

PROSIT 2 : Mission Apache

[]

	Role
Animateur	rayan
Gestionnaire	yanis
Scribe	sabrina
Secrétaire	

1. Mots clés :

http/https

Mots à définir :

- **Serveur Apache local** : Installation du serveur web Apache sur une machine locale pour tester et développer des sites web sans avoir besoin d'un serveur distant.
- • **Hôtes virtuels (Virtual Hosts)** : Fonctionnalité d'Apache permettant d'héberger plusieurs sites web sur un même serveur en configurant différentes adresses ou noms de domaine.
- • **Sys/DevOps** : Combinaison des rôles de l'administration système (SysAdmin) et des pratiques DevOps, visant à automatiser le déploiement, la gestion et la scalabilité des applications.
- • **Répertoire racine (Root Directory)** : Dossier principal d'un système de fichiers où sont stockées les données essentielles, souvent utilisé pour désigner le dossier où un site web est hébergé sur un serveur.
- • **Code d'état HTTP (HTTP Status Code)** : Réponse envoyée par un serveur web indiquant le résultat d'une requête HTTP (ex. : 200 OK pour une requête réussie, 404 Not Found pour une page inexistante).
- • **Log Apache** : Fichiers journaux générés par le serveur Apache, contenant des informations sur les requêtes HTTP, les erreurs et l'activité du serveur.
- • **Document Root** : Répertoire spécifique d'un serveur web où sont stockés les fichiers accessibles publiquement via HTTP (ex. : /var/www/html pour Apache).
- • **Méthodes HTTP GET et POST** :
 - GET : Méthode HTTP utilisée pour récupérer des ressources sans modifier l'état du serveur.
 - POST : Méthode utilisée pour envoyer des données au serveur, souvent pour soumettre des formulaires ou interagir avec une API.
- **.htaccess** : Fichier de configuration utilisé par Apache pour définir des règles spécifiques pour un répertoire, comme la redirection d'URL, la protection par mot de passe ou la gestion des erreurs.
-

2. Analyse du contexte :

Une entreprise a besoin de configurer un serveur local Apache afin d'améliorer les performances de leur site web en répartissant les ressources de celui-ci.

3. Définition de la problématique :

Comment configurer efficacement cet environnement pour garantir sécurité, performances et un bon suivi des erreurs et des requêtes?

4. Plan d'actions :

1. faire des recherches sur les serveurs locaux apache :
2. *

Apache est un serveur web open-source largement utilisé pour héberger des sites en local ou en production. Il permet de :

- Servir des fichiers HTML, CSS, JS et PHP.
- Gérer plusieurs sites avec les hôtes virtuels.
- Appliquer des règles de sécurité et de redirection avec `.htaccess`.
- Surveiller les requêtes grâce aux logs.

Installation sur différentes plateformes :

- **Windows** : XAMPP (Apache + MySQL + PHP + Perl)
- **Linux** : `sudo apt install apache2` (Debian/Ubuntu)
- **Mac** : `brew install httpd`

Vérifier si Apache fonctionne :

```
sh
```

```
apachectl -v # Version d'Apache
```

```
sudo systemctl status apache2 # Vérifier l'état du service
```

3. configurer des hôtes virtuels (répertoire root +ressources statiques):

L'objectif est d'avoir **deux hôtes virtuels** :

- `cesi-site.local` pour le site principal.
- `cesi-static.local` pour les fichiers statiques (JS, CSS, images...).

Étape 1 : Modifier la configuration d'Apache

- Ouvre le fichier de configuration des hôtes virtuels :

makefile

C:\xampp\apache\conf\extra\httpd-vhosts.conf

- Ajoute ces lignes à la fin :

apache

```
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/cesi-site"
    ServerName cesi-site.local
    <Directory "C:/xampp/htdocs/cesi-site">
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/cesi-static"
    ServerName cesi-static.local
    <Directory "C:/xampp/htdocs/cesi-static">
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Étape 2 : Modifier le fichier `hosts` pour lier les noms de domaine locaux

- Ouvre Bloc-notes en mode administrateur, puis édite le fichier :

makefile

C:\Windows\System32\drivers\etc\hosts

- Ajoute ces lignes :

lua

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

Étape 3 : Redémarrer Apache

-

Ouvre **XAMPP Control Panel**, puis clique sur **Stop** et **Start** pour **Apache**.

-
-

Teste dans ton navigateur :

- - `http://cesi-site.local` → Doit afficher ton site principal.
 - `http://cesi-static.local` → Doit afficher tes fichiers statiques.

4. analyser et surveiller les logs

4. Analyser et surveiller les logs

Apache génère des **logs** qui permettent d'analyser les erreurs et suivre les requêtes.

Emplacement des logs dans XAMPP

-

Log des erreurs :

- **Log des erreurs :**
 - lua
 - C:\xampp\apache\logs\error.log
 - Contient les erreurs (ex : fichiers manquants, erreurs PHP...).
- **Log des accès :**
 - pgsql
 - C:\xampp\apache\logs\access.log
 - Contient toutes les requêtes HTTP (ex : méthode GET, POST...).
-

5. gérer les requêtes http

Les requêtes HTTP permettent d'échanger des données entre le client et le serveur.

Méthodes HTTP principales

- **GET** → Lire des données (ex : afficher une page web).
- **POST** → Envoyer des données (ex : soumettre un formulaire).
- **PUT** → Mettre à jour une ressource.
- **DELETE** → Supprimer une ressource.

Outils pour analyser les requêtes

- **Chrome DevTools (F12 > Network)** → Permet de voir les requêtes en temps réel.
- **Postman** → Tester les API en envoyant des requêtes GET, POST...

Exemple d'un .htaccess pour gérer les requêtes

Dans `C:/xampp/htdocs/cesi-site/.htaccess` :

```
apache
```

```
# Redirige tout vers index.php sauf les fichiers existants
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php?route=$1 [QSA,L]
```

• Cela permet d'avoir des URLs propres (`/page/contact` au lieu de `page.php?name=contact`).

6. implémenter https et configurer le .HT Access

Activer HTTPS dans XAMPP

XAMPP inclut **OpenSSL**, qui permet d'activer HTTPS.

1.

Générer un certificat SSL auto-signé

2.

```
cd C:\xampp\apache  
makecert.bat
```

3. Cela va générer un certificat.
- 3.

Activer le certificat dans Apache

```
makefile  
  
C:\xampp\apache\conf\extra\httpd-ssl.conf
```

CopierModifier

```
Listen 443  
<VirtualHost _default_:443>  
    DocumentRoot "C:/xampp/htdocs/cesi-site"  
    ServerName cesi-site.local  
    SSLEngine on  
    SSLCertificateFile "conf/ssl.crt/server.crt"  
    SSLCertificateKeyFile "conf/ssl.key/server.key"  
</VirtualHost>
```

hypothèse :

**une serveur locale apache n'augmente pas beaucoup les performance
d'un site simple
virtuel hosts servent à l'hébergement du site**

