

Virtual Hosts

Projet m2l.org



DUMAS Lucie

Table des matières

Qu'est-ce qu'un Virtual Host ?.....	3
Les serveurs web virtuels.....	4
Les fichiers .htaccess.....	11
Les répertoires personnels	11



Qu'est-ce qu'un Virtual Host ?

Un Virtual Host (ou hôte virtuel) est une fonctionnalité utilisée par les serveurs web pour héberger plusieurs sites web sur une seule machine physique, utilisant une seule adresse IP. Cela permet de maximiser l'utilisation des ressources du serveur et de faciliter la gestion de multiples sites.

Il existe deux principaux types de Virtual Hosts :

- Virtual Hosts basés sur le nom : Utilisent le nom de domaine pour différencier les sites hébergés. Le serveur web examine le champ Host de l'en-tête HTTP pour déterminer quel site doit répondre à la requête.
- Virtual Hosts basés sur l'adresse IP : Chaque site a une adresse IP unique. Ce type est moins courant mais peut être utile pour des configurations réseau spécifiques ou des besoins de sécurité.



Les serveurs web virtuels

Installation des Virtual Hosts

Dans cette mission, nous mettrons en place plusieurs sites web sur une même machine. Les noms de domaine des sites seront les suivants :

- www.m2l.org
- intranet.m2l.org
- extranet.m2l.org
- wiki.m2l.org

Pour que nos différents sites soient reconnus dans le DNS, nous devons dans un premier temps modifier notre fichier db.m2l.conf dans notre conteneur DNS, afin d'y insérer les Alias de nos nouveaux sites web :

```
@ IN SOA ns1.m2l.org. root.m2l.org. (
    2020122601;
    43200;
    3600;
    3600000;
    172800 );

@ IN A 10.31.96.80;
@ IN NS ns1.m2l.org.;
@ IN NS ns2.m2l.org.;

ns1 IN A 10.31.96.80;
ns2 IN A 10.31.96.54;

backup IN A 10.31.96.99;
www IN A 10.31.96.80;

console IN CNAME www;

# Nouveaux Alias
intranet IN CNAME www;
extranet IN CNAME www;
wiki IN CNAME www;
```



Nous redémarrons le service bind pour que la modification du fichier de configuration soit prise en compte :

```
systemctl restart bind9
```

Nous créons maintenant les dossiers qui contiendront les fichiers liés aux différents sites Internet : les Virtual Hosts.

```
mkdir -p /home/htdocs/m2l.org/www  
mkdir -p /home/htdocs/m2l.org/intranet  
mkdir -p /home/htdocs/m2l.org/extranet  
mkdir -p /home/htdocs/m2l.org/wiki
```

Nous allons maintenant créer les fichiers de configuration pour chaque futur site. Pour cela, regardons un fichier de configuration type afin d'en comprendre le contenu :

```
# Début de la configuration du Virtual Host. Nous pouvons spécifier l'adresse IP ou le nom sur  
lequel ce Virtual Host répondra, mais également quel en sera le port d'écoute  
<VirtualHost XX:XX>  
    # Nom de domaine principal associé au Virtual Host  
    ServerName XXX  
    # Alias permettant la gestion des noms spécifiés gérés par le Virtual Host  
    ServerAlias XXX  
    # Chemin principal du répertoire racine du site web associé à ce Virtual Host  
    DocumentRoot XXX  
    # Chemin du fichier de journal des erreurs  
    ErrorLog XXX  
    # Chemin du fichier de journal d'accès personnalisé pour ce Virtual Host  
    CustomLog XXX  
    # Début de la configuration des autorisations et des paramètres spécifiques au  
répertoire racine du site  
    <Directory XXX>  
        # Autorise l'accès à tous les utilisateurs à partir du répertoire spécifié  
        Require all granted  
    </Directory>  
</VirtualHost>
```



A partir de ce fichier d'exemple, nous allons créer nos fichiers de configuration pour nos quatre sites :

/etc/apache2/sites-available/www.m2l.org.conf :

```
<VirtualHost *:80>
    ServerName m2l.org
    ServerAlias www.m2l.org
    DocumentRoot /home/htdocs/m2l.org/www
    ErrorLog /var/log/apache2/www-error.log
    CustomLog /var/log/apache2/www-access.log combined
    <Directory /home/htdocs/m2l.org/www>
        Require all granted
    </Directory>
</VirtualHost>
```

/etc/apache2/sites-available/intranet.m2l.org.conf :

```
<VirtualHost *:80>
    ServerName intranet.m2l.org
    DocumentRoot /home/htdocs/m2l.org/intranet
    ErrorLog /var/log/apache2/intranet-error.log
    CustomLog /var/log/apache2/intranet-access.log combined
    <Directory /home/htdocs/m2l.org/intranet>
        Require all granted
    </Directory>
</VirtualHost>
```

/etc/apache2/sites-available/extranet.m2l.org.conf :

```
<VirtualHost *:80>
    ServerName extranet.m2l.org
    DocumentRoot /home/htdocs/m2l.org/extranet
    ErrorLog /var/log/apache2/extranet-error.log
    CustomLog /var/log/apache2/extranet-access.log combined
    <Directory /home/htdocs/m2l.org/extranet>
        Require all granted
    </Directory>
</VirtualHost>
```



/etc/apache2/sites-available/wiki.m2l.org.conf :

```
<VirtualHost *:80>
    ServerName wiki.m2l.org
    DocumentRoot /home/htdocs/m2l.org/wiki
    ErrorLog /var/log/apache2/wiki-error.log
    CustomLog /var/log/apache2/wiki-access.log combined
    <Directory /home/htdocs/m2l.org/wiki>
        Require all granted
    </Directory>
</VirtualHost>
```

Une fois que nos fichiers de configuration sont créés, nous devons activer ces nouveaux Virtual Hosts. Pour ce faire, nous créons un lien symbolique entre les fichiers de configuration du dossier /etc/apache2/sites-available/ (le dossier des sites disponibles) et le dossier /etc/apache2/sites-enabled/ (le dossier des sites activés) en entrant successivement les commandes suivantes :

```
a2ensite www.m2l.org
a2ensite intranet.m2l.org
a2ensite extranet.m2l.org
a2ensite wiki.m2l.org
```

Nous redémarrons le service apache2 pour que la création et la configuration des Virtual Hosts soient pris en compte :

```
systemctl restart apache2
```

Nous pouvons vérifier dans le dossier /var/log/apache2 que nous avons bien nos fichiers de logs de créés :

```
error.log.8.gz      intranet-error.log      www-error.log
error.log.9.gz      other_vhosts_access.log
extranet-access.log wiki-access.log
extranet-error.log  wiki-error.log
intranet-access.log www-access.log
```



Nous créons des pages HTML/CSS pour tester nos Virtual Hosts :

```
<!--index.html-->
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Site Web www.m2l.org</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
<div class="content"><br/><br/><br/>Bienvenue sur le site web <br/>www.m<em>2</em>l.org</div>
  </body>
</html>
```

```
/*style.css*/
body {background-color: #111;}
.content {
  width:100%;
  text-align: center;
  color: white;
  font-size: 25px;
}
em {
  font-weight: bold;
  font-style: normal;
  color: #44cc99;
  font-size: 25px;
}
```

Nous pouvons nous rendre sur le site internet :



DokuWiki

Qu'est-ce que DokuWiki ?

DokuWiki est un système de gestion de contenu (CMS) open-source utilisé pour créer et gérer des sites web, en particulier des wikis. Il a été conçu principalement pour la création de documentation collaborative, de bases de connaissances et de wikis d'entreprise, bien qu'il puisse également être utilisé pour d'autres types de sites web.

Installation de DokuWiki

Pour commencer l'installation de notre DokuWiki qui sera hébergé sur le site wiki.m2l.org, nous devons d'abord installer le paquet `wget` :

```
apt update && apt upgrade  
apt install wget
```

Nous pouvons maintenant télécharger l'archive du DokuWiki et la décompresser :

```
wget https://download.dokuwiki.org/src/dokuwiki/dokuwiki-stable.tgz  
tar xzvf dokuwiki-stable.tgz
```

Nous allons maintenant remplacer le Virtual Host de `wiki` par le dossier du DokuWiki :

```
mv dokuwiki-2022-07-31a ..  
cd ..  
rm -rf wiki  
mv dokuwiki-2022-07-31a/ wiki  
chown -R www-data wiki
```

Nous pouvons maintenant nous rendre sur l'adresse wiki.m2l.org/install.php dans notre navigateur pour finaliser la configuration du DokuWiki.

Nous créons un super utilisateur (`ldumas`) et on lui choisit un mot de passe. Nous appellerons notre DokuWiki documentation.





DokuWiki Installer

Wiki Name

☒ Enable ACL (recommended)

Superuser

Real name

E-Mail

Password


once again

Initial ACL policy

Open Wiki (read, write, upload for everyone) ▼

☐ Allow users to register themselves

Notre DokuWiki sera donc accessible depuis l'adresse wiki.m2l.org/documentation :

 documentation

Log In

Search

Recent Changes Media Manager Sitemap







Trace:

start

This topic does not exist yet

You've followed a link to a topic that doesn't exist yet. If permissions allow, you may create it by clicking on **Create this page**.

Except where otherwise noted, content on this wiki is licensed under the following license: CC Attribution-Share Alike 4.0 International

 BY-SA  DONATE  PHP POWERED  W3C HTML5  WP-CSS 



Les fichiers .htaccess

Un fichier .htaccess est un fichier de configuration utilisé principalement avec le serveur web Apache, bien que d'autres serveurs web puissent également le prendre en charge. Il permet de définir des règles et des directives spécifiques pour la configuration du serveur web au niveau du répertoire.

Création d'une page de connexion

Nous devons maintenant protéger l'intranet. Pour cela, nous modifions le fichier de configuration du site intranet (/etc/apache2/sites-available/intranet.m2l.org.conf) :

```
<VirtualHost *:80>
    ServerName intranet.m2l.org
    DocumentRoot /home/htdocs/m2l.org/intranet
    ErrorLog /var/log/apache2/intranet-error.log
    CustomLog /var/log/apache2/intranet-access.log combined
    <Directory /home/htdocs/m2l.org/intranet>
        Require all granted
        AllowOverride All
    </Directory>
</VirtualHost>
```

L'instruction AllowOverride All permet au répertoire de prendre en compte les fichiers .htaccess. Nous créons dès à présent le fichier .htaccess dans le dossier /home/htdocs/m2l.org/intranet. Ce fichier permet de verrouiller le dossier dans lequel il se trouve. Seule une personne dont le compte est inscrit dans le fichier .htpasswd pourra s'authentifier :

```
AuthType Basic
AuthUserFile /var/www/accounts/.htpasswd
AuthName "Reserved Access"
Require valid-user
```

Nous créons ensuite le dossier qui accueillera le fichier .htpasswd :

```
mkdir /var/www/accounts
```



Nous entrons ensuite la commande suivante, permettant de créer le fichier .htpasswd et d'y enregistrer un utilisateur :

```
htpasswd -c /var/www/accounts/.htpasswd sio
```

L'argument -c permet de créer le fichier htpasswd. Il n'est pas nécessaire de l'intégrer à la commande une fois que ce dernier est créé. Après avoir effectué ces changements, nous redémarrons le service Apache2.

```
systemctl restart apache2
```

Nous vérifions ensuite sur la page intranet.m2l.org si la page de connexion a été mise en place.

Personnalisation des pages d'erreur

Il est également possible grâce aux fichiers .htaccess de pouvoir personnaliser les pages d'erreur des différents sites. Pour ce faire, nous rajoutons dans le fichier .htaccess du dossier intranet les lignes suivantes, permettant de personnaliser les erreurs 401,403,404 et 500 :

```
ErrorDocument 401 /401.html  
ErrorDocument 403 /403.html  
ErrorDocument 404 /404.html  
ErrorDocument 500 /500.html
```

Nous créons ensuite les pages associées aux erreurs :

401.html :

```
<!doctype html>  
<html lang="fr">  
  <head>  
    <meta charset="utf-8">  
    <title>Site Web www.m2l.org</title>  
    <link rel="stylesheet" href="style.css">  
  </head>  
  <body>  
    <div class="content"><br /><br /><br />Les identifiants sont incorrects.</div>  
  </body>  
</html>
```



403.html :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Site Web www.m2l.org</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div class="content"><br /><br /><br />Accès restreint. </div>
  </body>
</html>
```

404.html :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Site Web www.m2l.org</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div class="content"><br /><br /><br />Page inexistante</div>
  </body>
</html>
```

500.html :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Site Web www.m2l.org</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div class="content"><br /><br /><br />Problème serveur</div>
  </body>
</html>
```



style.css :

```
body {  
  background-color: #111;  
}  
  
.content {  
  width:100%;  
  text-align: center;  
  color: white;  
  font-size: 25px;  
}
```



Les répertoires personnels

Nous devons maintenant mettre en place un répertoire personnel. Les répertoires personnels permettent de limiter l'accès à des parties du site en créant un espace dédié à chaque utilisateur. Dans le cas de DokuWiki, par exemple, cela permet de pouvoir avoir plusieurs utilisateurs alimentant chacun leur documentation sans qu'ils puissent accéder à la documentation des autres.

Pour créer un répertoire personnel, nous configurons Apache2 pour activer le module userdir :

```
a2enmod userdir
```

Nous créons le dossier /home/sio/public_html qui fera office de répertoire public :

```
mkdir /home/sio/public_html
```

Nous créons les pages index.html et index.php

index.html :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Site Web www.m2l.org</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div class="content"><br /><br /><br />Bienvenue sur le site
web<br/>www.m<em>2</em>l.org</div>
  </body>
</html>
```



index.php :

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

Nous modifions ensuite le fichier `/etc/apache2/mods-enabled/php7.4.conf` en changeant la ligne « `php_admin_flag engine Off` » en « `php_admin_flag engine On` » :

```
<IfModule mod_userdir.c>
  <Directory /home/*/public_html>
    php_admin_flag engine On #il faut passer cette valeur a on car elle est en off par défaut
  </Directory>
</IfModule>
```

Nous redémarrons le service Apache2 :

```
systemctl restart apache2
```

