

Cours Complet

Installation et Configuration des Serveurs Web

IIS et Apache

Avec Travaux Dirigés Pratiques

Table des matières

PARTIE 1 : COURS IIS	3
Introduction	3
Activation d'IIS sur Windows	3
Configuration de base d'IIS	5
Configuration avancée	7
Maintenance et dépannage	8
PARTIE 2 : COURS APACHE	10
Introduction	10
Installation sur Windows	10
Installation sur Linux	12
Installation sur macOS	14
Configuration de base	16
Configuration HTTPS/SSL	19
Fichiers .htaccess	20
Maintenance et dépannage	21
Optimisation et sécurité	23
PARTIE 3 : TRAVAUX DIRIGÉS	25
TD : Déploiement d'un projet web	25
Exercice 1 : Déploiement sur IIS	25
Exercice 2 : Déploiement sur Apache	27

PARTIE 1

Installation et Configuration d'IIS

Introduction

Internet Information Services (IIS) est le serveur web de Microsoft intégré à Windows. Il permet d'héberger des sites web, des applications web et des services web sur des systèmes Windows.

Partie 1 : Activation d'IIS sur Windows

Prérequis

- Système d'exploitation : Windows 10/11 (version Professionnel ou Entreprise) ou Windows Server
- Droits administrateur sur la machine

Procédure d'activation

Étape 1 : Accéder aux fonctionnalités Windows

Sur Windows 10/11, ouvrez le Panneau de configuration, puis sélectionnez « Programmes et fonctionnalités ». Cliquez sur « Activer ou désactiver des fonctionnalités Windows » dans le menu de gauche.

Étape 2 : Sélectionner IIS

Dans la fenêtre qui s'ouvre, recherchez « Services Internet (IIS) » ou « Internet Information Services ». Cochez la case principale pour activer IIS. Pour une installation complète, développez l'arborescence et sélectionnez les composants souhaités, notamment Services World Wide Web, qui contient les fonctionnalités essentielles comme le serveur HTTP, les fonctionnalités de sécurité, et les outils de gestion.

Étape 3 : Composants recommandés

Pour un serveur web fonctionnel, assurez-vous de cocher au minimum les éléments suivants :

- Fonctionnalités HTTP communes (document par défaut, exploration de répertoires, erreurs HTTP)
- Développement d'applications (ASP.NET si nécessaire, ISAPI Extensions et Filters)
- Intégrité et diagnostics (journalisation HTTP, observateur de requêtes)
- Outils de gestion (Console de gestion IIS)

Étape 4 : Installation

Cliquez sur « OK » pour lancer l'installation. Windows télécharge et installe les composants nécessaires, ce qui peut prendre plusieurs minutes. Un redémarrage n'est généralement pas nécessaire, mais peut être demandé dans certains cas.

Étape 5 : Vérification

Pour vérifier que IIS fonctionne correctement, ouvrez un navigateur web et tapez « `http://localhost` » dans la barre d'adresse. Si l'installation a réussi, vous devriez voir la page d'accueil par défaut d'IIS.

Partie 2 : Configuration de base d'IIS

Accéder au gestionnaire IIS

Recherchez « Gestionnaire des services Internet (IIS) » dans le menu Démarrer et lancez l'application. L'interface du gestionnaire IIS s'affiche avec une arborescence sur la gauche montrant votre serveur et ses sites.

Structure d'IIS

L'arborescence contient plusieurs niveaux : le serveur (nœud racine), les pools d'applications (groupes isolés pour l'exécution d'applications), et les sites web (sites hébergés sur le serveur).

Configuration d'un site web

Créer un nouveau site

Dans le gestionnaire IIS, faites un clic droit sur « Sites » dans l'arborescence de gauche, puis sélectionnez « Ajouter un site web ». Une fenêtre de configuration s'ouvre.

Paramètres du site

Renseignez le nom du site (par exemple « MonSiteWeb »), choisissez ou créez un pool d'applications dédié, indiquez le chemin physique vers le répertoire contenant vos fichiers web (créez ce dossier au préalable si nécessaire, par exemple C:\inetpub\monsite).

Configuration de la liaison (Binding)

Définissez le type de protocole (HTTP ou HTTPS), l'adresse IP (généralement « Toutes non attribuées »), le port (80 pour HTTP, 443 pour HTTPS), et éventuellement un nom d'hôte si vous hébergez plusieurs sites.

Gestion des pools d'applications

Les pools d'applications isolent les applications web les unes des autres pour améliorer la sécurité et la stabilité. Pour configurer un pool, sélectionnez « Pools d'applications » dans l'arborescence, puis cliquez sur le pool concerné. Vous pouvez modifier la version du .NET Framework, le mode pipeline (Intégré ou Classique), et l'identité sous laquelle le pool s'exécute.

Configuration des permissions

Les fichiers de votre site web doivent avoir les permissions appropriées. Le compte utilisateur du pool d'applications (généralement IIS_IUSRS ou ApplicationPoolIdentity) doit avoir accès en lecture aux fichiers du site. Pour les modifier, faites un clic droit sur le dossier du site dans l'Explorateur Windows, sélectionnez « Propriétés », puis l'onglet « Sécurité ».

Document par défaut

IIS cherche automatiquement certains fichiers lorsqu'on accède à un répertoire sans préciser de fichier. Pour configurer cela, sélectionnez votre site dans le gestionnaire IIS, puis double-cliquez sur « Document par défaut ». Vous pouvez ajouter, supprimer ou réorganiser les documents par défaut comme index.html,

default.aspx, etc.

Partie 3 : Configuration avancée

Configuration HTTPS/SSL

Pour sécuriser votre site avec HTTPS, vous devez d'abord obtenir un certificat SSL. Dans le gestionnaire IIS, sélectionnez votre serveur, puis double-cliquez sur « Certificats de serveur ». Vous pouvez créer un certificat auto-signé pour les tests ou importer un certificat d'une autorité de certification. Ensuite, retournez à votre site, faites un clic droit et sélectionnez « Modifier les liaisons », ajoutez une nouvelle liaison de type HTTPS sur le port 443, et sélectionnez votre certificat SSL.

Gestion des logs

IIS enregistre automatiquement les requêtes dans des fichiers journaux. Pour configurer la journalisation, sélectionnez votre site, double-cliquez sur « Journalisation », et choisissez le format, la fréquence de rotation et l'emplacement des fichiers de logs.

Restrictions IP et domaines

Pour contrôler qui peut accéder à votre site, double-cliquez sur « Restrictions IP et domaines » au niveau du site. Vous pouvez autoriser ou refuser l'accès à des adresses IP ou plages d'adresses spécifiques.

Configuration de l'authentification

IIS supporte plusieurs modes d'authentification. Dans les fonctionnalités du site, vous trouverez « Authentification » qui permet d'activer l'authentification anonyme, l'authentification Windows, l'authentification de base, etc.

Partie 4 : Maintenance et dépannage

Démarrer/Arrêter un site

Dans le gestionnaire IIS, cliquez sur un site et utilisez les actions « Démarrer », « Arrêter » ou « Redémarrer » dans le panneau de droite.

Recycler un pool d'applications

Si une application ne répond plus, vous pouvez recycler son pool d'applications. Sélectionnez le pool dans « Pools d'applications » et cliquez sur « Recycler ».

Vérifier les logs d'erreurs

Les logs IIS se trouvent généralement dans C:\inetpub\logs\LogFiles. Consultez également l'Observateur d'événements Windows (Journaux Windows > Application et Système) pour les erreurs liées à IIS.

Conclusion

IIS est un serveur web puissant et flexible pour l'hébergement d'applications Windows. Cette base vous permet de démarrer avec IIS, mais le serveur offre de nombreuses fonctionnalités avancées comme la réécriture d'URL, la compression, le caching, et bien d'autres que vous pourrez explorer selon vos besoins.

besoins.

PARTIE 2

Installation et Configuration d'Apache

Introduction

Apache HTTP Server (souvent appelé simplement Apache) est un serveur web open source très populaire, multi-plateforme et hautement configurable. Il est utilisé pour héberger des sites web statiques et dynamiques (PHP, Python, etc.).

Partie 1 : Installation d'Apache sur Windows

Prérequis

- Système d'exploitation : Windows 10/11 ou Windows Server
- Droits administrateur
- Visual C++ Redistributable (généralement inclus)

Méthode 1 : Installation via binaires pré-compilés

Étape 1 : Téléchargement

Rendez-vous sur le site Apache Lounge (<https://www.apachelounge.com/download/>), une source fiable pour les binaires Windows d'Apache. Téléchargez la version appropriée (Win64 ou Win32) au format ZIP.

Étape 2 : Extraction

Extrayez le contenu de l'archive ZIP. Vous obtiendrez un dossier « Apache24 ». Déplacez ce dossier à la racine de votre disque C:\ pour obtenir C:\Apache24. Ce n'est pas obligatoire mais c'est la convention recommandée.

Étape 3 : Configuration initiale

Ouvrez le fichier de configuration C:\Apache24\conf\httpd.conf avec un éditeur de texte (Notepad++ recommandé). Vérifiez la directive ServerRoot qui doit pointer vers votre installation, par exemple ServerRoot "c:/Apache24". Notez l'utilisation de slashes normaux même sous Windows.

Étape 4 : Installation en tant que service

Ouvrez une invite de commandes en tant qu'administrateur. Naviguez vers le dossier bin d'Apache avec la commande cd C:\Apache24\bin, puis installez Apache comme service Windows avec la commande httpd.exe -k install. Pour démarrer le service, utilisez httpd.exe -k start.

Méthode 2 : Installation via XAMPP

XAMPP est un package qui inclut Apache, MySQL, PHP et Perl. Téléchargez XAMPP depuis <https://www.apachefriends.org>, lancez l'installateur et suivez les instructions. Lancez ensuite le « XAMPP Control Panel » et cliquez sur « Start » à côté d'Apache.

Partie 2 : Installation d'Apache sur Linux

Ubuntu/Debian

Étape 1 : Mise à jour du système

Ouvrez un terminal et mettez à jour la liste des paquets avec la commande :

```
sudo apt update
```

Étape 2 : Installation

Installez Apache avec la commande :

```
sudo apt install apache2
```

Étape 3 : Gestion du service

Pour démarrer Apache :

```
sudo systemctl start apache2
```

Pour l'arrêter :

```
sudo systemctl stop apache2
```

Pour le redémarrer :

```
sudo systemctl restart apache2
```

Pour activer le démarrage automatique au boot :

```
sudo systemctl enable apache2
```

Étape 4 : Configuration du pare-feu

Si vous utilisez UFW, autorisez le trafic HTTP et HTTPS :

```
sudo ufw allow 'Apache'
```

CentOS/RHEL/Fedora

Sur ces distributions, Apache s'appelle httpd. Installez-le avec :

```
sudo dnf install httpd
```

Démarrez le service :

```
sudo systemctl start httpd
```

Autorisez HTTP et HTTPS dans le pare-feu :

```
sudo firewall-cmd --permanent --add-service=http
```

```
sudo firewall-cmd --reload
```

Structure des dossiers sous Linux

Sur Ubuntu/Debian :

- Fichiers de configuration : /etc/apache2/

- Fichier principal : /etc/apache2/apache2.conf
- Sites disponibles : /etc/apache2/sites-available/
- Sites actifs : /etc/apache2/sites-enabled/
- Répertoire web par défaut : /var/www/html/
- Logs : /var/log/apache2/

Partie 3 : Installation d'Apache sur macOS

Méthode 1 : Apache préinstallé

Les anciennes versions de macOS incluaient Apache. Vous pouvez vérifier sa présence avec httpd -v dans le terminal. Pour démarrer Apache, utilisez sudo apachectl start.

Méthode 2 : Installation via Homebrew (recommandé)

Étape 1 : Installation de Homebrew

Si Homebrew n'est pas installé, installez-le depuis <https://brew.sh> en exécutant la commande fournie sur le site.

Étape 2 : Installation d'Apache

Dans le terminal, exécutez :

```
brew install httpd
```

Étape 3 : Démarrage

Pour démarrer Apache :

```
brew services start httpd
```

Étape 4 : Configuration

Le fichier de configuration se trouve dans /opt/homebrew/etc/httpd/httpd.conf (Apple Silicon) ou /usr/local/etc/httpd/httpd.conf (Intel). Le répertoire web par défaut est /opt/homebrew/var/www.

Méthode 3 : Installation via MAMP

MAMP (Mac, Apache, MySQL, PHP) est similaire à XAMPP. Téléchargez-le depuis <https://www.mamp.info>, installez l'application, lancez MAMP, et cliquez sur « Start Servers ». Accédez à <http://localhost:8888> par défaut.

Partie 4 : Configuration de base d'Apache

Fichier de configuration principal

Le fichier httpd.conf (ou apache2.conf sur Ubuntu) contient la configuration principale d'Apache. Avant toute modification, créez une sauvegarde avec sudo cp httpd.conf httpd.conf.backup.

Directives importantes

Listen

Définit le port d'écoute, par défaut 80 pour HTTP. Exemple : Listen 80 ou Listen 8080 pour un port alternatif.

ServerName

Définit le nom du serveur. Exemple : ServerName localhost:80 ou ServerName www.example.com:80.

DocumentRoot

Indique le répertoire contenant vos fichiers web. Exemple : DocumentRoot "/var/www/html" sur Linux.

DirectoryIndex

Spécifie les fichiers à servir par défaut. Exemple : DirectoryIndex index.html index.php.

Configuration des répertoires

Pour chaque répertoire servi, vous devez définir les permissions avec la directive Directory. Exemple :

```
<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

Création d'un Virtual Host

Les Virtual Hosts permettent d'héberger plusieurs sites sur un même serveur Apache.

Sur Linux (Ubuntu/Debian)

Créez un fichier de configuration dans /etc/apache2/sites-available/, par exemple monsite.conf :

```
<VirtualHost *:80>
    ServerName monsite.local
    DocumentRoot /var/www/monsite
    ErrorLog ${APACHE_LOG_DIR}/monsite-error.log
    <Directory /var/www/monsite>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Créez le répertoire avec :

```
sudo mkdir /var/www/monsite
```

Activez le site avec :

```
sudo a2ensite monsite.conf
```

Rechargez Apache avec :

```
sudo systemctl reload apache2
```

Modules Apache

Apache utilise des modules pour étendre ses fonctionnalités.

Activer un module sur Linux

Sur Ubuntu/Debian, utilisez :

```
sudo a2enmod nom_du_module
```

Par exemple, pour activer le module de réécriture :

```
sudo a2enmod rewrite
```

Modules courants

- mod_rewrite : permet la réécriture d'URL
- mod_ssl : active le support HTTPS/SSL
- mod_php : intègre PHP dans Apache
- mod_headers : permet la manipulation des en-têtes HTTP
- mod_deflate : active la compression gzip

Partie 5 : Configuration HTTPS/SSL

Génération d'un certificat auto-signé (pour tests)

Sur Linux

Activez le module SSL :

```
sudo a2enmod ssl
```

Créez un certificat auto-signé avec :

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
-keyout /etc/ssl/private/apache-selfsigned.key \
-out /etc/ssl/certs/apache-selfsigned.crt
```

Créez un Virtual Host SSL dans /etc/apache2/sites-available/monsite-ssl.conf :

```
<VirtualHost *:443>
    ServerName monsite.local
    DocumentRoot /var/www/monsite
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>
```

Activez le site SSL :

```
sudo a2ensite monsite-ssl.conf
sudo systemctl reload apache2
```

Partie 6 : Fichiers .htaccess

Les fichiers .htaccess permettent de configurer Apache au niveau des répertoires sans modifier la configuration principale.

Activation

Dans httpd.conf, assurez-vous que AllowOverride est défini sur All pour le répertoire concerné.

Exemples d'utilisation

Réécriture d'URL

```
RewriteEngine On
RewriteRule ^ancien-page$ nouvelle-page.html [R=301,L]
```

Protection par mot de passe

```
AuthType Basic
AuthName "Zone protégée"
AuthUserFile /chemin/vers/.htpasswd
Require valid-user
```

Redirection HTTP vers HTTPS

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://{$HTTP_HOST}{REQUEST_URI} [R=301,L]
```

Partie 7 : Maintenance et dépannage

Tester la configuration

Avant de redémarrer Apache après une modification, testez la configuration.

Sur Linux Ubuntu/Debian :

```
sudo apache2ctl configtest
```

Sur CentOS/RHEL :

```
sudo httpd -t
```

Si le test retourne « Syntax OK », la configuration est valide.

Consultation des logs

Les logs sont essentiels pour le dépannage.

Sur Linux Ubuntu, consultez :

```
sudo tail -f /var/log/apache2/error.log
```

Sur CentOS :

```
sudo tail -f /var/log/httpd/error_log
```

Problèmes courants

Apache ne démarre pas

- Vérifiez que le port 80 n'est pas déjà utilisé avec netstat -tuln | grep :80
- Vérifiez les logs d'erreur
- Testez la configuration avec configtest

Erreur 403 Forbidden

- Vérifiez que le répertoire a les permissions appropriées dans la configuration
- Vérifiez que Require all granted est présent
- Vérifiez les permissions du système de fichiers

Changement de permissions (Linux)

Pour donner les bonnes permissions à votre répertoire web :

```
sudo chown -R www-data:www-data /var/www/monsite  
sudo chmod -R 755 /var/www/monsite
```

Partie 8 : Optimisation et sécurité

Sécurité de base

Masquer la version d'Apache

Dans httpd.conf, ajoutez :

```
ServerTokens Prod  
ServerSignature Off
```

Désactiver la liste des répertoires

Dans la directive Directory, retirez Indexes des Options :

```
Options -Indexes FollowSymLinks
```

Performance

Activer la compression

Activez mod_deflate et ajoutez :

```
<IfModule mod_deflate.c>  
    AddOutputFilterByType DEFLATE text/html text/plain text/xml  
</IfModule>
```

Cache du navigateur

Activez mod_expires et ajoutez :

```
<IfModule mod_expires.c>  
    ExpiresActive On  
    ExpiresByType image/jpg "access plus 1 year"  
</IfModule>
```

Conclusion

Apache est un serveur web robuste et flexible, disponible sur toutes les plateformes principales. Sa configuration modulaire permet de l'adapter à pratiquement tous les besoins d'hébergement web.

PARTIE 3

Travaux Dirigés Pratiques

Déploiement d'un projet web HTML/CSS/JavaScript

Objectifs du TD

Ce TD vous permettra de mettre en pratique les connaissances acquises dans les cours précédents en déployant un même projet web sur deux serveurs différents : IIS (Windows) et Apache (multi-plateforme). Vous apprendrez à configurer chaque serveur pour héberger correctement votre site web.

Prérequis

- Avoir suivi les parties 1 et 2 du cours (IIS et Apache)
- Avoir IIS installé et configuré (sur Windows)
- Avoir Apache installé et configuré (sur Windows, Linux ou Mac)
- Connaissances de base en HTML, CSS et JavaScript

Étape préliminaire : Création du projet web

Avant de déployer sur les serveurs, créons un projet web simple mais complet qui inclut HTML, CSS et JavaScript.

Structure du projet

Créez une structure de dossiers comme suit :

```
mon-site-web/
  index.html
  css/
    style.css
  js/
    script.js
  images/
    logo.png (optionnel)
```

Fichier index.html

Créez le fichier index.html avec le contenu suivant :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mon Site Web - TD Serveurs</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <header>
```

```
<h1>Bienvenue sur mon site web</h1>
<nav>
  <ul>
    <li><a href="#accueil">Accueil</a></li>
    <li><a href="#apropos">À propos</a></li>
    <li><a href="#contact">Contact</a></li>
  </ul>
</nav>
</header>
<main>
  <section id="accueil">
    <h2>Accueil</h2>
    <p>Site déployé sur <span id="serveur-type"></span></p>
    <button onclick="afficherHeure()">Afficher l'heure</button>
    <p id="heure"></p>
  </section>
</main>
<footer>
  <p>© 2026 - TD Serveurs Web</p>
</footer>
<script src="js/script.js"></script>
</body>
</html>
```

Fichier css/style.css

Créez le fichier CSS avec le contenu suivant :

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
    background-color: #f4f4f4;
}

header {
    background: #35424a;
    color: white;
    padding: 20px;
    text-align: center;
}

nav ul {
    list-style: none;
    display: flex;
    justify-content: center;
    gap: 20px;
    margin-top: 10px;
}

nav a {
    color: white;
    text-decoration: none;
}

main {
    max-width: 800px;
    margin: 20px auto;
    padding: 20px;
    background: white;
    border-radius: 5px;
}

button {
    background: #35424a;
    color: white;
    padding: 10px 20px;
    border: none;
    cursor: pointer;
}

footer {
    text-align: center;
    padding: 20px;
    background: #35424a;
    color: white;
}
```

Fichier js/script.js

Créez le fichier JavaScript avec le contenu suivant :

```
// Fonction pour afficher l'heure actuelle
function afficherHeure() {
    const maintenant = new Date();
    const heureElement = document.getElementById('heure');
    heureElement.textContent = 'Il est ' + maintenant.toLocaleTimeString();
}

// Déterminer le serveur (approximatif)
window.addEventListener('DOMContentLoaded', function() {
    const serveurElement = document.getElementById('serveur-type');
    // Cette détection est basique et à titre pédagogique
    serveurElement.textContent = 'Serveur Web';
});
```

Exercice 1 : Déploiement sur IIS (Windows)

Objectif

Déployer le projet web sur un serveur IIS et le rendre accessible localement via `http://monsite.local`

Étape 1 : Préparation du répertoire

1. Créez un dossier pour votre site :

```
C:\inetpub\mon-site-web\
```

2. Copiez tous les fichiers de votre projet (`index.html`, dossiers `css/`, `js/`, `images/`) dans ce répertoire.
3. Vérifiez que la structure est correcte et que tous les fichiers sont présents.

Étape 2 : Configuration des permissions

1. Faites un clic droit sur le dossier `C:\inetpub\mon-site-web\`
2. Sélectionnez « Propriétés » > Onglet « Sécurité »
3. Cliquez sur « Modifier » puis « Ajouter »
4. Ajoutez l'utilisateur « IIS_IUSRS » avec les permissions de lecture
5. Cliquez sur « OK » pour valider

Étape 3 : Création du site dans IIS

1. Ouvrez le « Gestionnaire des services Internet (IIS) »
2. Dans l'arborescence de gauche, faites un clic droit sur « Sites »
3. Sélectionnez « Ajouter un site web »
4. Remplissez les informations suivantes :
 - Nom du site : MonSiteWeb
 - Pool d'applications : DefaultAppPool (ou créez-en un nouveau)
 - Chemin physique : `C:\inetpub\mon-site-web`
 - Type : http
 - Adresse IP : Toutes non attribuées
 - Port : 80
 - Nom d'hôte : `monsite.local`
5. Cliquez sur « OK »

Étape 4 : Configuration du fichier hosts

Pour accéder au site via `http://monsite.local`, vous devez modifier le fichier hosts de Windows.

1. Ouvrez Notepad en tant qu'administrateur
2. Ouvrez le fichier : `C:\Windows\System32\drivers\etc\hosts`

3. Ajoutez la ligne suivante à la fin du fichier :

```
127.0.0.1 monsite.local
```

4. Enregistrez et fermez le fichier

Étape 5 : Test du déploiement

1. Dans le Gestionnaire IIS, sélectionnez votre site « MonSiteWeb »
2. Dans le panneau de droite, cliquez sur « Démarrer » si le site n'est pas déjà démarré
3. Ouvrez un navigateur web (Chrome, Firefox, Edge...)
4. Accédez à <http://monsite.local>
5. Vérifiez que votre site s'affiche correctement
6. Testez le bouton « Afficher l'heure » pour vérifier que JavaScript fonctionne

Étape 6 : Vérification des logs

1. Dans le Gestionnaire IIS, sélectionnez votre site
2. Double-cliquez sur « Journalisation »
3. Notez l'emplacement des fichiers journaux (généralement C:\inetpub\logs\LogFiles)
4. Accédez à ce dossier et ouvrez le fichier log le plus récent
5. Vérifiez que vos requêtes HTTP apparaissent dans les logs

Questions de réflexion

1. Que se passe-t-il si vous changez le port de 80 à 8080 ? Comment accédez-vous au site ?
2. Comment pourriez-vous héberger plusieurs sites sur le même serveur IIS ?
3. Quelles sont les différences entre un pool d'applications et un site web dans IIS ?

Exercice 2 : Déploiement sur Apache

Objectif

Déployer le même projet web sur un serveur Apache et le rendre accessible localement via `http://monsite-apache.local`

A. Déploiement sur Apache (Linux - Ubuntu/Debian)

Étape 1 : Préparation du répertoire

1. Créez un dossier pour votre site :

```
sudo mkdir -p /var/www/mon-site-apache
```

2. Copiez vos fichiers dans ce répertoire :

```
sudo cp -r mon-site-web/* /var/www/mon-site-apache/
```

3. Définissez les permissions appropriées :

```
sudo chown -R www-data:www-data /var/www/mon-site-apache
```

```
sudo chmod -R 755 /var/www/mon-site-apache
```

Étape 2 : Création du Virtual Host

1. Créez un fichier de configuration pour votre site :

```
sudo nano /etc/apache2/sites-available/mon-site-apache.conf
```

2. Ajoutez la configuration suivante :

```
<VirtualHost *:80>
    ServerName monsite-apache.local
    ServerAdmin admin@monsite-apache.local
    DocumentRoot /var/www/mon-site-apache

    ErrorLog ${APACHE_LOG_DIR}/mon-site-apache-error.log
    CustomLog ${APACHE_LOG_DIR}/mon-site-apache-access.log combined

    <Directory /var/www/mon-site-apache>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

3. Enregistrez et fermez (Ctrl+X, puis Y, puis Entrée)

Étape 3 : Activation du site

1. Activez le nouveau site :

```
sudo a2ensite mon-site-apache.conf
```

2. Testez la configuration Apache :

```
sudo apache2ctl configtest
```

3. Si le test affiche « Syntax OK », rechargez Apache :

```
sudo systemctl reload apache2
```

Étape 4 : Configuration du fichier hosts

1. Modifiez le fichier hosts :

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

2. Ajoutez la ligne suivante :

```
127.0.0.1 monsite-apache.local
```

3. Enregistrez et fermez

Étape 5 : Test du déploiement

1. Ouvrez un navigateur web
2. Accédez à http://monsite-apache.local
3. Vérifiez que votre site s'affiche correctement
4. Testez le bouton JavaScript

Étape 6 : Consultation des logs

1. Consultez les logs d'accès :

```
sudo tail -f /var/log/apache2/mon-site-apache-access.log
```

2. Consultez les logs d'erreur :

```
sudo tail -f /var/log/apache2/mon-site-apache-error.log
```

B. Déploiement sur Apache (Windows avec XAMPP)

Étape 1 : Préparation du répertoire

1. Copiez votre dossier mon-site-web dans :

```
C:\xampp\htdocs\mon-site-apache\
```

Étape 2 : Configuration du Virtual Host

1. Ouvrez le fichier : C:\xampp\apache\conf\extra\httpd-vhosts.conf
2. Ajoutez à la fin du fichier :

```
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/mon-site-apache"
    ServerName monsite-apache.local
    <Directory "C:/xampp/htdocs/mon-site-apache">
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

3. Enregistrez le fichier

Étape 3 : Configuration du fichier hosts

1. Ouvrez C:\Windows\System32\drivers\etc\hosts en administrateur
2. Ajoutez la ligne :

```
127.0.0.1 monsite-apache.local
```

Étape 4 : Redémarrage d'Apache

1. Ouvrez le XAMPP Control Panel
2. Arrêtez Apache (bouton Stop)
3. Démarrez Apache (bouton Start)

Étape 5 : Test

1. Accédez à http://monsite-apache.local dans votre navigateur
2. Vérifiez le bon fonctionnement du site

C. Déploiement sur Apache (macOS avec Homebrew)

Étape 1 : Préparation

1. Créez le dossier :

```
mkdir -p /opt/homebrew/var/www/mon-site-apache
```

2. Copiez vos fichiers :

```
cp -r mon-site-web/* /opt/homebrew/var/www/mon-site-apache/
```

Étape 2 : Configuration Virtual Host

1. Éditez le fichier de configuration :

```
nano /opt/homebrew/etc/httpd/httpd.conf
```

2. Recherchez et décommentez la ligne :

```
Include /opt/homebrew/etc/httpd/extra/httpd-vhosts.conf
```

3. Éditez le fichier vhosts :

```
nano /opt/homebrew/etc/httpd/extra/httpd-vhosts.conf
```

4. Ajoutez votre Virtual Host (similaire à la version Linux)

Étape 3 : Fichier hosts et test

1. Éditez /etc/hosts :

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

2. Ajoutez : 127.0.0.1 monsite-apache.local

3. Redémarrez Apache :

```
brew services restart httpd
```

4. Testez <http://monsite-apache.local:8080> (ou port 80 si configuré)

Exercices complémentaires

Exercice 3 : Comparaison des performances

1. Utilisez les outils de développement de votre navigateur (F12)
2. Comparez les temps de chargement entre IIS et Apache
3. Analysez les en-têtes HTTP retournés par chaque serveur
4. Notez les différences observées

Exercice 4 : Ajout de fonctionnalités

1. Ajoutez un formulaire de contact dans votre site
2. Créez un fichier .htaccess pour Apache avec des redirections

3. Configurez des pages d'erreur personnalisées (404, 500)
4. Testez ces fonctionnalités sur les deux serveurs

Exercice 5 : Sécurisation

1. Générez un certificat SSL auto-signé
2. Configurez HTTPS sur IIS
3. Configurez HTTPS sur Apache
4. Testez l'accès sécurisé (<https://>)
5. Configurez une redirection automatique HTTP vers HTTPS

Questions de synthèse

1. Quelles sont les principales différences d'installation entre IIS et Apache ?
2. Comparez les fichiers de configuration : httpd.conf (Apache) vs Gestionnaire IIS
3. Quels sont les avantages et inconvénients de chaque serveur ?
4. Comment gérez-vous les logs sur chaque serveur ?
5. Quel serveur préféreriez-vous utiliser pour un projet professionnel et pourquoi ?
6. Comment hébergeeriez-vous plusieurs sites sur un même serveur avec IIS et Apache ?
7. Quelles sont les bonnes pratiques de sécurité communes aux deux serveurs ?

Conclusion du TD

Ce travail dirigé vous a permis de mettre en pratique l'installation et la configuration de deux serveurs web majeurs : IIS et Apache. Vous avez appris à déployer un même projet sur différentes plateformes, à configurer des Virtual Hosts, à gérer les permissions, et à diagnostiquer les problèmes courants. Ces compétences sont essentielles pour tout administrateur système ou développeur web.

N'hésitez pas à explorer davantage les fonctionnalités avancées de chaque serveur, comme la mise en cache, la compression, le load balancing, et les modules de sécurité. La maîtrise de ces outils vous permettra de déployer et maintenir des applications web robustes et performantes.