

---

# Serveur Web, WordPress et hébergement - Configuration et mise en ligne

Cours sur la configuration et la sécurisation d'un serveur Web, la création de sites avec WordPress, la publication chez un hébergeur, le changement d'hébergeur et la sauvegarde des données (Apache/IIS, hébergement, FTP, HTTPS, authentification).

**Marketing et Business** **35 min de lecture** **Niveau Intermédiaire**

---

Document généré le 11/05/2026 à 12h48 · [nouv.fr/wiki/serveur-web-wordpress-hebergement](https://nouv.fr/wiki/serveur-web-wordpress-hebergement)

# Sommaire

19 section(s) · 35 min de lecture

↳ Objectifs opérationnels

↳ Objectifs d'apprentissage

## 1. Généralités Internet

## 2. Présentation de HTML, CSS et PHP

## 3. WordPress comme CMS

## 4. Mise en place de WordPress

## 5. Gestion des pages et contenus avec WordPress

## 6. Mise en ligne et hébergement

↳ 6.1 Choisir un hébergement

↳ 6.2 Export et import de la base de données

↳ 6.3 Copie du site via FTP

## 7. Sauvegarde des données d'un site

## 8. Serveurs Web - Présentation et mise en œuvre

↳ 8.1 Principales solutions du marché

↳ 8.2 Installation et configuration

↳ 8.3 Hôtes virtuels (multi-sites)

↳ 8.4 Mise en place d'HTTPS

↳ 8.5 Authentification

## Synthèse

Cours sur la configuration, l'optimisation et la sécurisation d'un serveur Web, la création de sites à l'aide de **WordPress**, la publication chez un hébergeur et la planification des sauvegardes.

---

## Objectifs opérationnels

- Configurer, optimiser et sécuriser un **serveur Web**.
- Créer un **site web** à l'aide de **WordPress**.
- **Publier** un site chez un hébergeur.
- **Changer d'hébergeur** (migration).
- **Planifier et mettre en œuvre la sauvegarde** des données d'un site.

## Objectifs d'apprentissage

- Comprendre les protocoles et standards du Web (HTTP, HTTPS, HTML, CSS, PHP).
- Installer et configurer un serveur Web (Apache ou IIS).
- Mettre en place **WordPress** (installation, base de données, thèmes, widgets).
- Gérer la mise en ligne (hébergement, FTP, export/import de base de données).
- Sécuriser un site (HTTPS, authentification).
- Assurer la continuité (sauvegardes, changement d'hébergeur).

## 1. Généralités Internet

---

- **Protocoles** : TCP/IP, HTTP/HTTPS, DNS, FTP, SSH.
- **Standards** : normes W3C, évolution du Web (historique, statique vs dynamique).
- **Rôle du serveur Web** : délivrer des pages et des ressources (HTML, CSS, JS, images) en réponse aux requêtes des clients (navigateurs).

Ce socle permet de comprendre le fonctionnement d'un site web et les échanges client-serveur.

---

## 2. Présentation de HTML, CSS et PHP

---

- **HTML** : structure des pages (balises, sémantique, formulaires).
- **CSS** : mise en forme et mise en page (sélecteurs, propriétés, responsive).
- **PHP** : langage côté serveur (variables, boucles, conditions, inclusion de fichiers, interaction avec une base de données).

Une maîtrise minimale de ces technologies est utile pour personnaliser WordPress (thèmes, templates) et comprendre le fonctionnement des sites dynamiques.

---

### 3. WordPress comme CMS

---

- **WordPress** : système de gestion de contenu (CMS) open source, basé sur PHP et MySQL/MariaDB, permettant de créer et gérer un site web sans tout coder à la main (contenus, menus, médias).
  - **Intérêt** : rapidité de mise en place, administration via l'interface d'administration (/wp-admin), thèmes et extensions (plugins), très répandu et bien documenté.
  - **Usages** : blog, site vitrine, portfolio, e-commerce (avec WooCommerce), multi-sites (WordPress Multisite).
- 

### 4. Mise en place de WordPress

---

- **Prérequis serveur** : serveur Web (Apache ou IIS), PHP (version minimale indiquée sur wordpress.org), base de données MySQL ou MariaDB.
- **Installation** : téléchargement sur [wordpress.org](https://wordpress.org), déploiement des fichiers dans la racine du site (ou sous-dossier), création de la base de données et d'un utilisateur MySQL. Lancer l'assistant en ouvrant l'URL du site : titre du site, identifiant et mot de passe administrateur, préfixe des tables (par défaut wp\_).
- **Fichier wp-config.php** : généré lors de l'installation ; contient les paramètres de connexion à la base (nom de la BDD, utilisateur, mot de passe, préfixe, clés de sécurité). À ne pas exposer ni modifier sans précaution.
- **Base de données** : WordPress stocke contenus (articles, pages), utilisateurs, réglages, thèmes et plugins dans la BDD. Sauvegarde et restauration via export/import SQL (phpMyAdmin, mysqldump).
- **PHP** : extensions recommandées pour WordPress (mysqli, GD, curl, mbstring, etc.) et limites (mémoire, temps d'exécution) selon la doc officielle.

Une installation propre et documentée facilite les mises à jour (noyau, thèmes, plugins) et les sauvegardes.

---

### 5. Gestion des pages et contenus avec WordPress

---

- **Articles et pages** : dans **Tableau de bord** → **Articles** et **Pages** : création, édition, publication, brouillons, catégories et étiquettes pour les articles.
- **Médias** : **Médias** → **Bibliothèque** : upload de fichiers (images, PDF, etc.), organisation en dossiers (ou par date), insertion dans les contenus (galeries, pièces jointes). Les fichiers sont stockés dans wp-content/uploads/.
- **Menus** : **Apparence** → **Menus** : création de menus (en-tête, pied de page, etc.), structure de navigation, liens internes (pages, articles) et externes.
- **Thèmes** : **Apparence** → **Thèmes** : choix et activation d'un thème, installation de thèmes depuis le dépôt WordPress ou upload. Personnalisation via **Apparence** → **Personnaliser** (couleurs, en-tête, logo, pied de page).
- **Widgets** : **Apparence** → **Widgets** (ou **Personnaliser** → **Widgets**) : zones de widgets (barre latérale, footer, etc.) et blocs réutilisables (texte, liens, recherche, etc.).

## 6. Mise en ligne et hébergement

---

### 6.1 Choisir un hébergement

- **Critères** : espace disque, trafic, base de données, PHP (version compatible WordPress), support (WordPress préinstallé ou installation manuelle), prix, contrat.
- **Types** : hébergement mutualisé, VPS, dédié, selon la taille et la criticité du site.

### 6.2 Export et import de la base de données

- **Export** : dump SQL (phpMyAdmin, ligne de commande `mysqldump`) pour sauvegarder la structure et les données.
- **Import** : restauration du dump sur le serveur de l'hébergeur (après création de la base et de l'utilisateur).
- **Adaptations** : après migration WordPress, il est souvent nécessaire d'adapter les URLs dans la base (options `siteurl` et `home` dans la table `wp_options`, et URLs en dur dans le contenu). Outils : recherche/remplacement SQL (avec précaution) ou plugins de migration (Better Search Replace, Duplicator).

### 6.3 Copie du site via FTP

- **FTP/SFTP** : transfert des fichiers du site WordPress (dossier racine : `wp-admin`, `wp-content`, `wp-includes`, `wp-config.php`, etc.) depuis la machine locale ou l'ancien hébergeur vers le nouveau serveur. Le dossier `wp-content` contient thèmes, plugins et uploads.
- **Bonnes pratiques** : utiliser SFTP plutôt que FTP (chiffrement), vérifier les droits des fichiers et dossiers après copie (souvent 755 pour les dossiers, 644 pour les fichiers), ne pas exposer de fichiers sensibles (`wp-config.php` doit rester protégé, pas de sauvegardes en ligne dans un répertoire public).

La combinaison **export BDD + copie fichiers + adaptation des URLs** constitue la base d'une migration réussie.

---

## 7. Sauvegarde des données d'un site

---

- **Objectif** : pouvoir restaurer le site en cas de panne, erreur ou piratage.
- **Données à sauvegarder** : base de données WordPress (dump SQL régulier) et fichiers (dossier WordPress : `wp-content` au minimum — thèmes, plugins, uploads — et `wp-config.php` ; éventuellement tout le site pour restauration complète).
- **Planification** : fréquence (quotidienne, hebdomadaire), rétention (combien de sauvegardes garder), stockage hors serveur (NAS, cloud, autre hébergeur).
- **Mise en œuvre** : scripts cron (dump BDD + archive des fichiers), outils fournis par l'hébergeur, ou **plugins WordPress** dédiés (UpdraftPlus, BackWPup, Duplicator, etc.) pour sauvegardes et restauration automatisées.

- **Test de restauration** : vérifier régulièrement qu'une sauvegarde peut être restaurée (environnement de test).

Sans sauvegarde planifiée et testée, un incident peut entraîner la perte définitive des données.

---

## 8. Serveurs Web - Présentation et mise en œuvre

---

### 8.1 Principales solutions du marché

Solution	Contexte	Points forts
<b>Apache</b>	Linux, multi-plateforme	Modules nombreux, .htaccess, très répandu
<b>IIS</b> (Internet Information Services)	Windows Server	Intégration Microsoft, support ASP.NET
<b>Nginx</b>	Linux, haute charge	Performances, reverse proxy

Le cours se concentre sur **Apache** ou **IIS** selon l'environnement (souvent Apache en contexte Linux).

### 8.2 Installation et configuration

- **Installation** : paquets (ex. `apache2` sous Debian/Ubuntu), rôle IIS sous Windows Server.
- **Configuration** : fichier de configuration principal, fichiers de site (virtual hosts), redémarrage du service.
- **Publier des sites** : pointer la racine documentaire (DocumentRoot) vers le répertoire d'installation de WordPress, droits des fichiers/dossiers (utilisateur du serveur Web, ex. `www-data` sous Apache).

### 8.3 Hôtes virtuels (multi-sites)

- **Notion** : un même serveur peut héberger plusieurs sites (noms différents ou domaines) : chaque site a sa racine documentaire et éventuellement son nom de domaine.
- **Mise en place** : un bloc (VirtualHost sous Apache, site sous IIS) par site, avec le chemin du site et le nom de domaine (ou alias).
- **Intérêt** : isolation des sites, maintenance et sauvegardes par site.

### 8.4 Mise en place d'HTTPS

- **Objectif** : chiffrer les échanges et authentifier le serveur (confiance, SEO).
- **Certificat** : Let's Encrypt (gratuit, renouvellement automatique), certificat fourni par l'hébergeur, ou certificat commercial.
- **Configuration** : module SSL/TLS (Apache : `mod_ssl` ; IIS : liaison HTTPS), redirection HTTP → HTTPS, HSTS si pertinent.

## 8.5 Authentification

- **Authentification HTTP** (accès restreint à un répertoire ou à un site) : fichiers `.htpasswd` (Apache) ou équivalent IIS, pour protéger une zone d'admin ou un site en construction.
  - **Authentification applicative** : gérée par WordPress (comptes utilisateurs, rôles : administrateur, éditeur, auteur, etc., connexion au back-office `/wp-admin`). Distincte de l'auth HTTP mais complémentaire (ex. double protection sur `/wp-admin` avec `.htpasswd` pour limiter les attaques par force brute).
- 

## Synthèse

---

- **Serveur Web** : installer, configurer et sécuriser (Apache ou IIS) pour publier des sites.
- **WordPress** : installer et configurer WordPress (base de données, PHP, `wp-config.php`), gérer contenus (articles, pages), médias, menus, thèmes et widgets.
- **Hébergement** : choisir un hébergeur, publier un site (FTP, import BDD), migrer (export BDD, copie fichiers, adaptation URLs).
- **Sauvegarde** : planifier et mettre en œuvre des sauvegardes (BDD + fichiers), tester la restauration.
- **Sécurisation** : HTTPS, authentification (HTTP et/ou applicative), bonnes pratiques (mises à jour, droits, pas de fichiers sensibles en ligne).