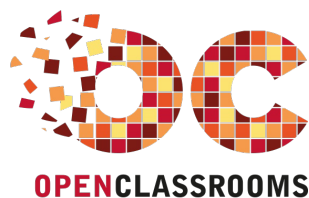


# Mise en place des serveurs Apache et DNS

Par wawanopoulos



[www.openclassrooms.com](http://www.openclassrooms.com)

*Licence Creative Commons 4 2.0  
Dernière mise à jour le 22/08/2010*

## Sommaire

Sommaire .....	2
Mise en place des serveurs Apache et DNS .....	3
Configuration IP du serveur .....	3
Introduction .....	3
Attribution d'une adresse IP à notre machine .....	3
Configuration du serveur DNS .....	4
Mise en place du serveur Apache .....	6
Installation du paquet apache2 sous Linux .....	6
Création d'une page HTML de test .....	7
Configuration du serveur Apache .....	7
Hébergement virtuel .....	10
Hébergement virtuel par nom de domaine .....	10
Hébergement virtuel par port .....	12
Hébergement virtuel par adresse IP .....	14
Partager .....	15



# Mise en place des serveurs Apache et DNS



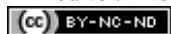
Par wawanopoulos

Mise à jour : 22/08/2010

Difficulté : Intermédiaire



Durée d'étude : 15 jours



Apache est un serveur web permettant de partager des pages web stockées localement à des utilisateurs connectés : des clients. La configuration d'un tel serveur peut paraître fastidieuse puisqu'elle demande quelques connaissances de base de Linux, mais ce tutoriel a pour but de vous prouver que cette configuration peut être très simple.

Pour vous mettre l'eau à la bouche, voici ce que vous devriez obtenir !



Sommaire du tutoriel :



- Configuration IP du serveur
- Configuration du serveur DNS
- Mise en place du serveur Apache
- Hébergement virtuel

## Configuration IP du serveur

### Introduction

Avant de mettre en place notre serveur apache, il est nécessaire d'attribuer des paramètres réseaux pour notre machine. Je veux bien sur parler d'une adresse IP et d'un nom de machine grâce au serveur DNS.



Ohh non, je suis déjà perdu, je ne sais pas faire ça !

Pas de panique, voici dans les parties suivantes, un petit rappel sur la configuration d'une adresse IP et d'un serveur DNS.

La configuration d'une adresse IP peut se faire de différentes manières. Dans notre cas, il s'agit d'attribuer à notre serveur, une adresse IP qui soit persistante, c'est à dire qu'elle soit toujours la même après un redémarrage de l'ordinateur.



Vous comprenez pourquoi ? Parce que lorsque nous allons mettre en place le serveur DNS, celui-ci se chargera de faire la correspondance entre une adresse IP et un nom de machine. Il ne faut donc pas que l'adresse IP change.

### Attribution d'une adresse IP à notre machine

Pour attribuer une adresse IP à votre machine, il est nécessaire d'ouvrir un fichier de configuration : le fichier **interfaces**.

#### Code : Console

```
vi /etc/network/interfaces
```



C'est quoi "vi" ?

Il s'agit d'un éditeur de texte présent sous Linux, qui va nous permettre d'ouvrir notre fichier. Vous pouvez en utiliser un autre comme gedit par exemple.

Une fois le fichier ouvert, remplacez simplement les quelques lignes présentes à l'intérieur, par celles-ci :

**Code : Console**

```
iface eth0 inet static
address 192.168.21.6
netmask 255.255.255.0
broadcast 192.168.21.255
```

Pour prendre en compte les modifications, enregistrez votre travail sous vi (en tapant ":wq"), et redémarrez votre interface en tapant ceci :

**Code : Console**

```
/etc/init.d/networking restart
```

Félicitations, vous venez d'attribuer une adresse IP fixe à votre machine !

Passons désormais à la mise en place du serveur DNS !

## Configuration du serveur DNS

Le réseau est un réseau local en 192.168.21.0/24. Il n'y a aucune passerelle. L'ordinateur salon.sdz.tp (192.168.21.1) est le serveur DNS. Cela indique que chaque client doit renseigner sa configuration sur cette machine. Quand je parle de configuration, il s'agit de renseigner l'adresse IP du client et un nom de machine associé. Chaque poste est connecté au réseau en utilisant l'interface eth0.

Comme vous le voyez, nous avons choisi un nom de domaine: **sdz.tp**.

Cela veut dire que chaque ordinateur client possédant un nom pourra être accessible via l'adresse suivante : **nomduclient.sdz.tp**.

Pour créer ce nom de domaine, il faut se rendre dans le fichier named.conf :

**Code : Console**

```
vi /etc/named.conf
```

Dans ce fichier se trouve la configuration de notre nom de domaine:

**Code : Console**

```
Zone « sdz.tp » {                                //Il s'agit de la zone direct. Voir explications c
Type master;                                     //Cette ligne indique que le type est maître c'
File « /var/cache/bind/sdz.tp.hosts » ;         //Il s'agit du chemin pour accéder au fi
};

Zone "21.168.192.in-addr.arpa"{                  //Il s'agit de la zone inverse. Voir explication
Type master;                                     //Cette ligne indique que le type est maître c'e
```

```
File "/var/cache/bind/sdz.tp.rev"; //Il s'agit du chemin pour accéder au fichier;  
};
```

Le fichier de zone directe permet à partir de n'importe quel nom du type \*.sdz.tp de fournir l'adresse IP de la machine. On dit que la machine serveur DNS a autorité sur la zone **sdz.tp**.

Le fichier de zone inverse permet à partir de n'importe quelle adresse IP du type 21.168.192.\* de fournir l'adresse de la machine. On dit que la machine serveur DNS a autorité sur la zone 110.50.210.IN-ADDR-ARPA.

**Notre serveur DNS a donc autorisé sur 2 zones : sdz.tp et 21.168.192.in-addr.arpa**

Maintenant que notre nom de domaine est créé, il faut déclarer chaque machine cliente avec un nom associé. Cette déclaration se fait dans les fichiers de zone directe et inverse associé à notre nom de domaine : **sdz.tp.hosts** et **sdz.tp.rev**.

#### Code : Console

```
vi /var/cache/bind/sdz.tp.rev
```

Et ajoutez la ligne suivante :

#### Code : Console

```
4 IN PTR sdz.sdz.tp.
```



Qu'est-ce que ça veut dire ?

PTR associe une adresse IP à un enregistrement de nom de domaine, aussi dit « reverse » puisqu'il fait exactement le contraire du A (ci-dessous).

Ajoutons maintenant l'enregistrement dans le fichier de zone direct :

#### Code : Console

```
vi /var/cache/bind/sdz.tp.hosts
```

#### Code : Console

```
sdz IN A 192.168.21.6
```



Qu'est-ce que ça veut dire ?

À faire correspondre un nom d'hôte à une adresse IPv4 de 32 bits distribués sur quatre octets (exemple : 192.168.21.2).

Il ne reste qu'à configurer les clients pour leur indiquer l'adresse du serveur DNS.

Sur chaque client, ouvrez le fichier resolv.conf :

**Code : Console**

```
Vi etc/resolv.conf
```

Et complétez le avec ces lignes :

**Code : Console**

```
Search sdz.tp. //Le client cherchera le nom de domai  
Nameserver 192.168.21.1 //L'adresse IP du serveur DNS (la ma
```

Pas mal, non ? 😊

## Mise en place du serveur Apache

Voilà, nous y sommes ! 😊

C'est dans cette partie que vous allez mettre en place notre serveur HTTP apache et le tester !

Au travail !

### *Installation du paquet apache2 sous Linux*

La première étape est l'installation du paquet apache2. Le paquet ? 🤖 Oui, sous Linux, pour ajouter un composant supplémentaire, il faut installer "son paquet" correspondant, et dans notre cas le paquet à installer s'appelle "apache2".

Pour ce faire, ouvrez une console et tapez ceci :

**Code : Console**

```
sudo apt-get install apache2
```

Quelques explications :

- **sudo** est un mot-clé qui permet de faire des manipulations en mode root c'est à dire en mode "administrateur" ;
- **apt-get install** est la commande par défaut pour installer un paquet ;
- **apache2** est le nom du paquet que je souhaite installer.



Note : Si le système vous pose des questions, répondez "yes" pour passer à l'étape suivante.



Si vous obtenez une erreur, tapez cette commande : `sudo apt-get update`. Celle-ci aura pour but de mettre à jour la liste des paquets logiciels disponibles pour votre environnement Linux.



Vous pouvez également installer le paquet **apache2** en passant par l'interface graphique. Si vous utilisez Ubuntu, rendez-vous dans le menu **Système** puis **Gestionnaire de paquet Synaptic** et recherchez le paquet désiré.

Vous venez maintenant d'installer le paquet **apache2** qui contient tous les fichiers de configuration du serveur. Il ne reste plus qu'à le configurer ! Enfin, plus que... façon de parler quoi. 😊

### *Création d'une page HTML de test*

Avant de passer à la configuration du serveur apache, nous allons créer une mini page web en HTML qui sera la page d'accueil de notre serveur Apache.

Créons tout d'abord le dossier qui contiendra cette page :

#### **Code : Console**

```
mkdir /var/www/sdz
```



La commande **mkdir** permet de créer un dossier sous Linux.

On se déplace dans ce dossier :

#### **Code : Console**

```
cd /var/www/sdz
```



La commande **cd** permet de se déplacer dans un dossier sous Linux.

À l'intérieur de ce dossier, on crée une page HTML nommée **index.html** :

#### **Code : Console**

```
vi index.html
```

Cette commande crée le fichier **index.html**, et nous ouvre le fichier vide. Ajoutons maintenant ces quelques lignes :

#### **Code : HTML**

```
<html>
<head> <title> Notre serveur Apache </title> </head>
<body> <p> Voici la page d'accueil de votre serveur Apache ! </p>
</body>
</html>
```

Enregistrez votre travail, et voilà, nous possédons maintenant une page web pour notre serveur apache. Passons désormais à la configuration du serveur. Enfin !

### *Configuration du serveur Apache*

Nous allons aborder deux exemples de configuration du serveur.

1er exemple : Configuration du serveur apache sans restrictions d'accès c'est à dire sans mot de passe. Dans ce cas, tout le monde peut accéder à notre serveur.

2ème exemple : Configuration du serveur apache avec authentification basique des utilisateurs en respectant simultanément les règles suivantes : accès permis seulement à partir de la machine salon identifiée par son adresse IP ou son nom et accès par mot de passe pour les utilisateurs admin et master.

### 1er exemple :

Déplaçons-nous dans le dossier apache2 :

#### Code : Console

```
cd /etc/apache2/sites-available
```

Créons dans ce dossier le fichier **sdz**. Ce fichier contiendra la configuration du serveur apache pour le nom de domaine **sdz**.

#### Code : Console

```
vi sdz
```

Et insérez à l'intérieur les quelques lignes suivantes :

#### Code : Console

```
NameVirtualHost 192.168.21.6:80          //Indiquez ici l'adresse IP de la machine
<VirtualHost 192.168.21.6 :80>           //Indiquez également ici l'adresse IP de la
ServerName sdz.truc.tp                   //Cette ligne est optionnelle. Vous pouvez i
DocumentRoot /var/www/sdz                //Indiquez ici le chemin de l'accès au fichi
<Directory /var/www/sdz>                 //Indiquez ici le chemin du répertoire apach
    Order allow,deny                     //Cette ligne représente les différentes opt
    Allow from all                       //Cette ligne permet l'accès au serveur à tous
</Directory>
</VirtualHost>
```

Maintenant que vous avez créé le fichier de configuration dans le dossier **sites-available**, il est nécessaire "d'activer" ce domaine en créant un lien symbolique dans le dossier **sites-enabled** (qui est le dossier contenant les sites dits "actifs") :

#### Code : Console

```
sudo ln -s /etc/apache2/sites-available/sdz /etc/apache2/sites-
enabled/sdz
```

Afin de prendre en compte votre configuration, redémarrez le serveur apache.

#### Code : Console

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

Vous pouvez désormais tester l'accès à votre serveur apache en tapant dans un navigateur web (Internet Explorer ou Mozilla Firefox) :



<http://192.168.21.6>

ou (si vous avez configuré un serveur DNS)

<http://sdz.truc.tp>



Afin de suivre les connexions au serveur apache et avoir des renseignements sur les erreurs liées à son utilisation, on peut utiliser les commandes suivantes.

**Code : Console**

```
tail -f /var/log/apache2/access.log
```

ou encore

**Code : Console**

```
tail -f /var/log/apache2/error.log
```

Ceci termine notre premier exemple. Comme vous le voyez, dans ce cas tout le monde peut accéder à notre page.

**2ème exemple :**

Intéressons-nous maintenant au deuxième exemple, un peu plus intéressant en configurant l'accès au serveur apache qu'à partir d'une seule machine (salon) et en protégeant l'accès par mot de passe pour les utilisateurs admin et superadmin.

Les modifications se font au niveau du même fichier :

**Code : Console**

```
vi /etc/apache2/sites-available/sdz
```

Remplacez maintenant les lignes du fichier par celles-ci :

**Code : Console**

```
NameVirtualHost 192.168.21.6:80
<VirtualHost 192.168.21.6:80>
    ServerName sdz.sdz.tp
    DocumentRoot /var/www/sdz
    <Directory /var/www/sdz>
        Order allow,deny
        Allow from 192.168.21.1 //Cette ligne définit l'accès au serveur uniquement
                                //à la machine 192.168.21.1
        Authname "salon" //Indiquez sur cette ligne le nom de la machine q
        Authtype basic //Cette ligne indique que le système utilisera le
        AuthUserFile /var/www/sdz/.htpasswd //Il s'agit du fichier qui sera utili
        Require valid-user //Cette ligne indique que pour que la connexion s
        Satisfy all
    </Directory>
</VirtualHost>
```

Enregistrez et quittez le fichier. Passons maintenant à la configuration du fichier **.htaccess**.

**Code : Console**

```
Authname « Accès protégé » //Cette ligne permettra d'afficher Accès protégé lors de
AuthType Basic
AuthUserFile /var/www/sdz/htpasswd //Cette ligne indique le chemin du fichier qu
Require valid-user
```

Pour finir, créons l'accès pour les deux utilisateurs admin et superadmin. Tapez simplement dans une terminal les lignes suivantes :

**Code : Console**

```
htpasswd -c -m htpasswd admin
htpasswd -m htpasswd superadmin
```

Enfin, afin de prendre en compte votre configuration, redémarrez le serveur apache.

**Code : Console**

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

Voilà, vous pouvez tester l'accès à votre serveur apache. Normalement, vous ne devriez pouvoir y accéder qu'à partir de la machine "salon" identifiée par l'adresse IP 192.168.21.1 et en utilisant un des deux utilisateurs admin ou superadmin.

Ceci termine la partie concernant la configuration d'un serveur HTTP apache.

Passons maintenant aux différents types d'hébergements possibles pour un serveur apache.

## Hébergement virtuel

### *Hébergement virtuel par nom de domaine*

**Ce type d'hébergement est utilisé pour accueillir plus d'un nom de domaine sur le même ordinateur sur la même adresse IP.**

Afin de tester ce type d'hébergement virtuel, chaque machine doit être recensée sous plusieurs noms différents sur le DNS en utilisant les enregistrements de type A et CNAME.

Maintenant que la configuration DNS n'a plus de secret pour vous, rajouter dans le fichier hosts du domaine (/var/cache/bind/sdz.truc.hosts), les informations suivantes :

**Code : Console**

```
sdzVH1 IN CNAME sdz
sdzVH2 IN CNAME sdz
```

Créons les trois fichiers index.html contenant comme informations le nom de domaine du serveur virtuel et plaçons-les dans les répertoires correspondant :

**Code : Console**

```
mkdir /var/www/sdzVH1
mkdir /var/www/sdzVH2
```

À l'intérieur de chaque dossier, on crée une page HTML nommée index.html :

**Code : HTML**

```
<html>
<head> <title> Notre serveur Apache </title> </head>
<body> <p> Voici la page d'accueil de votre serveur Apache sdzVH1 !
</p> </body>
</html>
```

**Code : HTML**

```
<html>
<head> <title> Notre serveur Apache </title> </head>
<body> <p> Voici la page d'accueil de votre serveur Apache sdzVH2 !
</p> </body>
</html>
```

On crée deux fichiers de configuration pour nos deux nouveaux domaines dans le dossier /etc/apache2/sites-available afin de créer ces deux serveurs virtuels pointant sur deux répertoires différents.

**Code : Console**

```
vi /etc/apache2/sites-available/sdzVH1
```

On ajoute dans ce fichier les lignes suivantes :

**Code : Console**

```
NameVirtualHost 192.168.21.6:80
<VirtualHost 192.168.21.6:80>
    ServerName sdzVH1.truc.tp
    ServerAlias sdzVH1
    DocumentRoot /var/www/sdzVH1
    <Directory /var/www/sdzVH1>
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

De même pour l'autre domaine :

**Code : Console**

```
vi /etc/apache2/sites-available/sdzVH2
```

On ajoute dans ce fichier les lignes suivantes :

**Code : Console**

```
NameVirtualHost 192.168.21.6:80
<VirtualHost 192.168.21.6:80>
    ServerName sdzVH2.truc.tp
    ServerAlias sdzVH2
    DocumentRoot /var/www/sdzVH2
    <Directory /var/www/sdzVH2>
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

C'est l'heure d'activer nos deux hébergements virtuels. Pour ce faire, il faut créer des liens symboliques dans le dossier /etc/apache2/sites-enabled (qui recensent les sites actifs) pour que la configuration soit lue lors du démarrage d'apache.

**Code : Console**

```
sudo ln -s /etc/apache2/sites-available/sdzVH1 /etc/apache2/sites-
enabled/sdzVH1
sudo ln -s /etc/apache2/sites-available/sdzVH2 /etc/apache2/sites-
enabled/sdzVH2
```



Redémarrez maintenant le serveur apache !

**Code : Console**

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

Il ne vous reste plus qu'à tester. Vous devriez pouvoir accéder à chaque page web correspondante à chaque nom de domaine.

- <http://sdz.truc.tp>
- <http://sdzVH1.truc.tp>
- <http://sdzVH2.truc.tp>



Voilà, vous voyez l'intérêt de ce type d'hébergement. C'est que tout en ne possédant qu'une seule adresse IP, il est possible de définir plusieurs sous-domaines pour gérer différentes pages web. C'est ti pas beau tout ça ? 😊

### *Hébergement virtuel par port*

Ce type d'hébergement permet d'utiliser un serveur web en précisant le port qu'il utilise. Si le port n'est pas précisé, la connexion ne s'effectuera pas ! 😞

Dans ce cas, chaque serveur sera adressé par une requête HTTP à l'aide d'un numéro de port différent. Par exemple pour la machine "salon" :

```
http://sdz.truc.tp:80
http://sdzVH1.truc.tp:40001
http://sdzVH2.truc.tp:40002
```

Afin que apache écoute sur les ports correspondants, on les ajoute dans le fichier ports.conf :

**Code : Console**

```
/etc/apache2/ports.conf
```

**Code : Console**

```
Listen 80
Listen 40001
Listen 40002
```

Effectuons maintenant les modifications dans chaque fichier de configuration pour nos différents domaine (que vous connaissez maintenant par cœur) :

**Code : Console**

```
vi /etc/apache2/sites-available/sdz
```

**Code : Console**

```
NameVirtualHost 192.168.21.6:80
<VirtualHost 192.168.21.6:80>
    ServerName sdz.truc.tp
    ServerAlias sdz
    DocumentRoot /var/www/sdz
    <Directory /var/www/sdz>
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Puis dans le fichier de configuration du domaine sdzVH1 :

**Code : Console**

```
NameVirtualHost 192.168.21.6:80
<VirtualHost 192.168.21.6:40001>
    ServerName sdzVH1.truc.tp
    ServerAlias sdzVH1
    DocumentRoot /var/www/sdzVH1
    <Directory /var/www/sdzVH1>
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Puis dans le fichier de configuration du domaine sdzVH2 :

**Code : Console**

```
NameVirtualHost 192.168.21.6:80
<VirtualHost 192.168.21.6:40002>
    ServerName sdzVH2.truc.tp
```

```
ServerAlias sdzVH2
DocumentRoot /var/www/sdzVH2
<Directory /var/www/sdzVH2>
    Order allow,deny
    Allow from all
</Directory>
</VirtualHost>
```

C'est fini ! Il ne vous reste plus qu'à tester :

- <http://sdz.truc.tp:80>
- <http://sdzVH1.truc.tp:40001>
- <http://sdzVH2.truc.tp:40002>

### *Hébergement virtuel par adresse IP*

Ce type d'hébergement permet, tout en associant deux adresses IP à une interface, de se connecter aux pages associées à chaque adresse IP.

Pour démarrer, on doit créer un alias `eth0:1` sur l'interface `eth0` afin d'associer 2 adresses ip à une seule interface. On ajoute pour cela les lignes suivantes dans le fichier `/etc/network/interfaces` :

#### **Code : Console**

```
Auto eth0 :1
Iface eth0 :1 inet static
Address 192.168.21.106
Netmask 255.255.255.0
Broadcast 192.168.21.255
```

On modifie notre fichier adoré ! 😊

Pour le domaine `sdz` :

#### **Code : Console**

```
vi /etc/apache2/sites-available/sdz
```

#### **Code : Console**

```
NameVirtualHost 192.168.21.6:80
<VirtualHost 192.168.21.106:80> //On indique simplement que pour accéder au domaine
    ServerName sdz.truc.tp
    ServerAlias sdz
    DocumentRoot /var/www/sdz
    <Directory /var/www/sdz>
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Vous pouvez faire la même chose pour les domaines sdzVH1 et sdzVH2 si vous le souhaitez. 🤔

😊 Il ne reste qu'à apprécier votre travail, à vous lancer des roses, des fleurs, et à dérouler le tapis rouge à vos pieds ! 😊

- <http://192.168.21.6:80>
- <http://192.168.21.106:80>

Eh bien voilà ! Je crois que vous avez fait le tour de ce qu'il faut savoir sur les serveurs Apache et DNS.

N'hésitez pas à partager vos expériences !

@wawanopoulos

**Partager**

