encore, il existe plusieurs solutions pour le faire.

1. En modifiant le fichier `charset` du répertoire `/etc/apache2/conf.d`.

Comme nous l'avons vu, par défaut, le **Serveur Web apache2** utilise le jeu de caractères **UTF-8**, ce dernier étant défini par le contenu du fichier `charset` qui se trouve dans le répertoire `/etc/apache2/conf.d` lui-même intégré à la configuration du **Serveur Web Apache2** grâce à la directive Include => `Include /etc/apache2/conf.d/[^.#]`.

Pour modifier le jeu de caractères utilisé par défaut par le **Serveur Web Apache2**, il vous suffit donc d'éditer le fichier `charset` et de remplacer la ligne :

```
AddDefaultCharset UTF-8
```

par celle de votre choix.

Exemple pour l'encodage **ISO-8859-1** :

```
AddDefaultCharset ISO-8859-1
```

2. En modifiant le fichier `apache2.conf`

Si vous éditez votre fichier `apache2.conf` qui se trouve dans le répertoire `/etc/apache2`, vous pourrez y trouver la directive et l'argument suivants :

```
#AddDefaultCharset ISO-8859-1
```

Comme vous pourrez le constater, cette ligne est commentée par défaut. Si vous désirez que le jeu de caractères **ISO-8859-1** soit utilisé par défaut par le **Serveur Web Apache2**, il vous suffit donc de décommenter cette ligne.

Cette ligne deviendra donc :

```
AddDefaultCharset ISO-8859-1
```

Note explicative

A ce stade, et si vous avez bien suivi mes explications depuis le début, vous devriez vous poser la question suivante :

Si je décommente la ligne **#AddDefaultCharset ISO-8859-1** qui se trouve dans le fichier **/etc/apache2/apache2.conf**, que deviendra la directive **AddDefaultCharset** qui se trouve dans le fichier **/etc/apache2/conf.d/charset** ?

Et bien la réponse est simple.

Lorsque vous demandez au **Serveur Web apache2** de relire ses fichiers de configuration, il suit un ordre qui est celui du plus court chemin. Autrement dit, si deux directives ayant la même fonction sont déclarées, c'est la dernière qui sera prise en compte par le **Serveur Web Apache2**.

Dans notre cas, le contenu fichier **/etc/apache2/conf.d/charset** est inclut dans le fichier **/etc/apache2/apache2.conf** avant la déclaration de la directive **AddDefaultCharset ISO-8859-1**. C'est pour cette raison que ce sera cette dernière directive qui sera prise en compte par le **Serveur Web Apache2** si nous la décommentons.

Attention tout de même, car ceci n'est pas vérifié pour toutes les directives. Dans le cas présent, cette directive s'applique à la configuration générale du **Serveur Web Apache2**. Il en serait autrement s'il s'agissait d'une directive propre à un hôte virtuel (Virtualhost) où encore, à un répertoire Web (<Directory > ... </Directory>)...

Comme usuellement, il convient de demander au **Serveur Web Apache2** de relire ses fichiers de configuration après avoir effectué des modifications pour que ces dernières soient prises en comptes.

Pour ce faire, il vous suffit de taper la commande suivante dans un terminal :

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

II. Les Hôtes Virtuels (VirtualHost) – Méthodes

Introduction

Le **Serveur Web Apache2** est capable de gérer simultanément plusieurs arborescences Web grâce à la notion d'hôtes Virtuels (Virtual Hosts). Dans cette section je vous propose de prendre connaissance des divers méthodes existantes pour mettre en place vos **Hôtes Virtuels** (*VirtualHost*).

Dans le cadre de cette mise en application je renseigne le fichier **/etc/hosts** afin d'assurer la résolution de nom (Nom --> Adresse Ip) des Hôtes Virtuels que je crée. Ceci est nécessaire que si votre Serveur DNS (*Serveur Bind*) n'est pas installé et/ou configuré pour vos domaines ou encore, que les noms DNS attribués aux Hôtes Virtuels sont purement fictifs ou ne vous appartiennent pas.

Ps : Si vous désirez tester toutes les méthodes proposées, pensez à effacer la configuration précédente pour ne pas créer des conflits entre les Hôtes Virtuels. Il en va de même pour le fichier **/etc/hosts**.

Comme évoqué ci-dessus, le **Serveur Web Apache2** peut gérer plusieurs arborescences Web grâce à la notion d'hôtes Virtuels (*VirtualHost*).

Pour cela, plusieurs méthodes existent :

- Les Hôtes Virtuels **basés sur l'adresse Ip** ;
- Les Hôtes Virtuels **basés sur le numéro de port** ;
- Les Hôtes Virtuels **basés sur le nom**.

1. Les Hôtes Virtuels basés sur l'adresse Ip :

Dans le cadre de cette méthode, le Serveur est soit doté de plusieurs interfaces réseau (plusieurs cartes réseau), soit doté de plusieurs adresses Ip associées à une seule interface réseau (une seule carte réseau). Dans ce dernier cas, on parlera d'Ip aliasing. Les systèmes Linux, notamment les distributions Ubuntu et Debian, permettent de mettre facilement en oeuvre cette dernière fonctionnalité (Ip aliasing).

A. Association d'une deuxième adresse Ip à une carte Réseau (Ip aliasing) :

Dans cet exemple, nous partirons du principe que nous possédons une seule interface réseau (une seule carte réseau) à laquelle nous voulons associer une seconde adresses Ip.

Dans un premier temps, nous tapons la commande suivante dans un terminal pour prendre connaissance de la configuration de notre interface réseau :

```
sudo ifconfig
```

et voici le résultat que nous obtenons :

```
coucou@serveur:~$ sudo ifconfig
eth0  Lien encap:Ethernet HWaddr 00:13:D3:3C:58:84
      inet adr:192.168.0.2 Bcast:192.168.0.255 Masque:255.255.255.0
      adr inet6: fe80::213:d3ff:fe3c:5884/64 Scope:Lien
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      Packets reçus:1257 erreurs:0 :0 overruns:0 frame:0
      TX packets:1247 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 lg file transmission:1000
      Octets reçus:1037747 (1013.4 KiB) Octets transmis:423117 (413.2 KiB)
      Interruption:209 Adresse de base:0x8000

lo    Lien encap:Boucle locale
      inet adr:127.0.0.1 Masque:255.0.0.0
      adr inet6: ::1/128 Scope:Hôte
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      Packets reçus:429 erreurs:0 :0 overruns:0 frame:0
      TX packets:429 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 lg file transmission:0
      Octets reçus:117583 (114.8 KiB) Octets transmis:117583 (114.8 KiB)
```

Ce qui nous intéresse ici c'est la première série d'information. Il nous est indiqué que nous disposons d'une interface réseau de type Ethernet d'adresse Ip 192.168.0.2.

Nous disposons donc d'une seule et unique adresse Ip qui est 192.168.0.2.

Nous devons donc associer une seconde adresse Ip à notre interface réseau. Pour ce faire, il nous suffit simplement de taper la commande suivante dans un terminal :

```
sudo ifconfig eth0:0 192.168.0.100
```

Grâce à la commande évoquée ci-dessus, nous avons associé une seconde adresse Ip ==> 192.168.0.100 à notre interface réseau (carte réseau).

Dès à présent, notre interface réseau dispose de deux adresse Ip bien distinctes :

192.168.0.2 (Adresse Ip)

et

192.168.0.100 (Alias Ip)

Ceci est vérifiable en tapant la commande suivante dans un terminal :

```
sudo ifconfig -a
```

B. Configuration des Hôtes Virtuels :

Ayant associé une nouvelle adresse Ip à notre unique carte réseau, nous pouvons désormais créer nos deux **Hôtes Virtuels** (sites virtuels) utilisant chacun une adresse Ip bien distincte.

Dans un premier temps, nous renseignons le fichier **/etc/hosts** avec deux noms de sites (ceux de nos hôtes Virtuels) associés respectivement aux deux adresses Ip disponibles afin de permettre la résolution de nom (Nom --> Adresse Ip).

Exemple de fichier hosts :

```
127.0.0.1    localhost
192.168.0.2  virtual1.com
192.168.0.100 virtual2.com
```

Une fois notre fichier **/etc/hosts** correctement renseigné, il ne nous reste plus qu'à configurer nos Hôtes Virtuels.

Pour ce faire, il nous suffit de créer deux nouveaux fichiers dans le répertoire **/etc/apache2/sites-available** :

1er fichier de configuration :

Dans cet exemple, le premier fichier de configuration se nomme : **virtual1.conf**

Ce fichier contient le contenu minimal suivant :

```
<VirtualHost 192.168.0.2:80>
    DocumentRoot    /var/www/virtual1.com
    ServerName  virtual1.com
</VirtualHost>
```

2ème fichier de configuration :

Dans cet exemple, le deuxième fichier de configuration se nomme : **virtual2.conf**

Ce fichier contient le contenu minimal suivant :

```
<VirtualHost 192.168.0.100:80>
    DocumentRoot /var/www/virtual2.com
    ServerName  virtual2.com
</VirtualHost>
```

Note explicative

Dans chacun des fichiers de configuration, nous devons prévoir une strophe **<VirtualHost>** avec les deux directives de base : **DocumentRoot** et **ServerName**.

La directive **DocumentRoot**, à ne pas confondre avec la directive **ServerRoot**, permet de déclarer l'emplacement de l'arborescence Web de l'hôte virtuel qui sera accessible par les clients. Le nom du répertoire ne doit pas comporter le caractère / final.

La directive **ServerName** permet de définir le nom DNS du Serveur ainsi que le port utilisé par défaut. Dans le cadre d'Hôtes Virtuels, on parlera de Serveurs Virtuels.

Ps : Les répertoires **/var/www/virtual1.com** et **/var/www/virtual2.com** doivent exister et doivent bénéficier des bonnes permissions. A défaut, le **Serveur Web Apache2**, lors du reload (re-lecture de ses fichiers de configurations), renverra des erreurs. Pour les tests, vous pouvez appliquer un **chmod 755** sur ces répertoires.

Nos deux Hôtes Virtuels étant configurés, nous devons dès à présent les activer. Pour ce faire, nous tapons les commandes suivantes dans un terminal :

```
sudo a2ensite virtual1.conf
```

et

```
sudo a2ensite virtual2.conf
```

Note explicative

La commande **a2ensite** a pour effet de créer des liens symboliques des deux fichiers de configuration (VirtualHost) créés précédemment, liens qui seront placés dans le répertoire **/etc/apache2/sites-enabled**, ce qui permettra au Serveur Web Apache2 de les interpréter grâce à la directive d'inclusion ==> **Include /etc/apache2/sites-enabled/[^.#]** qui elle, se trouve dans le fichier de configuration principale **/etc/apache2/apache2.conf** du **Serveur Web Apache2**.

Enfin, pour que les modifications soient prises en comptes par le **Serveur Web Apache2**, nous demandons à ce dernier de relire ses fichiers de configuration. Pour ce faire, il nous suffit de taper la commande suivante dans un terminal :

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

2. Hôtes Virtuels basés sur le numéro de port :

Par défaut, le **Serveur Web Apache2** est configuré pour écouter sur le **port 80**. Ce faisant, il est possible d'utiliser des ports bien distincts pour chaque Hôtes Virtuels.

Exemple :

Dans cette exemple, les deux hôtes Virtuels **virtual1.com** et **virtual2.com** se partagent la même adresse Ip : **192.168.0.2**. Seul le port d'écoute est différent.

Dans un premier temps, nous allons éditer le fichier **/etc/hosts** afin de donner deux noms bien distincts à la seule et unique adresse Ip de notre interface réseau. Ceci permettra la résolution de nom (Nom --> Adresse Ip).

Exemple de fichier hosts :

```
127.0.0.1    localhost
192.168.0.2  virtual1.com
192.168.0.2  virtual2.com
```

Ensuite, nous créons deux fichiers de configuration (VirtualHost) que nous plaçons dans le répertoire `/etc/apache2/sites-available`.

Voici ce que contiennent ces deux fichiers de configuration :

1er fichier de configuration qui se nomme `virtual1.conf` :

```
<VirtualHost 192.168.0.2:80>
    DocumentRoot    /var/www/virtual1.com
    ServerName  virtual1.com
</VirtualHost>
```

2ème fichier de configuration qui se nomme `virtual2.conf` :

```
<VirtualHost 192.168.0.2:8080>
    DocumentRoot    /var/www/virtual2.com
    ServerName  virtual2.com
</VirtualHost>
```

Note explicative :

Dans cet exemple, le site `virtual1.com` sera accessible via l'url : `http://virtual1.com` et le site `virtual2.com` sera accessible via l'url : `http://virtual2.com:8080`.

Ps : Les répertoires `/var/www/virtual1.com` et `/var/www/virtual2.com` doivent exister et doivent bénéficier des bonnes permissions. A défaut, le **Serveur Web Apache2**, lors du reload, renverra des erreurs. Pour les testes, vous pouvez appliquer un `chmod 755` sur ces répertoires.

Encore une fois, nous devons activer les deux Hôtes Virtuels avec la commande `a2ensite` (cf. ci-avant) et nous devons demander au **Serveur Web Apache2** de relire ses fichiers de configuration pour que les changements soient pris en comptes ==> `/etc/init.d/apache2 reload`.

3. Hôtes Virtuels basés sur le nom :

Cette troisième méthode et celle qui est la plus utilisée et aussi la plus conseillée. Elle tend même à devenir un standard. Il s'agit simplement d'associer plusieurs noms DNS à une seule adresse IP.

Dans l'exemple suivant, nous allons configurer deux Hôtes Virtuels qui utiliseront tout les deux l'adresse Ip `192.168.0.2`.

Dans un premier temps, nous allons éditer le fichier `/etc/hosts` afin de donner deux noms bien distincts à la seule et unique adresse Ip de notre interface réseau. Ceci permettra la résolution de nom (Nom --> Adresse Ip).

Exemple de fichier `/etc/hosts` :

```
127.0.0.1    localhost
192.168.0.2  virtual1.com
```

192.168.0.2 virtual2.com

Notre fichier `/etc/hosts` étant correctement renseigné, nous allons rajouter la directive `NameVirtualHost` à la fin du fichier `/etc/apache2/apache2.conf` qui n'est autre que le fichier de configuration principale du **Serveur Web Apache2**.

Cette directive correspond à celle sur laquelle le **Serveur Web Apache2** acceptera uniquement les requêtes adressées aux Hôtes Virtuels que nous allons définir ci-après.

A la fin du fichier `/etc/apache2/apache2.conf` nous rajoutons donc :

NameVirtualHost 192.168.0.2:80

Ensuite, nous créons deux nouveaux fichiers de configurations (VirtualHost) que nous plaçons dans le répertoire `/etc/apache2/sites-available`.

Voici ce que contiennent ces deux fichiers de configuration :

1er fichier de configuration qui se nomme virtual1.conf :

```
<VirtualHost 192.168.0.2:80>
    DocumentRoot    /var/www/virtual1.com
    ServerName  virtual1.com
</VirtualHost>
```

2ème fichier de configuration qui se nomme virtual2.conf :

```
<VirtualHost 192.168.0.2:80>
    DocumentRoot    /var/www/virtual2.com
    ServerName  virtual2.com
</VirtualHost>
```

Ps : Les répertoires `/var/www/virtual1.com` et `/var/www/virtual2.com` doivent exister et doivent bénéficier des bonnes permissions. A défaut, le **Serveur Web Apache2**, lors du reload, renverra des erreurs. Pour les tests, vous pouvez appliquer un `chmod 755` sur ces répertoires.

Encore une fois, nous devons activer les deux Hôtes Virtuels avec la commande `a2ensite` et nous devons demander au **Serveur Web Apache2** de relire sa configuration pour que les modifications soient pris en comptes ==> `/etc/init.d/apache2 reload`.

ATTENTION : Dans les exemples ci-dessus, je n'ai pas fait état des directives qui d'usage, doivent être présentes dans les fichiers de configurations des Hôtes Virtuels (*VirtualHost*), notamment : `ServerAdmin`, `ErrorLog` et `CustomLog` qui sont des directives qu'il est fortement conseillé d'inclure dans les fichiers de configuration de vos Hôtes Virtuels.

Quelques précisions concernant ce tutorial

J'ai rédigé ce tutorial pendant l'installation de la solution LAMP proposée sur mon serveur de développement. Je précise que je suis parti d'une nouvelle installation d'Ubuntu Dapper Drake (6.06 LTS) et donc, qu'aucun des logiciels sus-mentionnés n'avaient été installés sur mon système auparavant.

Enfin, Je tiens à préciser que je n'ai rencontré aucune erreur particulière pendant l'installation de cette solution LAMP.

L'auteur de ce tutorial

Tutorial rédigé par Monsieur **Laurent DECLERCQ**, Administrateur du Portail nuxwin.com, membre de la communauté Ubuntu.

Ce tutorial est distribué sous double licence : [CC BY-SA](#) & [GNU FDL](#)

Une remarque, un commentaire, besoin d'aide concernant ce tutorial ? =>

Adresse E-Mail de l'auteur : ubuntu@nuxwin.com

Portail Web de l'auteur : <http://www.nuxwin.com>

Date de création : **06/06/2006 02:02**

Ce document PDF a été remis en forme et mis à jour le 27 décembre 2006.

Travaux d'amélioration en cours :

- Sécuriser un site Web via protocol SSL (*Secure Socket Layer*) ;

Bonne continuation avec Ubuntu.