

# Configuration de serveur web

Hasna Daoui

January 2025

## 1 Mise en Place d'un Serveur Web Hors Connexion Internet

Un serveur web hors connexion Internet peut être utilisé dans des environnements isolés, tels que des réseaux internes (Intranet), des laboratoires ou des situations où l'accès à Internet n'est pas requis. Ce type de serveur est idéal pour fournir des services web, comme des portails internes, des bases de données locales, ou des applications collaboratives.

### Objectif

L'objectif de cette configuration est de déployer un serveur web fonctionnel capable de répondre aux requêtes des clients internes sans dépendre d'une connexion Internet. Cela permet d'héberger et de gérer des ressources web localement tout en assurant une meilleure sécurité et un contrôle total sur les données.

### Environnement

Le serveur web sera configuré en utilisant un logiciel tel qu'Apache ou Nginx. Les clients du réseau local pourront accéder aux services hébergés en utilisant l'adresse IP ou le nom de domaine local attribué au serveur.

### Avantages

- **Indépendance** : Aucun besoin de connexion Internet pour les services web internes.
- **Sécurité** : Élimine les risques liés à l'exposition directe sur Internet.
- **Performance** : Réponse rapide grâce à l'absence de latence liée à un réseau externe.

Les prochaines sections détailleront les étapes pratiques pour installer et configurer un serveur web Apache hors connexion.

## 2 Configurer le serveur web pour le domaine

Créez un fichier de configuration pour le site :

```
sudo nano /etc/apache2/sites-available/eidia.conf
```

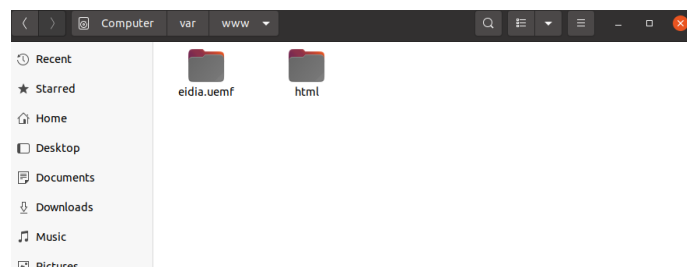


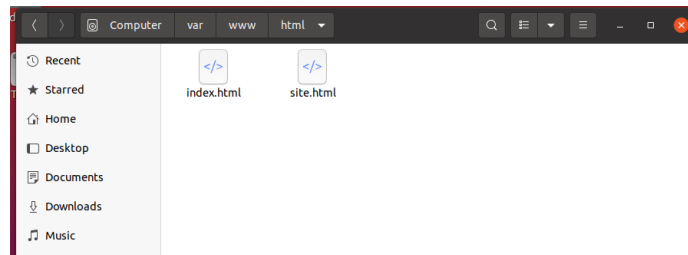
```
1<VirtualHost *:80>
2    ServerName www.eidia.uemf
3    DocumentRoot /var/www/html
4
5    # Specify the custom DirectoryIndex
6    DirectoryIndex site.html
7
8    <Directory /var/www/html>
9        AllowOverride All
10       Require all granted
11    </Directory>
12
13    ErrorLog ${APACHE_LOG_DIR}/eidia_error.log
14    CustomLog ${APACHE_LOG_DIR}/eidia_access.log combined
15</VirtualHost>
```

le **site.html** est le fichier qui conteint le code de website. qu'ete creer dans /etc/var/www/html avec commande:

```
cd /etc/var/www/html
sudo gedit site.html
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome to EIDIA.UEMF</title>
</head>
<body>
    <h1>Hello, welcome to www.eidia.uemf!</h1>
</body>
</html>
```





Le fichier `index.html` est la page d'accueil par défaut servie par Apache lorsqu'aucun fichier spécifique n'est demandé dans une URL. Il contient le contenu principal du site web et peut être personnalisé ou remplacé via la directive `DirectoryIndex`.

### 3 Activation de site et recharger Apache :

```
sudo a2ensite eidia.  
conf sudo systemctl reload apache2
```

#### Explication des commandes

- `sudo chown -R www-data:www-data /var/www/html` : Change le propriétaire et le groupe du répertoire `/var/www/html` et de tout son contenu de manière récursive (`-R`). Le répertoire est attribué à `www-data`, l'utilisateur et le groupe par défaut utilisés par Apache pour accéder aux fichiers.
- `sudo chmod -R 755 /var/www/html` : Modifie les permissions du répertoire `/var/www/html` et de tout son contenu de manière récursive (`-R`). Les permissions 755 signifient que :
  - Le propriétaire (`www-data`) peut lire, écrire et exécuter.
  - Les autres utilisateurs peuvent seulement lire et exécuter.
- `sudo ufw allow 80` : Autorise les connexions entrantes sur le port 80 via le pare-feu UFW. Ce port est utilisé pour les connexions HTTP non sécurisées.
- `sudo ufw allow 443` : Autorise les connexions entrantes sur le port 443 via le pare-feu UFW. Ce port est utilisé pour les connexions HTTPS sécurisées.
- `sudo ufw reload` : Recharge la configuration du pare-feu UFW pour appliquer les nouvelles règles (autorisations des ports 80 et 443).

```

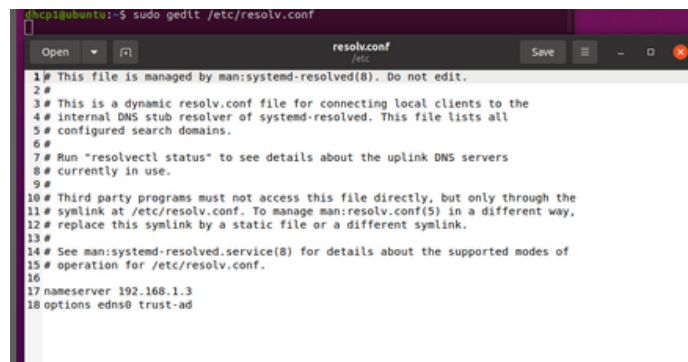
dhcp1@ubuntu:~$ sudo chown -R www-data:www-data /var/www/html
dhcp1@ubuntu:~$ sudo chmod -R 755 /var/www/html
dhcp1@ubuntu:~$ sudo ufw allow 80
Skipping adding existing rule
Skipping adding existing rule (v6)
dhcp1@ubuntu:~$ sudo ufw allow 443
Skipping adding existing rule
Skipping adding existing rule (v6)
dhcp1@ubuntu:~$ sudo ufw reload
Firewall not enabled (skipping reload)
dhcp1@ubuntu:~$ sudo nano /etc/apache2/sites-available/eidia.conf

```

En résumé, ces commandes préparent le répertoire `/var/www/html` pour être utilisé par Apache et configurent le pare-feu pour permettre les connexions HTTP et HTTPS.

En change le ip de server utiliser, en ajoutant l'adresse IP de serveur DNS dans `/etc/resolv.conf` :

nameserver 192.168.1.3 (adresse ip de server dhcp)



```

dhcp1@ubuntu:~$ sudo gedit /etc/resolv.conf
resolv.conf
/etc/
1# This file is managed by man:systemd-resolved(8). Do not edit.
2#
3# This is a dynamic resolv.conf file for connecting local clients to the
4# internal DNS stub resolver of systemd-resolved. This file lists all
5# configured search domains.
6#
7# Run "resolvectl status" to see details about the uplink DNS servers
8# currently in use.
9#
10# Third party programs must not access this file directly, but only through the
11# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
12# replace this symlink by a static file or a different symlink.
13#
14# See man:systemd-resolved.service(8) for details about the supported modes of
15# operation for /etc/resolv.conf.
16
17nameserver 192.168.1.3
18options edns0 trust-ad

```

## 4 Tester

dans le client(Apres changer dans `/etc/resolv.conf` le IP address de serveur au IP de dhcp server 192.168.1.3) ou DHCP serveur en tape le site **http://www.eidia.uemf** sur firefox après en avoir ce site la.



OU bien:

```
machine3@ubuntu:~$ nslookup www.eidia.uemf
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   www.eidia.uemf
Address: 192.168.1.3
```