

Workshop

Administration d'un Serveur Apache

Concepteur : A. DAOUDI

Durée totale : 3h30

Sommaire

- Introduction
- Objectifs Pédagogiques
- Plan du Workshop
- 1. Préparation de l'Environnement (15 minutes)
- 2. Configuration de Base d'Apache (45 minutes)
- 3. Mise en Place de Plusieurs Hôtes Virtuels (vhosts) (60 minutes)
- 4. Configuration de Redirections avec `.htaccess` (30 minutes)
- 5. Analyse et Interprétation des Logs d'Apache (30 minutes)
- 6. Introduction à HTTPS (30 minutes)
- 7. Synthèse et Conclusion (15 minutes)
- Ressources Supplémentaires
- Conclusion

Introduction

Bienvenue dans ce workshop qui vous guidera pas à pas dans la configuration d'un serveur Apache en local. Ce tutoriel est spécialement conçu pour vous aider à acquérir les compétences nécessaires pour le projet **Web4All** du CESI. À la fin de cette session, vous serez capable de :

- Configurer un serveur Apache avec plusieurs hôtes virtuels (vhosts), dont un dédié aux ressources statiques.
- Mettre en place des redirections avec le fichier **.htaccess**.
- Comprendre les bases de HTTPS et l'importance de la sécurité web.

Objectifs Pédagogiques

- **Comprendre** les principes de base de la configuration d'Apache.
- **Configurer** plusieurs hôtes virtuels (vhosts) sur un même serveur Apache.
- **Mettre en place** des redirections avec le fichier **.htaccess**.
- **Localiser et interpréter** les journaux (logs) d'Apache pour diagnostiquer des problèmes.
- **Connaître** les fondements de HTTPS et son importance pour la sécurité.

Plan du Workshop

1. Préparation de l'Environnement (15 minutes)
2. Configuration de Base d'Apache (45 minutes)
3. Mise en Place de Plusieurs Hôtes Virtuels (vhosts) (60 minutes)
4. Configuration de Redirections avec **.htaccess** (30 minutes)
5. Analyse et Interprétation des Logs d'Apache (30 minutes)
6. Introduction à HTTPS (30 minutes)
7. Synthèse et Conclusion (15 minutes)

1. Préparation de l'Environnement (15 minutes)

Objectif

S'assurer que vous avez Apache installé et fonctionnel sur votre machine Windows ou Linux.

Étapes à Suivre

1. Vérifier l'installation d'Apache :

- **Ouvrez votre navigateur** et accédez à **http://localhost**.
- Si vous voyez la page par défaut d'Apache (souvent une page avec "It works!"), vous pouvez passer à l'étape suivante.

- **Sinon**, vous devez installer Apache.

2. Installer Apache si nécessaire :

- **Windows :**
 - Téléchargez Apache depuis [Apache Lounge](#).
 - **Astuce :** Choisissez la version compatible avec votre système (32 ou 64 bits).
 - Suivez les instructions d'installation fournies.
 - **Important :** Installez-le dans un répertoire facile d'accès, par exemple [C:/Apache24/](#).
- **Linux (Ubuntu/Debian) :**
 - Ouvrez le Terminal.
 - Exécutez les commandes suivantes :

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

- **Astuce :** Le mot de passe administrateur peut être requis.

3. Vérifier à nouveau :

- Une fois l'installation terminée, retournez sur <http://localhost> pour confirmer qu'Apache fonctionne.

Félicitations ! Vous avez préparé votre environnement pour le workshop.

2. Configuration de Base d'Apache (45 minutes)

Objectif

Définir le répertoire racine de votre site web dans Apache et comprendre les implications en matière de sécurité.

Étapes à Suivre

1. Localiser le fichier de configuration principal d'Apache :

- **Windows :** [C:/Apache24/conf/httpd.conf](#)
- **Linux :** [/etc/apache2/apache2.conf](#)

Astuce : Vous aurez besoin de droits administrateur pour modifier ce fichier.

2. Modifier la directive **DocumentRoot** :

- **Recherchez** la ligne commençant par **DocumentRoot**.
- **Remplacez** le chemin actuel par celui de votre projet. Par exemple :

■ **Windows :**

```
DocumentRoot "C:/Apache24/htdocs/cesi-site"
```

■ **Linux :**

```
DocumentRoot /var/www/html/cesi-site
```

3. **Mettre à jour la directive `<Directory>` correspondante :**

- **Recherchez** le bloc `<Directory>` correspondant à votre `DocumentRoot`.
- **Modifiez-le** pour qu'il corresponde au nouveau chemin, et assurez-vous que les options sont correctement définies :

```
<Directory "chemin/vers/cesi-site">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

Explication :

- `AllowOverride All` permet l'utilisation de fichiers `.htaccess`.
- `Require all granted` autorise l'accès à tous les clients.

4. **Créer le répertoire du projet :**

- **Windows :**
 - Allez dans `C:/Apache24/htdocs/` et créez un nouveau dossier nommé `cesi-site`.
- **Linux :**

```
sudo mkdir /var/www/html/cesi-site
```

Astuce : Vérifiez que vous avez les permissions nécessaires.

5. **Créer un fichier `index.html` :**

- **Dans le répertoire `cesi-site`**, créez un fichier nommé `index.html`.
- **Contenu du fichier :**

```
<!DOCTYPE html>
<html>
<head>
  <title>Site CESI</title>
</head>
<body>
  <h1>Bienvenue sur le site CESI</h1>
</body>
</html>
```

Astuce : Vous pouvez utiliser n'importe quel éditeur de texte (Notepad++, Sublime Text, etc.).

6. Redémarrer Apache :

◦ Windows :

- Ouvrez l'**Invite de commandes** en tant qu'administrateur.
- Exécutez :

```
httpd -k restart
```

- **Ou** redémarrez le service Apache via le Gestionnaire des services.

◦ Linux :

```
sudo systemctl restart apache2
```

7. Tester dans le navigateur :

- Accédez à <http://localhost>.
- **Vous devriez voir** votre page personnalisée avec le message "Bienvenue sur le site CESI".

Super ! Vous avez configuré le répertoire racine de votre site.

3. Mise en Place de Plusieurs Hôtes Virtuels (vhosts) (60 minutes)

Objectif

Configurer plusieurs vhosts pour séparer le site principal et les ressources statiques.

Étapes à Suivre

1. Modifier le fichier **hosts** pour ajouter des noms de domaine locaux :

- **Windows :**
 - Ouvrez `C:\Windows\System32\drivers\etc\hosts` avec un éditeur de texte en mode administrateur.
- **Linux :**

```
sudo nano /etc/hosts
```

- **Ajoutez** les lignes suivantes à la fin du fichier :

```
127.0.0.1 cesi-site.local
127.0.0.1 cesi-static.local
```

Explication : Cela permet à votre ordinateur de reconnaître `cesi-site.local` et `cesi-static.local` comme pointant vers votre machine locale.

2. Créer les répertoires pour les sites :

- **Windows :**
 - Dans `C:/Apache24/htdocs/`, créez deux dossiers :
 - `cesi-site`
 - `cesi-static`
- **Linux :**

```
sudo mkdir /var/www/html/cesi-site
sudo mkdir /var/www/html/cesi-static
```

Astuce : Organisez vos fichiers pour ne pas les confondre.

3. Créer les fichiers de configuration pour les vhosts :

- **Windows :**
 - Ouvrez `C:/Apache24/conf/extra/httpd-vhosts.conf`.
- **Linux :**
 - Créez deux fichiers :
 - `/etc/apache2/sites-available/cesi-site.conf`
 - `/etc/apache2/sites-available/cesi-static.conf`
- **Configuration pour `cesi-site` :**

```
<VirtualHost *:80>
    ServerName cesi-site.local
    DocumentRoot "chemin/vers/cesi-site"
    <Directory "chemin/vers/cesi-site">
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

◦ **Configuration pour `cesi-static` :**

```
<VirtualHost *:80>
    ServerName cesi-static.local
    DocumentRoot "chemin/vers/cesi-static"
    <Directory "chemin/vers/cesi-static">
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>
</VirtualHost>
```

Explication :

- `ServerName` définit le nom de domaine du vhost.
- `DocumentRoot` spécifie le répertoire racine du vhost.
- `AllowOverride None` pour `cesi-static` signifie que les fichiers `.htaccess` ne seront pas pris en compte, renforçant la sécurité.

4. Activer les vhosts :

◦ **Windows :**

- Assurez-vous que la ligne suivante est décommentée dans `httpd.conf` :

```
Include conf/extra/httpd-vhosts.conf
```

◦ **Linux :**

```
sudo a2ensite cesi-site.conf
sudo a2ensite cesi-static.conf
```

- **Redémarrez Apache :**

```
sudo systemctl restart apache2
```

5. Créer des fichiers `index.html` pour tester les vhosts :

- Dans `cesi-site` :

- Créez un `index.html` avec le contenu :

```
<h1>Bienvenue sur CESI Site</h1>
```

- Dans `cesi-static` :

- Créez un `index.html` avec le contenu :

```
<h1>Bienvenue sur CESI Static</h1>
```

6. Tester les vhosts dans le navigateur :

- Accédez à `http://cesi-site.local` et `http://cesi-static.local`.
- **Vous devriez voir** les pages respectives.

Bravo ! Vous avez mis en place plusieurs hôtes virtuels.

4. Configuration de Redirections avec `.htaccess` (30 minutes)

Objectif

Apprendre à utiliser le fichier `.htaccess` pour configurer des redirections.

Étapes à Suivre

1. Activer la prise en charge des fichiers `.htaccess` :

- Dans la configuration du vhost `cesi-site`, assurez-vous que `AllowOverride All` est bien défini.

2. Créer un fichier `.htaccess` dans `cesi-site` :

- **Windows :**

- Dans `C:/Apache24/htdocs/cesi-site/`, créez un fichier nommé `.htaccess`.
- **Astuce :** Si Windows n'autorise pas la création de fichiers commençant par un point, créez-le via l'invite de commandes.

- **Linux :**


```
sudo touch /var/www/html/cesi-site/.htaccess
```

3. Ajouter une redirection simple :

- **Ouvrez** le fichier `.htaccess` avec un éditeur de texte.
- **Ajoutez** la ligne suivante :

```
Redirect /ancienne-page.html http://cesi-site.local/nouvelle-page.html
```

Explication : Toute requête vers `/ancienne-page.html` sera redirigée vers `/nouvelle-page.html`.

4. Créer le fichier `nouvelle-page.html` :

- Dans `cesi-site`, créez `nouvelle-page.html` avec un contenu simple :

```
<h1>Ceci est la nouvelle page</h1>
```

5. Tester la redirection :

- Accédez à `http://cesi-site.local/ancienne-page.html`.
- **Vous devriez être redirigé** vers `http://cesi-site.local/nouvelle-page.html`.

6. Configurer une redirection avec `mod_rewrite` :

- **Activer le module `mod_rewrite` :**

▪ Linux :

```
sudo a2enmod rewrite
sudo systemctl restart apache2
```

▪ Windows :

- Vérifiez que la ligne `LoadModule rewrite_module modules/mod_rewrite.so` est décommentée dans `httpd.conf`.

- **Ajouter une règle de réécriture dans `.htaccess` :**

```
RewriteEngine On
RewriteRule ^ancien-repertoire/(.*)$ nouveau-repertoire/$1 [R=301,L]
```

Explication :

- Cette règle redirige toutes les requêtes de **ancien-repertoire** vers **nouveau-repertoire**, en conservant le reste du chemin.

7. Tester la redirection avancée :

- Créez des dossiers **ancien-repertoire** et **nouveau-repertoire** avec des fichiers pour tester.
- Accédez à `http://cesi-site.local/ancien-repertoire/fichier.html`.
- **Vous devriez être redirigé** vers `http://cesi-site.local/nouveau-repertoire/fichier.html`.

Excellent travail ! Vous maîtrisez désormais les redirections avec **.htaccess**.

5. Analyse et Interprétation des Logs d'Apache (30 minutes)

Objectif

Apprendre à localiser et interpréter les logs d'Apache pour diagnostiquer des problèmes.

Étapes à Suivre

1. Localiser les fichiers de logs :

- **Windows :** `C:/Apache24/logs/`
 - `access.log` et `error.log`
- **Linux :** `/var/log/apache2/`
 - `access.log` et `error.log`

2. Générer des erreurs pour alimenter les logs :

- Accédez à une page inexistante, par exemple :

```
http://cesi-site.local/page-inexistante.html
```

3. Analyser `access.log` :

- **Ouvrez** le fichier `access.log`.
- **Recherchez** les entrées correspondantes.
- **Exemple d'entrée :**

```
127.0.0.1 - - [Date] "GET /page-inexistante.html HTTP/1.1" 404 209
```

4. Analyser `error.log` :

- **Ouvrez** le fichier `error.log`.
- **Recherchez** les messages d'erreur.
- **Exemple d'entrée :**

```
[Date] [error] [client 127.0.0.1] File does not exist:
/chemin/vers/cesi-site/page-inexistante.html
```

5. Interpréter les logs :

- **Comprendre les codes de statut HTTP :**
 - **200 OK** : Requête réussie.
 - **404 Not Found** : Ressource non trouvée.
 - **500 Internal Server Error** : Erreur interne du serveur.
- **Utiliser les logs pour diagnostiquer :**
 - Identifiez les pages manquantes ou les erreurs de configuration.

Vous savez maintenant comment utiliser les logs pour résoudre les problèmes !

6. Introduction à HTTPS (30 minutes)

Objectif

Comprendre les fondements de HTTPS et son importance pour la sécurité web.

Étapes à Suivre

1. Comprendre le rôle de HTTPS :

- **HTTPS** signifie **HyperText Transfer Protocol Secure**.
- Il s'agit de la version sécurisée de HTTP, utilisant SSL/TLS pour chiffrer les communications entre le client et le serveur.

2. Pourquoi HTTPS est important :

- **Sécurité des données** : Protège les informations sensibles (mots de passe, données personnelles).
- **Confiance des utilisateurs** : Les navigateurs indiquent que le site est sécurisé.
- **SEO** : Les moteurs de recherche favorisent les sites en HTTPS.

3. Mise en place de HTTPS en local (optionnel) :

- **Générer un certificat auto-signé :**
 - **Linux :**

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
cesi-site.local.key -out cesi-site.local.crt
```

Astuce : Lorsqu'on vous demande le **Common Name**, entrez `cesi-site.local`.

- **Windows :**
 - Installez OpenSSL pour Windows.
 - Exécutez une commande similaire dans l'invite de commandes.
- **Configurer Apache pour utiliser SSL :**
 - **Créez un vhost pour le port 443** (HTTPS).
 - **Ajoutez** les directives suivantes :

```
SSLEngine on  
SSLCertificateFile "chemin/vers/cesi-site.local.crt"  
SSLCertificateKeyFile "chemin/vers/cesi-site.local.key"
```

- **Activer le module SSL :**

- **Linux :**

```
sudo a2enmod ssl  
sudo systemctl restart apache2
```

- **Windows :**
 - Décommentez la ligne `LoadModule ssl_module modules/mod_ssl.so` dans `httpd.conf`.

- **Redémarrez Apache** et testez en accédant à `https://cesi-site.local`.

- **Note :** Vous recevrez un avertissement de sécurité car le certificat n'est pas signé par une autorité de confiance.

Important : En environnement de production, utilisez des certificats émis par une autorité reconnue comme Let's Encrypt.

7. Synthèse et Conclusion (15 minutes)

Ce que vous avez accompli

- **Configuré Apache** pour héberger votre site web en local.
- **Créé plusieurs vhosts**, séparant le site principal des ressources statiques.
- **Utilisé .htaccess** pour mettre en place des redirections.
- **Appris à analyser les logs** pour diagnostiquer des problèmes.
- **Compris l'importance de HTTPS** et comment le mettre en place en local.

Discussion

- **Questions à réfléchir :**
 - Quelles sont les bonnes pratiques en matière de sécurité web ?
 - Comment ces compétences peuvent-elles être appliquées au projet Web4All ?
- **Prochaines étapes :**
 - Intégrer ces configurations dans le développement du projet Web4All.
 - Explorer des outils complémentaires pour la gestion et la sécurité des serveurs web.

Félicitations ! Vous êtes désormais prêt à appliquer ces compétences à votre projet.

Ressources Supplémentaires

- **Documentation Apache :**
 - [Apache HTTP Server Documentation](#)
 - [Guide sur les modules Apache](#)
- **Tutoriels en ligne :**
 - [Configurer SSL avec Apache](#)
 - [Comprendre les certificats SSL](#)
- **Outils d'analyse de logs :**
 - [GoAccess](#) : outil en temps réel pour analyser les logs d'Apache.
 - [AWStats](#) : générateur de rapports graphiques sur l'utilisation du serveur.

Conclusion

En suivant ce workshop, vous avez acquis des compétences essentielles pour la configuration d'un serveur Apache en local, compétences qui seront précieuses pour le projet **Web4All** du CESI. N'hésitez pas à revisiter ce guide si vous avez besoin d'un rappel, et à explorer les ressources supplémentaires pour approfondir vos connaissances.