

Abel

SIO26

Installation d'Apache avec Ubuntu Server



Table des matières

Prérequis	3
Installer Apache2	3
Configuration de 2 vHosts différents	4
Configuration du vHost par défaut.....	5
Configuration du second vHost	5
Créer le répertoire du deuxième vHost	6
Activer le site	7
Configuration du fichier Hosts de Windows	7

Prérequis

- Utilisation de **VMWare** pour cette documentation
- Installation d'**Apache** sur une VM **Ubuntu Server 24.04.1** :
 - o 2GB de ram
 - o 10GB de stockage
 - o 2 cœurs est préférable

Installer Apache2

Pour commencer, avant de pouvoir démarrer un vHost ou un site, il nous faut Apache sur notre Ubuntu Server.

Pour cela, nous allons écrire la commande suivante :

```
ubuntu@shadowserv:~$ sudo apt update
```

Écrire le mot de passe du compte afin de valider l'installation.

Cette commande permet de mettre à jour tous les services qui ne possède pas la dernière version sur notre serveur, afin d'éviter quelques soucis de sécurité.

À présent, nous allons installer Apache avec cette commande :

```
ubuntu@shadowserv:~$ sudo apt install apache2
```

Écrire le mot de passe et il n'y aura plus qu'à écrire « **y** » afin de valider l'installation.

Une fois l'installation terminée, nous allons vérifier que le service d'Apache soit bien en ligne.

Pour ce faire, nous allons taper la commande suivante : `systemctl status apache2`

```
ubuntu@shadowserv:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-01-31 10:51:59 UTC; 17min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 16670 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 16674 (apache2)
    Tasks: 55 (limit: 2218)
   Memory: 6.1M (peak: 6.6M)
      CPU: 100ms
   CGroup: /system.slice/apache2.service
           └─16674 /usr/sbin/apache2 -k start
             └─16676 /usr/sbin/apache2 -k start
               └─16677 /usr/sbin/apache2 -k start

Jan 31 10:51:59 shadowserv systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jan 31 10:51:59 shadowserv apachectl[16673]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1:80
Jan 31 10:51:59 shadowserv systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-17/17 (END)
```

Bien vérifier que ce soit marqué « **active (running)** » et surtout « **enabled** ».

Configuration de 2 vHosts différents

À présent, on va configurer deux vHost différents afin d'avoir deux sites web fonctionnels sur le même serveur.

Tout d'abord, nous allons vérifier si la page de base d'Apache est bien accessible.

Pour cela, nous allons regarder l'ip de la machine afin d'accéder à la page.

Taper la commande **ip a** :

```
ubuntu@shadowserv:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 00:0c:29:fe:4b:d3 brd ff:ff:ff:ff:ff:ff
   altname enp2s1
   inet 192.168.136.133/24 metric 100 brd 192.168.136.255 scope global dynamic ens33
       valid_lft 1194sec preferred_lft 1194sec
   inet6 fe80::20c:29ff:fe4b:d3/64 scope link
       valid_lft forever preferred_lft forever
ubuntu@shadowserv:~$
```

Nous allons regarder la ligne **2** et regarde la ligne « **inet** » pour récupérer l'adresse attribuée à la machine qui est ici : **192.168.136.133**

Maintenant, nous allons ouvrir un onglet sur un navigateur quelconque. Ici, nous allons utiliser **Google Chrome**.

Sur la barre d'**URL**, écrire l'ip qu'on a récupéré au-dessus, et rajouter « **:80** » comme ceci : **192.168.136.133 :80**

Cette page devrait s'afficher :



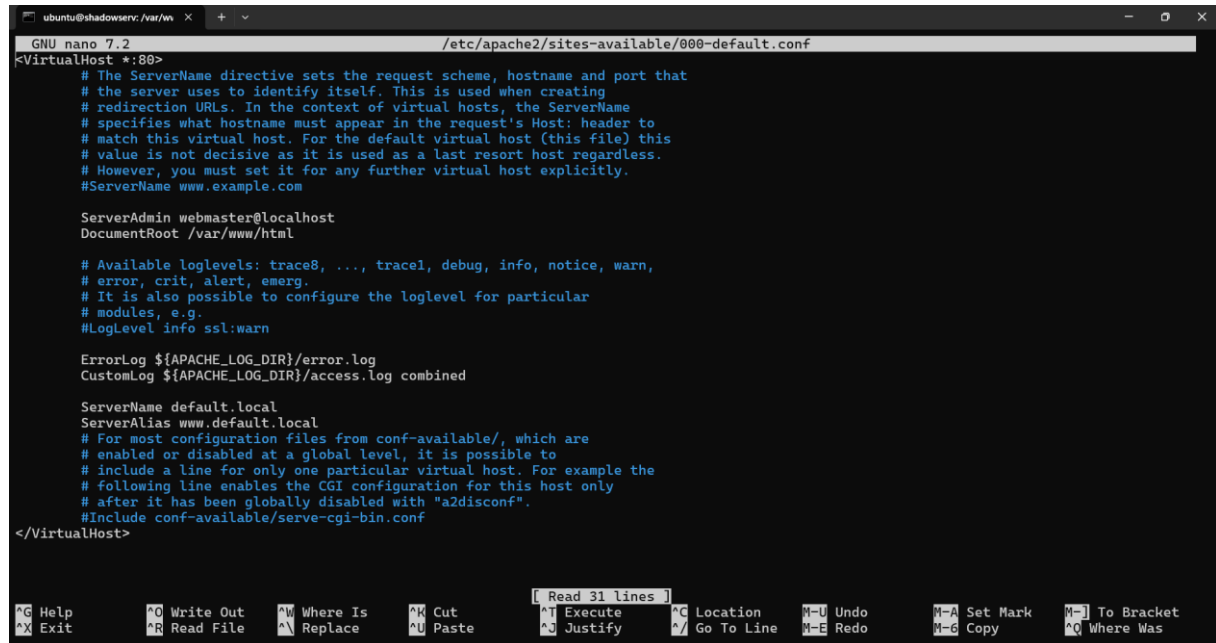
Configuration du vHost par défaut

Le fichier de configuration du vHost par défaut d'Apache se trouve dans « **/etc/apache2/sites-available/** » le fichier se nommant « **000-default.conf** », pour le modifier nous allons taper cette commande qui suit :

```
ubuntu@shadowserv:~$ sudo nano /etc/apache2/sites-available/000-default.conf
```

Et taper votre mot de passe.

Nous nous retrouvons donc sur cette page :



```
GNU nano 7.2 /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

#
# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Nous allons y rajouter les lignes « **ServerName** » et « **ServerAlias** » afin de pouvoir accéder au site directement, ainsi les deux sites auront la même IP, cependant le serveur saura vous rediriger précisément.

Pour enregistrer le fichier, appuyer sur la combinaison de touche « **CTRL+X** » puis « **y** » et appuyer sur « **Entrée** ».

Configuration du second vHost

Nous allons créer un autre fichier dans le même répertoire afin d'avoir notre deuxième site hébergé sur la même machine.

Pour ceci, nous allons taper la même commande qu'au-dessus mais en modifiant la fin, pour avoir le nom de fichier qu'on souhaite pour notre deuxième site.

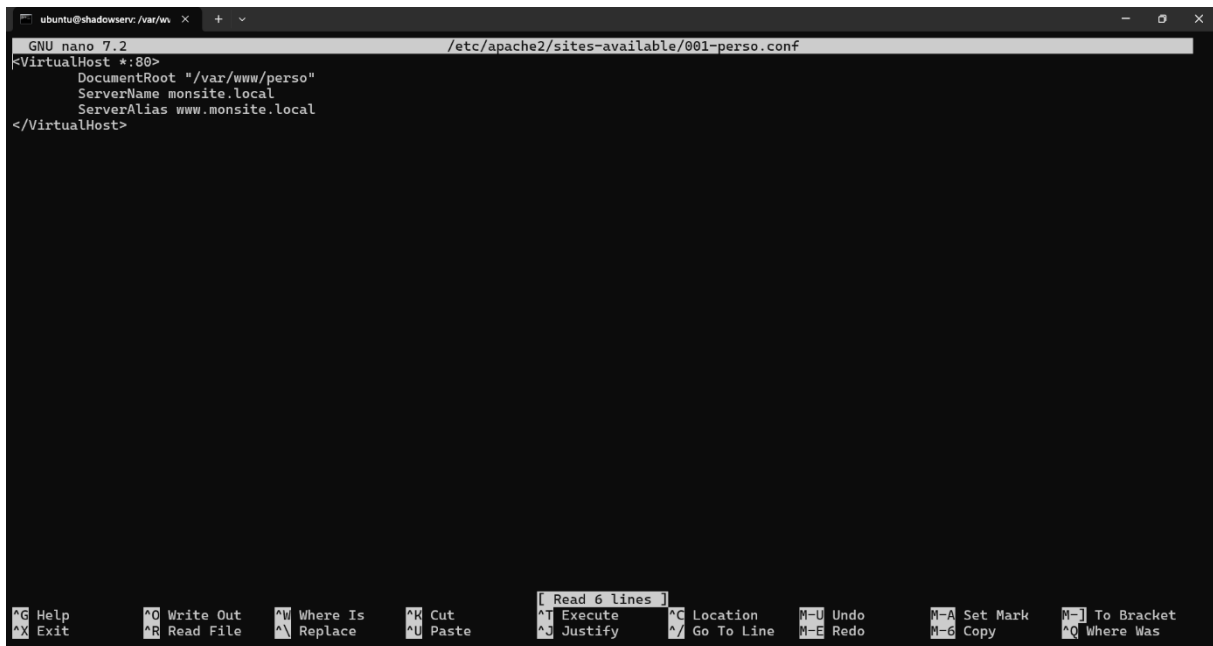
Par exemple, comme cela :

```
ubuntu@shadowserv:~$ sudo nano /etc/apache2/sites-available/001-perso.conf
```

(Le fichier peut être nommé autrement que « **001-perso** » mais doit finir par « **.conf** ».)

Écrire son mot de passe si nécessaire.

Sinon, écrire dans le fichier comme ci-dessous :



```
ubuntu@shadowserv: /var/www
GNU nano 7.2 /etc/apache2/sites-available/001-perso.conf
<VirtualHost *:80>
  DocumentRoot "/var/www/perso"
  ServerName monsite.local
  ServerAlias www.monsite.local
</VirtualHost>

[ Read 6 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        [ Read 6 lines ]  ^C Location   M-U Undo      M-A Set Mark   M-J To Bracket
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^T Execute      ^_ Go To Line  M-E Redo      M-G Copy       ^Q Where Was
```

Pour « **DocumentRoot** », vous pouvez nommer le répertoire comme vous le souhaitez.

Donc « **/var/www/<NOM DU RÉPERTOIRE>** »

Personnaliser « **ServerName** » et « **ServerAlias** » comme vous souhaitez pour pouvoir accéder à votre site.

Fermez et enregistrez le fichier comme plus haut : combinaison de touche « **CTRL+X** » puis « **y** » et appuyez sur « **Entrée** ».

Créer le répertoire du deuxième vHost

Pour ceci, nous allons taper la commande suivante :

```
ubuntu@shadowserv:~$ sudo mkdir /var/www/perso/
```

(Si vous avez donné un autre nom de répertoire dans l'étape précédente, mettre ce nom à la place de « **perso** »)

Taper votre mot de passe afin de pouvoir créer ce répertoire.

Activer le site

Maintenant que le fichier est créé, il faut activer le vHost soit avec une commande, soit en créant un lien symbolique du fichier vers le répertoire :

« **/etc/apache2/sites-enabled** »

- Pour le faire en lien symbolique :

`sudo ln -sf /etc/apache2/sites-available/001-perso.conf /etc/apache2/sites-enabled/`

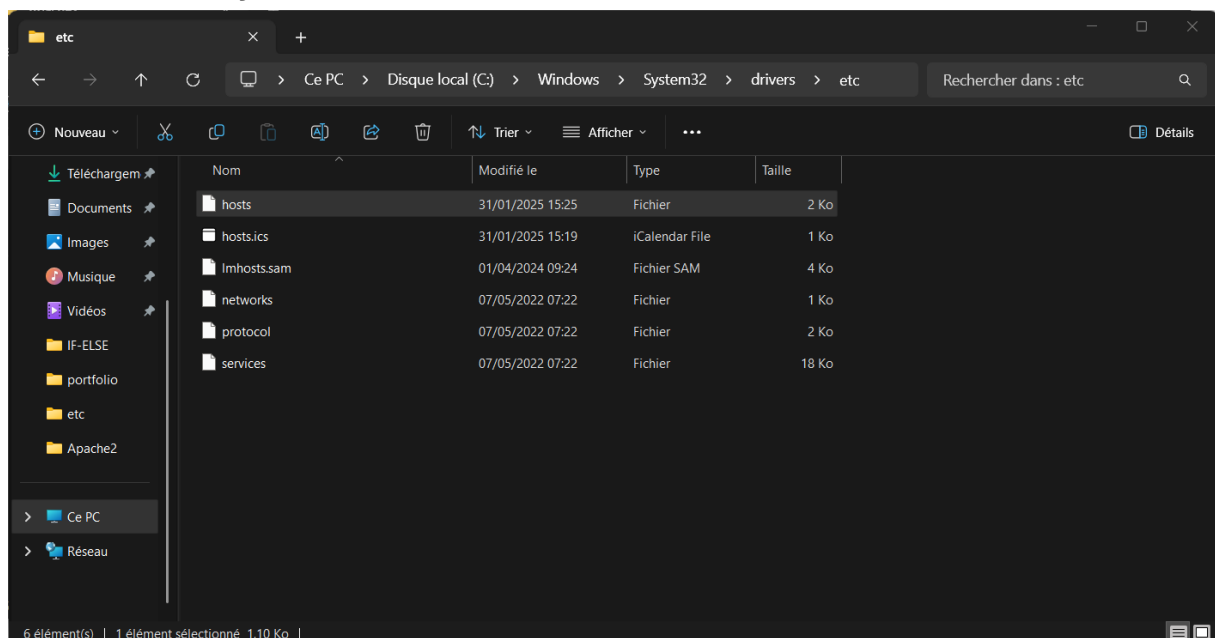
- Avec une commande d'Apache directement : **`sudo a2ensite 001-perso.conf`**

° **Note :** Vous pouvez aussi désactiver le site par défaut d'Apache avec la commande : **`sudo a2dissite 000-default.conf`**

Configuration du fichier Hosts de Windows

Désormais, nous allons configurer le fichier « **Hosts** » qui se trouve dans

« **C:/Windows/System32/drivers/etc/** »



Afin de pouvoir modifier ce fichier, il faut l'exécuter en tant qu'Administrateur.

Rajouter ces lignes suivantes en tenant compte de ce que vous avez écrit dans vos fichiers :



```
C: > Windows > System32 > drivers > etc > $ hosts
1  # Copyright (c) 1993-2009 Microsoft Corp.
2  #
3  # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4  #
5  # This file contains the mappings of IP addresses to host names. Each
6  # entry should be kept on an individual line. The IP address should
7  # be placed in the first column followed by the corresponding host name.
8  # The IP address and the host name should be separated by at least one
9  # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #         102.54.94.97       rhino.acme.com   # source server
17 #         38.25.63.10      x.acme.com       # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1       localhost
21 #   ::1            localhost
22
23 #Port 12753 #Samsung Analytics
24 192.168.136.133 www.monsite.local monsite.local www.default.local default.local
25
```

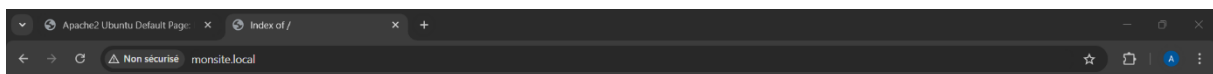
Ajouter la ligne 24 de l'image ci-dessus et rajouter les noms de serveurs que vous avez configuré dans les fichiers de vos sites.

Ensuite, taper la commande suivante afin de redémarrer Apache :

systemctl restart apache2

Maintenant, nous allons vérifier que tous ce qui a été fait a bien été enregistré et que ce soit fonctionnel.

Aller sur un navigateur et taper les alias que vous avez créé pour vérifier que vous avez ces pages ci-dessous :



Pour le deuxième site, il est normal d'avoir un « **Index of /** » car nous n'avons pas créé de page « **index.html** » dans le répertoire « **/var/www/perso** ».