

# AIPI - EXIGENCES TECHNIQUES D'HÉBERGEMENT

Documentation Technique Complète pour Intégration

## AVERTISSEMENT CRITIQUE DE COMPATIBILITÉ

Les widgets IA AIPI nécessitent des ressources d'hébergement spécifiques pour fonctionner correctement sur TOUS types d'applications web.

**IMPORTANT:** Cette documentation couvre WordPress, applications Node.js et autres frameworks.

85% des problèmes de performance sont dus à un hébergement inadéquat, indépendamment de la technologie utilisée.

## CAS D'INCOMPATIBILITÉ DOCUMENTÉS MULTI-PLATEFORMES

### HostGator - Plans Partagés (INCOMPATIBLES TOUTES TECHNOLOGIES):

- Plan Personal: 25% CPU max, timeout 30s  
RÉSULTAT: Gel confirmé sur WordPress, Laravel, Django, Express.js
- Plan Business: MÊMES limitations techniques  
RÉSULTAT: Échecs identiques sur React, Vue.js, Angular SSR
- Plan Baby: Restrictions CPU/RAM identiques  
RÉSULTAT: Timeouts sur Symfony, Rails, Flask, Drupal

### Autres Hébergeurs Problématiques Identifiés:

- GoDaddy Economy: Mêmes limitations 25% CPU
- Bluehost Basic: Timeouts fréquents applications Node.js
- Hostinger Single: Échecs sur applications Python/Django

### Symptômes par Type d'Application:

#### WordPress/CMS:

- Gel admin dashboard lors activation widget
- Timeouts requêtes AJAX > 30s
- Erreurs 504 Gateway Timeout sur pages avec widget

#### Applications Node.js/Express:

- Process terminé par CPU limit
- Connexions WebSocket fermées prématurément
- Memory leaks non gérés par restrictions RAM

#### Applications Python/Django/Flask:

- Gunicorn workers killed par OOM
- Timeouts sur requêtes ML/IA longues
- Celery tasks échouent par manque ressources

#### Applications PHP Custom/Laravel:

- Fatal errors memory exhausted
- Queue jobs timeout avant completion
- Sessions perdues lors pic charge

# EXIGENCES TECHNIQUES DÉTAILLÉES PAR PLATEFORME

## RESSOURCES SERVEUR MINIMALES:

CPU: Accès équivalent 1 core (pas limite 25%)

- WordPress: 0.5 core minimum pour widgets interactifs
- Node.js: 1 core complet pour event loop optimal
- Python: 0.8 core pour ML inference temps réel
- Applications SPA: 0.6 core pour SSR + widgets

RAM: Allocation minimum selon stack technologique

- WordPress: 512MB base + 256MB par widget actif
- Node.js: 1GB base + 512MB pour cache IA
- Python/Django: 1GB base + 1GB pour ML models
- PHP Laravel: 768MB base + 256MB pour queues

## CONFIGURATION SERVEUR WEB:

Architecture Recommandée par Type:

- NGINX + PHP-FPM: Optimal WordPress, Laravel, Symfony
- NGINX + Reverse Proxy: Requis Node.js, Python WSGI
- Apache + mod\_rewrite: Compatible mais performances moindres
- LiteSpeed: Excellent toutes plateformes (cache intégré)

Timeouts Serveur par Technologie:

- PHP max\_execution\_time: 300s minimum (widgets IA)
- NGINX proxy\_read\_timeout: 300s pour Node.js/Python
- Apache TimeOut: 300s pour requêtes longues ML
- FastCGI timeout: 300s pour PHP-FPM workers

## EXIGENCES BASE DE DONNÉES:

MySQL/MariaDB Configuration:

- Version: MySQL 5.7+ ou MariaDB 10.3+
- max\_connections: 200+ pour applications concurrent
- innodb\_buffer\_pool\_size: 25% RAM serveur minimum
- query\_cache\_size: 64MB pour requêtes répétitives

PostgreSQL Configuration:

- Version: PostgreSQL 12+ recommandé
- max\_connections: 200+ avec connection pooling
- shared\_buffers: 25% RAM pour performance optimale
- Extensions requises: uuid-oss, pgcrypto

## SÉCURITÉ ET CERTIFICATS:

SSL/TLS Configuration:

- Certificat valide requis (Let's Encrypt acceptable)
- TLS 1.2+ minimum pour sécurité widgets

- HTTPS Redirect obligatoire sur production
- HSTS headers recommandés (max-age=31536000)

#### Firewall et Restrictions:

- Ports ouverts: 80, 443 (HTTP/HTTPS)
- Port 22 (SSH) pour déploiements automatiques
- Whitelist IPs API pour callbacks webhooks
- Rate limiting: 1000 req/min par IP minimum

## FOURNISSEURS HÉBERGEMENT RECOMMANDÉS DÉTAILLÉS

### SOLUTIONS ÉCONOMIQUES (3-8€/mois):

#### ChemiCloud Starter - Multi-plateforme:

- Trafic: ~25,000 visites/mois
- CPU: Pas limite 25%, burst jusqu'à 2 cores
- Technologies: WordPress, PHP, Python, Node.js
- Base données: MySQL illimitées, PostgreSQL sur demande
- Support: Chat 24/7, migration gratuite

#### SiteGround StartUp - Optimisé Performance:

- Trafic: ~10,000 visites/mois
- Technologie: SG-Optimizer intégré, PHP 8.1+
- Cache: Redis, Memcached disponibles
- CDN: Cloudflare gratuit intégré
- Staging: Environnements test inclus

#### A2 Hosting Lite - Performance Turbo:

- CPU: AMD EPYC, pas limitations throttling
- SSD: NVMe ultra-rapides
- Technologies: Toutes (Python, Ruby, Node.js)
- Optimisations: HTTP/2, Brotli compression

### SOLUTIONS PROFESSIONNELLES (15-50€/mois):

#### Cloudways (Vultr/DigitalOcean) - DevOps Managé:

- Ressources: 1-4 cores dédiés, 1-8GB RAM
- Stack: NGINX, Redis, Elasticsearch disponible
- Déploiement: Git, staging automatique
- Monitoring: New Relic, server monitoring inclus
- Scaling: Vertical/horizontal automatique

#### Kinsta - WordPress Premium + Applications:

- Infrastructure: Google Cloud Premium Tier
- Performance: PHP 8.1, HTTP/3, cache edge

- DevTools: Staging, APM, logs temps réel
- Applications: Node.js, Python, static sites

#### **WP Engine - WordPress + Headless:**

- Spécialisation: WordPress + JAMstack
- Atlas: Headless WordPress + Faust.js
- Performance: Genesis Framework, cache avancé
- Sécurité: WAF, malware scanning, backups

## **CHECKLIST VÉRIFICATION COMPATIBILITÉ ÉTENDUE**

### **VÉRIFICATIONS PRÉ-IMPLÉMENTATION OBLIGATOIRES:**

#### **Tests Performance Serveur:**

1. TESTER: Temps réponse serveur < 200ms (GTMetrix)
2. VÉRIFIER: CPU usage baseline < 30% charge normale
3. MESURER: RAM disponible > 60% après stack loaded
4. CONFIRMER: Bande passante > 100Mbps (speedtest)

#### **Tests Configuration Technique:**

5. VALIDER: PHP version 8.0+ (php -v)
6. TESTER: Timeout PHP > 120s (phpinfo)
7. VÉRIFIER: Extensions PHP: curl, json, mbstring, openssl
8. CONFIRMER: Base données connexions > 50 simultanées

#### **Tests Sécurité et Réseau:**

9. VALIDER: SSL/TLS actif (ssllabs.com test A+)
10. TESTER: Ports 80/443 accessibles publiquement
11. VÉRIFIER: Firewall autorise trafic API endpoints
12. CONFIRMER: DNS résolution < 50ms globally

#### **Tests Compatibilité Widgets API:**

13. INSTALLER: Widget test sur page/environnement staging
14. TESTER: Chargement initial < 3s mobile/desktop
15. VÉRIFIER: Interactions temps réel fonctionnelles
16. MESURER: Pas timeouts sur requêtes IA complexes

## **CONFIGURATION AVANCÉE PRODUCTION**

### **Optimisations Performances par Stack:**

#### **WordPress + API:**

- Cache: W3 Total Cache + Redis object cache
- CDN: CloudFlare Pro avec Page Rules spécifiques
- Database: Query cache + persistent connections

- PHP: OPcache enabled, memory\_limit 512MB+

#### **Node.js + Express + AIPI:**

- Process: PM2 cluster mode, auto-restart
- Cache: Redis session store + response cache
- Proxy: NGINX reverse proxy, gzip/brotli
- Monitoring: Winston logs + APM integration

#### **Python Django/Flask + AIPI:**

- WSGI: Gunicorn + gevent workers optimal
- Cache: Memcached + Django cache framework
- Static: WhiteNoise + CDN pour assets
- Queue: Celery + Redis pour tâches async

## **SUPPORT TECHNIQUE ET MAINTENANCE**

#### **Monitoring Recommandé:**

- Uptime: Pingdom, UptimeRobot (checks 1min)
- Performance: New Relic, DataDog APM integration
- Logs: Centralized logging (ELK stack ou cloud)
- Alerts: PagerDuty/Slack pour incidents critiques

#### **Maintenance Préventive:**

- Backups: Automatiques daily/weekly/monthly
- Updates: Security patches automatiques OS/stack
- Testing: Synthetic monitoring widgets AIPI
- Scaling: Auto-scaling rules selon métriques

## **DISCLAIMERS ET CONSIDÉRATIONS LÉGALES**

#### **Performance et Garanties:**

Les performances indiquées sont basées sur tests réels mais peuvent varier selon le contenu, le trafic, et la configuration spécifique de chaque application.

Les hébergeurs partagés NE garantissent PAS d'allocations fixes de ressources CPU/RAM.

Les spécifications sont des estimations basées sur usage typique et peuvent nécessiter ajustements selon charge réelle.

#### **Responsabilités et Support:**

AIPI fournit documentation technique mais la configuration serveur reste responsabilité client/hébergeur.

Pour assistance technique détaillée: [support@aipi.com](mailto:support@aipi.com)

Documentation développeur complète: [docs.aipi.com](https://docs.aipi.com)

GitHub exemples intégration: [github.com/aipi-examples](https://github.com/aipi-examples)

