tutoriel

Configuration des hôtes virtuels Apache sous Ubuntu (20.04)

La configuration du serveur Apache se fait par une approche modulaire qui permet d'ajouter et de modifier de nouveaux fichiers selon les besoins, et non dans un seul fichier monolithique. Cette conception modulaire permet de créer un site ou un domaine individuel appelé hôte virtuel.

Les hôtes virtuels permettent à une même instance d'Apache de servir plusieurs sites web. Chaque domaine ou site individuel configuré avec Apache renvoie le visiteur vers un répertoire contenant les données du site en question. Le visiteur n'est pas averti que le même serveur sert aussi d'autres sites. Ce schéma n'a aucune limite tant que le serveur peut supporter la charge.

Ce guide explique comment configurer des hôtes virtuels Apache sur un serveur Ubuntu 20.04. Vous apprendrez comment servir des contenus différents à des visiteurs différents selon les domaines demandés en créant deux hôtes virtuels.

Nous supposons que :

- **nous possédons les domaines exemple.org et test.org** (noms de domaine pour l'accès en production)
- **mais pas exemple.loc ni test.loc** (noms de domaine pour l'accès en local)

Nous allons créer un hôte virtuel pour example.com et un autre pour test.com. Il y sera fait référence tout au long du guide, mais vous devrez y substituer vos propres domaines ou valeurs tout en suivant le guide.

Nous allons créer un hôte virtuel avec le nom de domaine monsite.loc, qui pointera sur le site web en local.

Pré-requis

- Un serveur Ubuntu 20.04 doté d'un utilisateur non root avec les privilèges sudo. Vous pouvez utiliser notre guide **Configuration initiale du serveur avec Ubuntu 20.04** pour configurer ce serveur.
- **Apache ou un environnement LAMP** (Linux/Apache/MySQL/PHP) ou équivalent installé sur le serveur

Première étape : Création de la structure des répertoires

Chacun des répertoires (nommés ici exemple.org et test.org) contiendra les fichiers du site web.

1. **créez les répertoires des sites web et attribuez les permissions** : La racine (root) de chaque site (le répertoire qu'Apache consulte pour trouver le contenu à servir) sera placée dans des sous-répertoires du répertoire /var/www/html :

```
USER@MACHINE:~$ sudo mkdir -p /var/www/html/{exemple,test}.org
USER@MACHINE:~$ sudo chown -R www-data:www-data /var/www/html
USER@MACHINE:~$ sudo chmod -R 2755 /var/www/html
```

```
/var/www/html
├── exemple.org
└── test.org
```

Les dossiers des sites se nomment **exemple.org** et **test.org** (noms de domaine pour l'accès en production) et non pas exemple.loc et test.loc (noms de domaine pour l'accès en local).

Pour faciliter le nommage du dossier projet et des fichiers de configuration, c'est le nom de domaine de production qui est utilisé.

Cette pratique de nommage n'est pas une obligation, mais reste toutefois une bonne pratique.

En modifiant le fichier /etc/hosts

Éditez avec les droits d'administration le fichier /etc/hosts

Chaque ligne définit l'association entre une adresse IP et un nom de domaine, plusieurs noms de domaine peuvent avoir la même adresse IP. Par exemple :

/etc/hosts

```
# site web local exemple
127.0.0.1 exemple.loc

# Autres sites web locaux
127.0.0.1 mon-projet-personnel.loc
127.0.0.1 un-autre-site-web-
local.loc
127.0.0.1 test.loc
```

Avec un serveur DNS

Autres étapes

1. **Créez des pages HTML pour chaque hôte virtuel** informant les visiteurs du site auquel ils sont connectés :
    1. **exemple.org** :

```
USER@MACHINE:~$ echo "Bienvenue sur exemple.org !" | sudo tee /var/www/html/exemple.org/index.html
```

    2. **test.org** :

```
USER@MACHINE:~$ echo "Bienvenue sur test.org !" | sudo tee /var/www/html/test.org/index.html
```

    3. **Rétablissez les droits** :

```
USER@MACHINE:~$ sudo chown -R www-data:www-data /var/www/html ; sudo chmod -R 2755 /var/www/html
```

2. **Créez les fichiers d'hôtes virtuels** : Les fichiers d'hôtes virtuels spécifient la configuration des hôtes virtuels et indiquent comment le serveur web Apache répondra aux différentes requêtes.
Apache est livré avec un fichier d'hôte virtuel par défaut nommé /etc/apache2/sites-available/000-default.conf. Copiez ce fichier pour créer des fichiers d'hôtes virtuels pour chacun de vos domaines.
    1. **exemple.org** :
        1. **Copiez la configuration par défaut** :

```
USER@MACHINE:~$ sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/exemple.org.conf
```

Notez que le nom de domaine de production **exemple.org** est repris aussi bien pour le répertoire du site que pour le nom du fichier de configuration et que l'extension du fichier de configuration est **.conf**

        2. **Éditez avec les droits d'administration le fichier /etc/apache2/sites-available/exemple.org.conf** pour le personnaliser et ajouter quelques directives supplémentaires :
            - **ServerAdmin** : admin@exemple.org (adresse électronique de

l'administrateur du site)

- **ServerName** : exemple.org (domaine de base pour cet hôte virtuel)
- **ServerAlias** : www.exemple.org (autres noms qui doivent correspondre comme s'ils étaient le nom de base ; utile pour faire correspondre les hôtes que vous avez définis, comme www)
- **DocumentRoot** : /var/www/html/exemple.org (emplacement de la racine – root)

/etc/apache2/sites-available/exemple.org.conf

```
<VirtualHost *:80>
    # e-mail de l'administrateur
    ServerAdmin admin@exemple.org
    # Associer le nom de domaine au dossier du
projet
    ServerName exemple.loc
    ServerAlias www.exemple.loc
    DocumentRoot /var/www/html/exemple.org
    # fichiers journaux
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log
combined
    <directory /var/www/html/exemple.org>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Cette section d'hôte virtuel correspond à toutes les requêtes qui sont faites sur le port 80, le port HTTP par défaut. Ici, nous nous limitons au protocole http et non https.

2. **test.org** :
    1. **copiez l'hôte virtuel exemple.org** :

```
USER@MACHINE:~$ sudo cp /etc/apache2/sites-
available/exemple.org.conf /etc/apache2/sites-
available/test.org.conf
```

    2. **Éditez avec les droits d'administration le fichier /etc/apache2/sites-available/test.org.conf** pour le personnaliser :

/etc/apache2/sites-available/test.org.conf

```
<VirtualHost *:80>
    # e-mail de l'administrateur
```

```
    ServerAdmin admin@test.org
    # Associer le nom de domaine au dossier du
projet
    ServerName test.loc
    ServerAlias www.test.loc
    DocumentRoot /var/www/html/test.org
    # fichiers journaux
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log
combined
    <directory /var/www/html/test.org>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

3. **Activez les nouveaux hôtes virtuels** :

```
USER@MACHINE:~$ sudo a2ensite /etc/apache2/sites-
available/exemple.org.conf
Enabling site exemple.org.
To activate the new configuration, you need to run:
  systemctl reload apache2
USER@MACHINE:~$ sudo a2ensite /etc/apache2/sites-
available/test.org.conf
Enabling site test.org.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

si vous avez besoin de les désactiver, c'est :

```
USER@MACHINE:~$ sudo a2dissite
/etc/apache2/sites-
available/exemple.org.conf
USER@MACHINE:~$ sudo a2dissite
/etc/apache2/sites-
available/test.org.conf
```

4. **Vérifiez qu'il n'y a pas d'erreurs de configuration** :

```
USER@MACHINE:~$ sudo apache2ctl configtest
. . .
Syntax OK
```

5. **Redémarrez Apache et vérifiez le succès du redémarrage** :

```
USER@MACHINE:~$ sudo systemctl restart apache2
```

```
USER@MACHINE:~$ sudo systemctl status apache2
```

Le serveur est maintenant configuré pour servir deux sites web.

6. **(Facultatif) Modification du fichier local /etc/hosts** : Si vous ne possédez pas de noms de domaine réels pour vos serveurs, et que vous avez utilisé des domaines fictifs, vous pouvez tester la configuration pour votre ordinateur personnel en modifiant le fichier /etc/hosts sur votre ordinateur local. Cela interceptera les requêtes pour les domaines configurés et les dirigera vers votre serveur local, comme le ferait le système DNS si vous utilisiez des domaines enregistrés. Le contenu ne sera accessible que localement : les autres visiteurs ne pourront pas accéder au site depuis l'extérieur via le nom de domaine fictif.
    1. Éditez le fichier **/etc/hosts** pour y ajouter deux lignes :

       /etc/hosts

       ```
       ...
       127.0.0.1 exemple.org
       127.0.0.1 test.org
       ```

       Cela dirigera toute demande de exemple.org et test.org sur votre ordinateur en l'envoyant au serveur local. C’est ce que nous voulons si nous ne sommes pas réellement les propriétaires de ces domaines afin de tester nos hôtes virtuels.

7. **Testez vos résultats** :
    1. **Ouvrez http://exemple.org** → Bienvenue sur exemple.org !
    2. **Ouvrez http://test.org** → Bienvenue sur test.org !
8. Il ne reste plus qu'à installer les sites dans leurs emplacements.

Si vous avez modifié le fichier hosts de votre ordinateur local, vous pouvez supprimer les lignes ajoutées une fois vérifiée la configuration.

Vous éviterez ainsi de surcharger votre fichier hosts avec des entrées devenues inutiles.

Conclusion

Désormais, un seul serveur gère deux noms de domaine distincts.

Pour étendre ce processus, il suffit de suivre les mêmes étapes pour ajouter d'autres hôtes virtuels.

Il n'y a pas de limite au nombre de noms de domaine gérables par Apache, alors n'hésitez pas à créer autant d'hôtes virtuels que le serveur peut en supporter.

Problèmes connus

Voir aussi

- **(en)** How To Set Up Apache Virtual Hosts on Ubuntu 20.04
- **(fr)** Apache : Apprendre à créer des virtual Hosts

Basé sur « How To Set Up Apache Virtual Hosts on Ubuntu 20.04 » par Kong Yang et Lisa Tagliaferri.

From:
https://doc.wikis.frapp.fr/ - **doc**

Permanent link:
https://doc.wikis.frapp.fr/doku.php?id=tutoriel:reseau:http:serveur:apache:vhost:start

Last update: **2024/09/21 10:12**