

COMMENT

# ACCÉLÉRER

VOTRE SITE

# WORDPRESS



Brian Jackson

**KINSTA**

Hébergement WordPress Infogéré



# Comment accélérer votre site WordPress

(GUIDE ULTIME)



Pour la version toujours à jour, scannez le QR code ci-dessus ou allez ici :  
<https://kinsta.com/fr/accellerer-wordpress/>

Publié par Kinsta

Traduction Française par Christophe Asselin de Beauville

Design du livre par Maja Szakadat



## À propos de l'auteur - Brian Jackson

Brian est le directeur du marketing chez Kinsta. Il se concentre sur tout, du développement de nouvelles stratégies de croissance en ligne à la création de contenu, en passant par le SEO et la sensibilisation au sein de la communauté. Il aime beaucoup écrire des articles de fond et a publié plus de 2 500 articles de blog.

L'une des plus grandes passions de Brian est WordPress, qu'il utilise depuis plus d'une décennie. Ayant travaillé à la fois dans l'industrie du CDN et maintenant dans le secteur de l'hébergement haute performance, il se spécialise dans l'optimisation de la performance Web et les moyens stratégiques d'accélérer les sites Web.

Comment accélérer votre site WordPress

# Sommaire

**10**

Types de sites  
WordPress : Statique  
ou Dynamique

**13**

Choisir un hébergement  
WordPress haute  
performance

**25**

Choisir un serveur  
proche de vos  
visiteurs

**34**

Les DNS Premium  
sont meilleurs que les  
DNS gratuits

**39**

Votre thème  
WordPress est  
important

**47**

Le point sur  
les plugins  
WordPress

**52**

Les paramètres  
WordPress  
optimaux

**58**

Pourquoi  
le cache est-il  
si important

**69**

L'optimisation  
des images est  
indispensable

**78**

Optimiser  
votre base  
de données

**97**

Utiliser un réseau  
de diffusion de contenu  
(CDN)

**107**

Décharger vos  
médias et vos emails  
en cas de besoin

**112**

Comment trouver  
les blocages et les  
plugins lents

**132**

Conseils sur  
l'optimisation  
du Back-End

**147**

Conseils sur  
l'optimisation  
du Front-End

**176**

Toujours optimiser  
avec le Mobile-First  
en tête



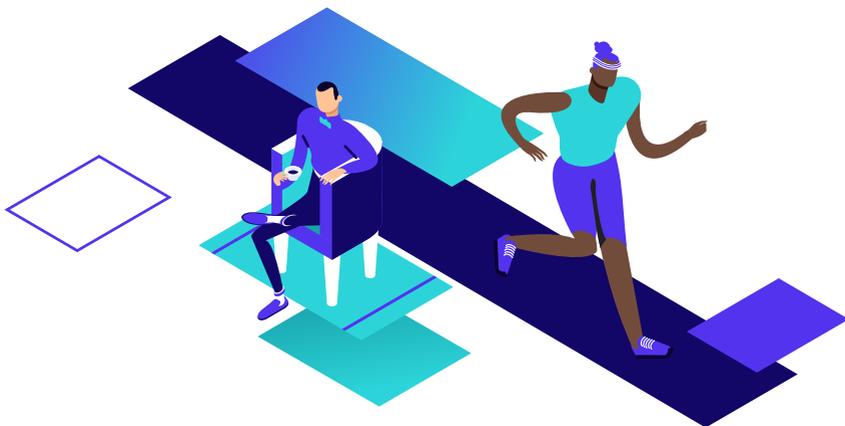
Nous avons publié de nombreux tutoriels au fil des ans avec des moyens d'optimiser et d'accélérer WordPress. Mais parfois, il peut être déroutant d'essayer de trouver tout ce dont vous avez besoin dans un seul endroit. Aujourd'hui, nous allons donc partager avec vous tout ce que nous savons sur l'optimisation et l'accélération de WordPress, plus de 15 ans d'expérience et de dures leçons apprises, le tout dans un guide ultime. Que vous commenciez tout juste à utiliser WordPress ou que vous soyez un développeur chevronné, nous vous promettons que vous trouverez quelque chose d'utile dans ce guide !

**Plus d'un tiers du Web est maintenant propulsé par WordPress.** Bien que cela soit génial, cela signifie aussi qu'il y a des milliers de thèmes, de plugins et de technologies différents qui doivent tous coexister. Pour l'utilisateur WordPress de tous les jours, cela peut rapidement se transformer en un cauchemar lorsque son site commence à se dégrader et qu'il ne sait pas pourquoi, ni même où commencer à regarder pour corriger les soucis.

Aujourd'hui, nous allons nous plonger dans les mesures applicables que vous pouvez prendre dès maintenant pour voir des améliorations sur vos propres sites WordPress. Nous partagerons également quelques ressources qui nous ont été précieuses.

CHAPITRE 01:

# Types de sites WordPress : Statique ou Dynamique



**Avant de nous plonger dans les optimisations, il est important de comprendre que tous les sites WordPress ne sont pas les mêmes. C'est pourquoi beaucoup d'utilisateurs ont des problèmes, car vous ne pouvez pas aborder chaque problème de la même manière. Nous donnons toujours aux sites WordPress une classification : statique ou dynamique. Examinons d'abord les différences entre ces deux types de sites.**

## **La plupart des sites sont statiques**

Les sites statiques comprennent généralement les sites tels que les blogs, les sites de petites entreprises, les sites de nouvelles à faible volume, les sites personnels, les sites de photographie, etc. Par statique, nous entendons que les données sur ces sites WordPress ne changent pas très souvent (peut-être quelques fois par jour). Même une bonne partie de notre site Kinsta serait considérée comme un site statique.

Cela devient d'autant plus important que de nombreuses requêtes peuvent être servies directement depuis le cache du serveur à une vitesse fulgurante ! Ne vous inquiétez pas, nous aborderons le sujet de la mise en cache plus loin dans ce qui suit. Cela signifie qu'ils auront moins d'appels de base de données et pas autant de ressources seront nécessaires pour atteindre les performances de google.

## Les sites très dynamiques

D'un autre côté, nous avons des sites très dynamiques. Il s'agit notamment des boutiques eCommerce (WooCommerce ou Easy Digital Downloads), les sites de communauté, les sites d'adhésion, les forums (bbPress ou BuddyPress) et les systèmes de gestion de l'apprentissage (LMS). Par dynamique, nous entendons que les données de ces sites WordPress **changent fréquemment** (les échanges serveur ont lieu toutes les quelques minutes ou même toutes les secondes). Cela signifie que toutes les requêtes adressées au serveur ne peuvent pas être servies directement à partir du cache et nécessitent des ressources serveur et des requêtes de base de données supplémentaires.

Ces sites ont aussi généralement **un grand nombre de visiteurs et de sessions simultanées**. Sur un site WordPress d'information ou d'entreprise qui est le plus souvent statique, un visiteur peut rester cinq ou dix minutes jusqu'à ce qu'il trouve ce dont il a besoin (et c'est un nombre élevé, généralement les taux de rebond sont beaucoup plus élevés). Sur les sites dynamiques, c'est le contraire qui se produit. Les visiteurs viennent généralement sur le site pour s'engager avec quelque chose ou quelqu'un. S'ils suivent un cours en ligne, il n'est pas rare qu'ils y restent pendant des heures.

Vous pouvez voir où cela nous mène. Les visiteurs connectés simultanément à votre hébergement WordPress s'additionnent rapidement. Pour empirer les choses, vous avez alors un grand nombre de visiteurs simultanés en plus d'un problème de "contenu non cachable".

***Vous ne pouvez pas traiter tous les sites WordPress de la même façon quand il s'agit de performances. Les sites statiques et très dynamiques sont deux bêtes très différentes.***

CHAPITRE 02:

# Choisir un hébergement WordPress haute performance



Un hébergeur WordPress est une entreprise qui stocke toutes les données de votre site Web. Vous vous inscrivez à un plan et toutes vos images, vos contenus, vos vidéos, etc., résident sur un serveur situé dans le centre de données de l'hébergeur. L'hébergeur WordPress vous permet d'accéder facilement aux données, de les gérer et de les acheminer vers vos visiteurs. Plutôt simple, non ? Eh bien, pas tout à fait.

Il existe trois types d'hébergeurs WordPress très différents que vous rencontrerez sur le Web. Analysons les avantages et les inconvénients de chacun. Il est important que vous choisissiez le bon dès le début, sinon, vous vous causerez simplement des maux de tête et vous perdrez du temps en cours de route.

## 1. Hébergement WordPress mutualisé

Le premier et le plus populaire type d'hébergement WordPress est ce que nous appelons "l'hébergement mutualisé". Il s'agit notamment des plus grands hébergeurs de l'industrie tels que les sociétés EIG comme Bluehost et HostGator ainsi que des fournisseurs comme Siteground, GoDaddy, et InMotion Hosting. Ils utilisent généralement cPanel, et le client moyen paie habituellement entre 3 \$ et 25 \$ par mois.

Quiconque utilise ce type d'hébergement connaîtra à un moment donné la lenteur, ce n'est qu'une question de temps. Pourquoi ? Parce **que les hébergeurs mutualisés ont**

**tendance à surpeupler leurs serveurs**, ce qui peut à son tour affecter les performances de votre site. Les suspensions de site ou les fréquentes erreurs 500 sont des choses courantes que vous rencontrerez car ces hébergeurs doivent imposer des limites à tout et restreindre les ressources pour survivre. Ou pire encore, les indisponibilités de votre site Web. Même si vous ne le savez pas, votre site WordPress est très probablement sur le même serveur que plus de 200 autres personnes. Tous les problèmes qui surgissent avec d'autres sites peuvent se répercuter sur votre site.



Hébergement WordPress Mutualisé

Peu importe la façon dont vous faites le calcul, après les dépenses, 3 \$ par mois ne génère aucun revenu pour la société d'hébergement. Surtout quand vous ajoutez du support à cela. Un ticket de support et ils sont déjà dans le rouge. La façon dont ils gagnent beaucoup d'argent, c'est grâce à la vente incitative et aux frais cachés. Ces ventes incitatives comprennent les migrations, les enregistrements de domaines, les certificats SSL, etc. Une autre tactique courante consiste à offrir d'énormes rabais d'inscription. Mais une fois que le renouvellement arrive, vous obtenez la vraie facture.

La plupart de ces hébergeurs offrent ce qu'ils appellent leur plan "ressources illimitées". Vous l'avez probablement tous vu. Dans le monde réel, il n'y a pas de ressources illimitées. Ce que les hébergeurs font dans les coulisses, c'est limiter les clients qui consomment une grande partie des ressources. Cela a pour effet de faire partir les clients en colère et de faire de la place à d'autres clients qui n'utilisent pas

beaucoup de ressources. En fin de compte, vous avez un cercle vicieux de la société d'hébergement proposant des plans bon marché et l'inscription des clients qui, ils espèrent, ne vont pas utiliser beaucoup de ressources et vont acheter des options supplémentaires et passer sur des offres plus chères.

Le service client et le support avec un hébergement mutualisé sont presque toujours inférieurs en raison du volume de sites par rapport aux représentants du support. Les hébergeurs mutualisés doivent s'éparpiller très finement pour faire un profit, ce qui conduit généralement à une expérience désagréable pour le client.



*Quand il s'agit d'hébergement mutualisé, vous en avez généralement pour votre argent.*

## 2. Hébergement WordPress VPS DIY

Le deuxième type d'hébergement WordPress est le VPS DIY, ou « Faites-le vous-même sur un serveur privé virtuel ». Cette assemblée est typiquement composée de startups bootstrap et d'utilisateurs avec un peu plus d'expérience dans le développement, la gestion de serveurs et WordPress. Ce sont des bricoleurs. Ces gens essaient encore généralement d'économiser de l'argent, mais ils se préoccupent aussi habituellement du rendement et réalisent l'importance de ce dernier dans le succès de leur entreprise. Les configurations communes peuvent inclure l'utilisation d'un fournisseur de VPS tiers tel que Digital Ocean, Linode ou Vultr, ainsi qu'un outil comme ServerPilot pour le gérer plus facilement.

Un petit VPS de DigitalOcean commence à 5 \$ par mois et le plan populaire chez ServerPilot commence à 10 \$ par mois. Donc, selon votre configuration, vous pourriez avoir à déboursier entre 5 \$ et 15 \$ ou plus par mois. L'approche du DIY peut réduire les coûts, mais elle signifie aussi que vous êtes responsable en cas de panne et de l'optimisation des performances de votre serveur.

L'approche du DIY peut être excellente, mais elle peut aussi se retourner contre vous si vous n'y prenez pas garde. N'empruntez pas cette voie si vous n'êtes pas un technicien averti ou simplement parce que vous voulez bricoler ! Votre temps vaut de l'argent et vous devriez le consacrer à la croissance de votre entreprise.

### 3. Hébergement WordPress infogéré

Le troisième type d'hébergement est ce que nous offrons chez Kinsta et c'est l'hébergement infogéré WordPress. Ces types d'hébergeurs s'occupent de toutes les tâches liées au back-end du serveur pour vous, en plus de vous fournir du support quand vous en avez besoin. Ils sont généralement optimisés pour fonctionner avec WordPress et incluent généralement des fonctionnalités telles que des environnements de développement (staging) en un clic et des sauvegardes automatiques. Leurs équipes de support seront mieux informées lorsqu'il s'agira de se familiariser avec le CMS puisqu'elles se concentrent quotidiennement sur une seule plateforme.

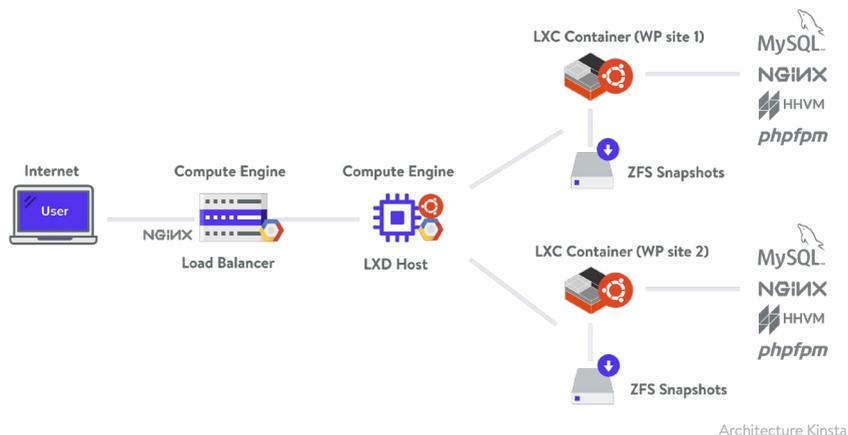
***Si vous voulez gagner du temps, l'hébergement infogéré WordPress est la solution qu'il vous faut !***

Les forfaits d'hébergement WordPress infogéré varient généralement de 25 \$ à 150 \$ par mois ou plus, selon la taille de votre site et vos besoins. De grandes entreprises comme jQuery, Intuit, Plesk, Dyn, NGINX, et même La Maison Blanche, utilisent toutes WordPress pour héberger leur site Web. Certains hébergeurs WordPress populaires que vous connaissez probablement ou que vous utilisez actuellement incluent WP Engine, Flywheel, Pressable, Media Temple, Pressidium, et Pagely.

### Kinsta adopte une approche différente

Kinsta, cependant, porte l'hébergement WordPress infogéré à un niveau supérieur. Notre plateforme d'hébergement n'entre dans aucune des catégories d'hébergement traditionnelles. Toute notre infrastructure est construite sur la plateforme Google Cloud et est différente des infrastructures traditionnelles mutualisées, VPS ou dédiées.

Chaque site WordPress sur notre plateforme fonctionne dans un conteneur logiciel isolé qui contient toutes les ressources logicielles nécessaires au fonctionnement du site (Linux, NGINX, PHP, MySQL). Cela signifie que le logiciel qui exécute chaque site est entièrement privé et n'est pas partagé, pas même entre vos propres sites.



Chaque conteneur de site fonctionne sur des machines virtuelles dans l'un des nombreux centres de données de GC. Chaque machine possède jusqu'à 96 CPU et des centaines de Go de RAM. Les ressources matérielles (RAM/CPU) sont allouées automatiquement à chaque conteneur de site par nos machines virtuelles en fonction des besoins (une fonction intéressante que nous appelons la mise à l'échelle automatique).

Chaque année, Review Signal publie ses benchmarks de performance d'hébergement WordPress, et nous sommes fiers que Kinsta soit la meilleure entreprise à tous les niveaux pour la cinquième année consécutive ! Et pas seulement sur un ou deux de nos plans, mais sur tous les plans, de Starter jusqu'à Entreprise.

Kinsta a eu essentiellement des tests LoadStorm et Blitz parfaits. De plus, ils n'avaient aucun défaut dans les autres tests. Je suis à court de mots pour louer leur performance.



Kevin Ohashi  
Fondateur et Consultant WP, ReviewSignal

Nous n'avons pas non plus de représentants de support de niveau 1 ou 2. Toute notre équipe de support est composée de développeurs WordPress et d'ingénieurs d'hébergement Linux dont beaucoup ont géré leurs propres serveurs, créé des thèmes et des plugins, et ont contribué au cœur de WordPress. Vous êtes ainsi assuré de recevoir des conseils d'experts de la part de quelqu'un qui utilise et développe activement WordPress.

Vous pouvez discuter avec les mêmes membres de l'équipe de support qui soutiennent nos clients du Fortune 500 et les clients entreprise. Nous sommes tellement exigeants quant à la qualité de notre équipe de support que nous n'embauchons que moins de 1 % des candidats qui présentent une demande. Vous ne trouverez pas de meilleur support ailleurs !

Pour en savoir plus sur les raisons pour lesquelles vous devriez choisir Kinsta pour l'hébergement WordPress infogéré, lisez pourquoi nous – en quoi Kinsta est différent. Mais quel que soit le fournisseur d'hébergement que vous choisissiez, vous devriez toujours rechercher les caractéristiques de serveur suivantes pour vous assurer que votre site Web fonctionne aussi rapidement que possible.

Avec WP Engine, les problèmes de base sont généralement réglés rapidement. Toutefois, pour toutes les questions complexes, une résolution prendra un certain temps, et il y aura beaucoup de va-et-vient. C'est un problème lorsque vous utilisez un site WordPress haut de gamme et qu'il y a un problème urgent qui doit être traité rapidement. Si vous demandez ma seule recommandation entre les deux, à mon avis, Kinsta est le meilleur. Ils offrent beaucoup plus que ce qu'ils promettent. Vous n'avez jamais à vous soucier de la lenteur du site, des temps d'indisponibilité, de l'obtention d'un support de qualité, ou de tout autre problème lié à l'hébergement.

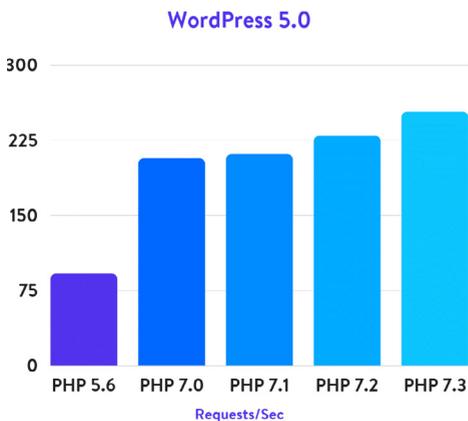


Harsh Agrawal  
Blogger Récompensé, ShoutMeLoud

# PHP 7 ou supérieur pour les meilleures performances

PHP est un langage de script et de programmation open-source, côté serveur, qui est principalement utilisé pour le développement web. La majeure partie du logiciel WordPress est écrite en PHP, avec vos plugins et vos thèmes, ce qui fait de PHP un langage très important pour la communauté WordPress. Vous devez vous assurer que votre hébergeur WordPress offre au moins PHP 7 ou supérieur.

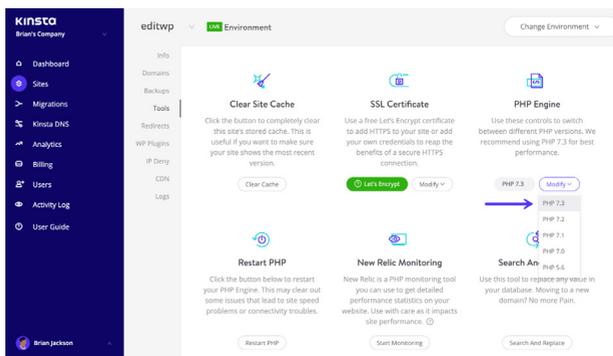
Il existe différentes versions de PHP que votre hébergeur vous fournira sur votre serveur, avec le nouveau PHP 7.3 offrant d'énormes améliorations de performances.



En fait, dans nos récents benchmarks PHP, si vous comparez PHP 7.3 à PHP 5.6, il peut traiter 3x plus de requêtes (transactions) par seconde ! PHP 7.3 est aussi environ 9% plus rapide que PHP 7.2. Cela peut également avoir un impact sur la réactivité de votre tableau de bord d'administration WordPress.

Des vitesses plus rapides et une sécurité améliorée, c'est pourquoi Kinsta propose toujours les versions les plus récentes de PHP. Vous pouvez changer de version de PHP en un seul clic.

Et méfiez-vous des hébergeurs WordPress qui offrent le HHVM comme alternative à PHP. Le HHVM n'est plus une solution adaptée à l'hébergement WordPress.



## Choisissez un hébergeur qui utilise **NGINX**

Dans les coulisses, chaque hébergeur WordPress utilise un serveur Web pour propulser vos sites WordPress. Les choix les plus courants sont NGINX et Apache.

Nous recommandons fortement d'opter pour un hébergeur qui utilise NGINX en raison de ses racines dans le domaine de l'optimisation des performances. NGINX surclasse souvent d'autres serveurs Web populaires dans les tests de référence, en particulier dans les situations avec un contenu statique ou des requêtes simultanées élevées, c'est pourquoi Kinsta utilise NGINX.

Parmi les entreprises de renom qui utilisent NGINX figurent Autodesk, Atlassian, Intuit, T-Mobile, GitLab, DuckDuckGo, Microsoft, IBM, Google, Adobe, Salesforce, VMWare, Xerox, LinkedIn, Cisco, Facebook, Target, Citrix Systems, Twitter, Apple, Intel et beaucoup plus. (source: Siftery.com)

Selon W3Techs, Apache alimente 44,0% de tous les sites Web, ce qui en fait l'option la plus utilisée. Mais si vous regardez le serveur web le plus populaire parmi les sites web à fort trafic (top 10,000), NGINX en a 41.9% alors que Apache en a 18.1%. Il est utilisé par certains des sites les plus gourmands en ressources existantes, dont Netflix, la NASA et même WordPress.com.

## Le réseau de votre hébergeur est important

Lorsque vous choisissez un hébergeur WordPress, vous ne pensez peut-être même pas à demander ou à faire des recherches sur le réseau qu'ils utilisent, mais vous devriez le faire. Le réseau peut avoir un impact énorme sur les performances de votre site et même sur la rapidité de votre tableau de bord WordPress. Beaucoup d'hébergeurs laisseront cela en dehors de leur marketing car ils opteront pour le réseau le moins cher afin de réduire les coûts.

Voici quelques questions que vous devriez poser :

- **Sur quels réseaux transmettez-vous des données ?** Est-ce que la majorité

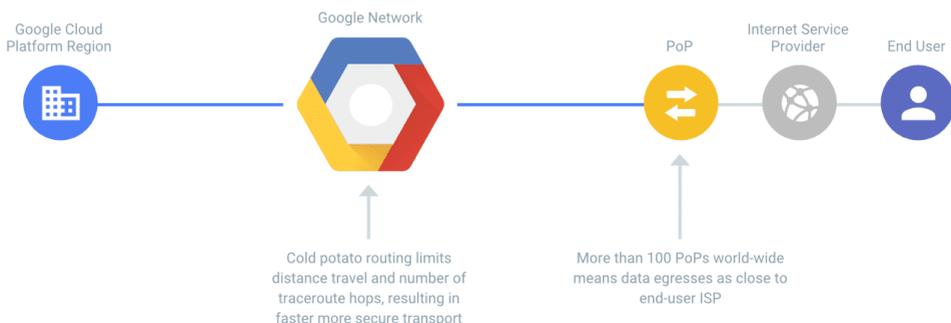
d'entre eux passent par des réseaux de FAI publics ou des infrastructures privées telles que Google ou Microsoft ? Ces grands fournisseurs ont des réseaux qui sont construits et optimisés pour une latence faible et la vitesse. Ils ont même leurs propres câbles Internet sous l'océan !

- **Les réseaux que vous utilisez sont-ils redondants ?** Que se passe-t-il si un câble est coupé accidentellement ? Cela arrive plus souvent que vous ne le pensez.

Retour en 2017, Google a annoncé leur réseau de niveau standard, qui est un réseau plus lent, mais à un coût moindre. Chez Kinsta, nous utilisons leur **réseau de niveau supérieur** pour tous nos plans d'hébergement. Bien qu'il s'agisse d'un coût supplémentaire pour nous, cela vous assure une vitesse fulgurante.

***Kinsta utilise le réseau Premium de la plateforme Google Cloud. Parce que vos sites Web méritent ce qu'il y a de mieux.***

Selon Google, le réseau premium permet d'améliorer les performances du réseau en réduisant la durée des déplacements sur Internet ; les paquets entrent (et sortent) du réseau de Google aussi près que possible de l'utilisateur et voyagent ensuite sur le backbone de Google avant d'arriver à la VM. Le niveau standard achemine le trafic sortant de la GCP (Plateforme Google Cloud) vers Internet par l'intermédiaire des réseaux de transport commun (ISP) plutôt que par le réseau de Google.



Réseau Gcp Premium

En d'autres termes pour que ce soit plus facile à comprendre :

- **Les paquets de niveau Premium passent plus de temps sur le réseau de Google,** avec moins de rebondissements, et sont donc plus performants (mais coûtent plus cher).
- Les paquets de niveau standard passent moins de temps sur le réseau de Google, et plus de temps à jouer à la patate chaude sur les réseaux publics, et donc, sont moins performants (mais moins chers).

Dans quelle mesure cela a-t-il un impact ? Eh bien, pour les données voyageant à travers les continents, leur réseau de niveau premium est en moyenne **41% plus rapide** que le réseau de niveau standard. Pour les données voyageant vers une région voisine (même continent), le niveau premium est environ 8% plus rapide. Alors que la mise en réseau ne représente qu'une fraction du temps total de chargement de vos pages, chaque milliseconde s'additionne !

***Pour les données voyageant d'un continent à l'autre, le réseau premium de Google Cloud est en moyenne 41% plus rapide !***

La redondance est également essentielle, et c'est pourquoi Google utilise au moins trois chemins indépendants (redondance N+2) entre deux sites du réseau Google, ce qui permet de garantir que le trafic continue à circuler entre les sites même en cas d'interruption.

Comme vous pouvez probablement le constater maintenant, il se passe beaucoup de choses dans les coulisses lorsqu'il s'agit de réseau. Assurez-vous que votre hébergeur WordPress en utilise un bien réputé et qu'il n'opte pas pour des réseaux de niveau inférieur afin de réduire les coûts.

## Le HTTP/2 est un incontournable

HTTP/2 est un protocole Web publié en 2015 qui a été conçu pour accélérer la livraison des sites Web. En raison de la prise en charge du navigateur, il nécessite HTTPS (SSL). Si votre hébergeur WordPress ne supporte pas HTTP/2, vous devriez commencer à en chercher un nouveau. Avec le passage de l'ensemble du web au HTTPS, ce n'est plus seulement une fonctionnalité agréable à avoir, c'est une nécessité.

L'amélioration des performances avec HTTP/2 est due à une variété de raisons telles que la prise en charge d'un meilleur multiplexage, le parallélisme, la compression HPACK avec codage Huffman, l'extension ALPN, et le serveur push. Il y avait auparavant pas mal de TLS surchargés lorsqu'il s'agissait de passer sur HTTPS, mais c'est maintenant beaucoup moins le cas grâce à HTTP/2 et TLS 1.3. **Kinsta supporte le HTTP/2 et TLS 1.3 sur tous nos serveurs et notre CDN.**

Une autre grande victoire avec HTTP/2 est qu'avec la plupart des sites WordPress, vous n'avez plus à vous soucier de la concaténation (combinaison de fichiers) ou du partage de domaines. Ces optimisations sont aujourd'hui obsolètes !

CHAPITRE 03:

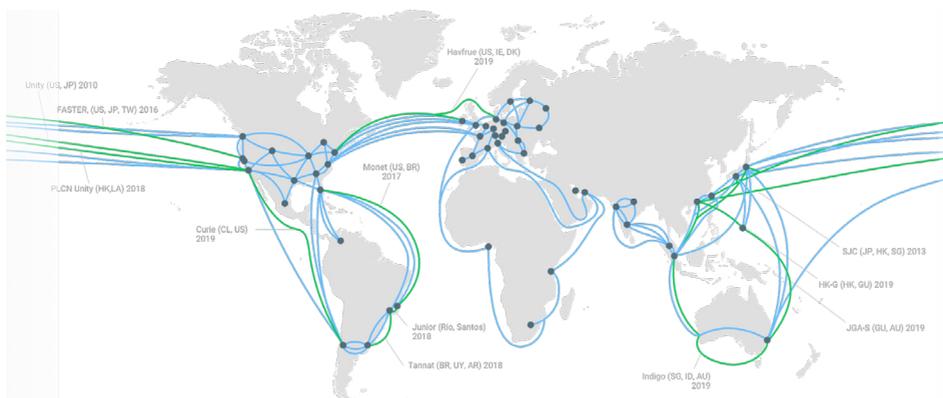
# Choisir un serveur proche de vos visiteurs



**L'une des toutes premières choses que vous devez faire lorsque vous hébergez votre site WordPress est de déterminer d'où vient la majorité de vos visiteurs ou clients. Pourquoi est-ce important ? Parce que l'emplacement où vous hébergez votre site Web joue un rôle important dans la détermination de la latence globale de votre réseau et du TTFB. Il a également un impact sur les vitesses SFTP et la réactivité du tableau de bord d'administration de WordPress.**

**Latence du réseau :** Il s'agit de la durée ou du délai de transmission des données sur un réseau. En d'autres termes, le temps qu'il faut pour qu'un paquet de données passe d'un point à un autre. De nos jours, cela se mesure généralement en millisecondes, mais cela peut prendre quelques secondes selon le réseau. Plus on se rapproche de zéro, mieux c'est.

En d'autres termes, plus la distance est grande, plus la latence est grande, ce qui équivaut à un délai plus long. Vous ne pouvez pas battre la vitesse de la lumière quand il s'agit de données voyageant sur des câbles à fibre optique. Par exemple, vous trouverez ci-dessous une carte du réseau de la plateforme Google Cloud. Ils ont des milliers de kilomètres de câbles qui passent sous l'océan et couvrent le globe entier. C'est l'une des raisons pour lesquelles c'est l'un des réseaux les plus rapides au monde. Et même avec ce vaste réseau à la fine pointe de la technologie, il y aura toujours des délais de latence.



Il y a quelques facteurs différents qui contribuent aux délais de latence du réseau, notamment :

- **Délai de transmission** : Différents types de supports, tels que les connexions sans fil ou à fibre optique, introduisent tous un certain délai, car ils ne peuvent pousser qu'un nombre limité de bits. Par exemple, une connexion à fibre optique aura un délai de transmission plus court qu'une ligne T1. Vous devez également tenir compte de la taille du paquet.
- **Délai de propagation** : C'est le temps qu'il faut pour que le paquet de données voyage. Il peut s'agir de la distance, des délais de mise en réseau, etc.
- **Délai de la file d'attente** : Si la bande passante est dépassée, il peut y avoir des délais de file d'attente pendant lesquels les données doivent attendre à l'hôte ou au routeur. Ceci peut être affecté par la congestion du réseau.

Certains pourraient soutenir que la latence du réseau n'est pas très importante, mais elle peut l'être par rapport à l'endroit où vous choisissez d'héberger votre site WordPress. Pour les médias et vos ressources (telles que les images, le JavaScript, le CSS, les Vidéos), un CDN peut corriger une grande partie de la latence supplémentaire impliquée, en délivrant une copie depuis un serveur plus proche. Cependant, dans la plupart des configurations, vous devez toujours interroger votre serveur d'hébergement pour connaître la charge DOC initiale. Et c'est pourquoi la latence est importante !

La latence est particulièrement importante lorsqu'il s'agit d'entreprises qui desservent des visiteurs dans un lieu géographique précis. Par exemple, disons que vous avez une boutique e-commerce à Sydney et que 90 % de vos clients viennent d'Australie. Votre entreprise aurait avantage à placer votre site sur un serveur en Australie, plutôt que de l'héberger en Europe ou aux États-Unis.

**TTFB** : C'est l'abréviation de Time To First Byte. En termes simples, il s'agit d'une mesure du temps d'attente du navigateur avant de recevoir son premier octet de données du serveur. Plus il faut de temps pour obtenir ces données, plus il faut de temps pour afficher votre page. Encore une fois, plus on se rapproche de zéro, mieux c'est.

Une idée fausse courante est que ce calcul est effectué après les temps de consultation DNS, mais le calcul initial du TTFB en réseau inclut toujours la latence du réseau. Il s'agit d'un processus en trois étapes et des délais ainsi que des latences peuvent survenir n'importe où entre les deux, ce qui s'ajoute à votre TTFB total.

Connection Setup		TIME
Queueing		4.56 ms
Stalled		1.35 ms
Request/Response		TIME
Request sent		0.11 ms
Waiting (TTFB)		291.39 ms
Content Download		1.47 ms
<b>Explanation</b>		<b>298.87 ms</b>

**1. Requête au serveur** : Lorsque quelqu'un visite votre site Web, la première chose qui se produit est une requête HTTP envoyée par le client (navigateur) au serveur. Au cours de cette étape, divers facteurs peuvent entraîner des délais. La lenteur des temps de recherche DNS peut contribuer à augmenter le temps nécessaire à la requête. Si le serveur est géographiquement éloigné, cela peut introduire une latence dans la distance que les données doivent parcourir. De plus, si vous avez des règles de pare-feu complexes, cela peut augmenter le temps de routage. Et n'oubliez pas la vitesse internet du client.

**2. Traitement du serveur :** Après l'envoi de la demande, le serveur doit maintenant la traiter et générer une réponse. Cela pourrait introduire un certain nombre de délais différents tels que des appels de base de données lents, trop de scripts tiers, ne pas mettre en cache votre première réponse, un code ou un thème WordPress mal optimisé, et des ressources serveur inefficaces telles que les I/O du disque ou la mémoire.

**3. Réponse au client :** Une fois que le serveur a traité la requête, il doit la renvoyer au client (ou plutôt renvoyer le premier octet). Ceci est fortement affecté à la fois par la vitesse du réseau du serveur et du client. Si le client a un accès Internet lent à partir d'un point d'accès Wi-Fi, cela va se refléter dans le TTFB.

En général, **tout ce qui est inférieur à 100 ms est un excellent et bon TTFB**. Google PageSpeed Insights recommande moins de 200 ms pour le temps de réponse du serveur. Si vous êtes dans la fourchette des 300-500 ms, c'est assez standard. Et si vous avez plus de 600 ms, il se peut que vous ayez quelque chose de mal configuré sur votre serveur ou qu'il soit temps de passer à une meilleure pile web. Ou suivez nos recommandations ci-dessous sur la façon de réduire votre TTFB. Et n'oubliez pas que la négociation du SSL/TLS peut aussi être un facteur.

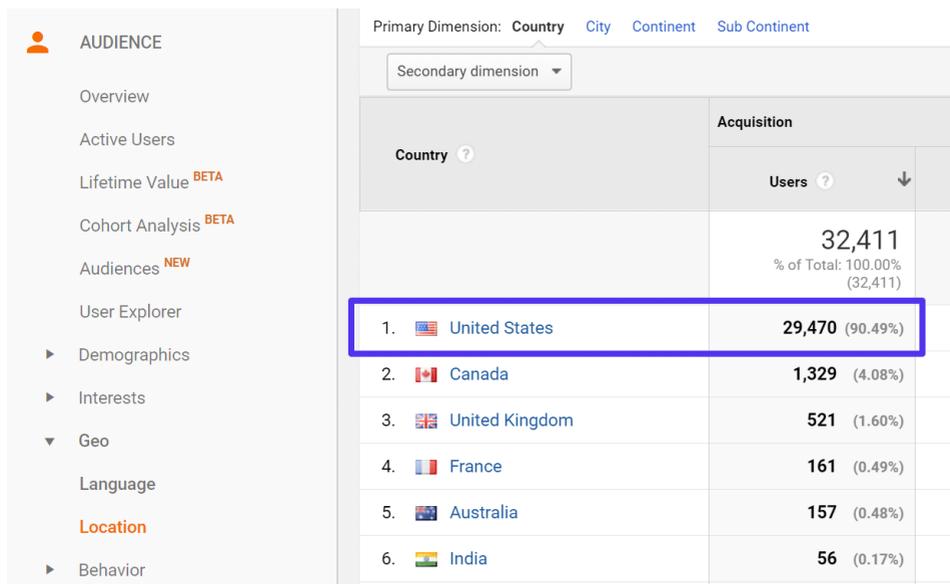
Nous n'allons pas vous ennuyer avec tous les détails techniques dans cet article, tout ce que vous avez besoin de savoir est que **vous voulez que votre latence réseau et votre TTFB soient aussi faibles que possible**. L'une des façons les plus simples d'y parvenir est de choisir le serveur le plus proche de vos visiteurs. Vous pouvez déterminer le meilleur emplacement en suivant les conseils ci-dessous.

## Astuce 1 – Vérifiez la géolocalisation de vos visiteurs dans Google Analytics

L'une des premières choses que vous pouvez faire est de regarder la géolocalisation de vos visiteurs dans Google Analytics. Vous la trouverez sous "Audience / Geo / Location."

Dans l'exemple ci-dessous, vous pouvez voir que plus de 90 % du trafic provient des

États-Unis. Ainsi, dans la plupart des cas, vous devrez placer votre site WordPress sur un serveur aux États-Unis. Vous pouvez également filtrer les données encore plus loin vers les villes. Ceci est particulièrement important si vous êtes une entreprise locale. Mais en général, nous recommandons un endroit central comme l'Iowa, aux États-Unis.



The screenshot shows the Google Analytics Audience report. The primary dimension is 'Country' and the secondary dimension is 'Acquisition'. The total number of users is 32,411, representing 100.00% of the total. The top countries are listed in a table below.

Country	Users	% of Total
1. United States	29,470	90.49%
2. Canada	1,329	4.08%
3. United Kingdom	521	1.60%
4. France	161	0.49%
5. Australia	157	0.48%
6. India	56	0.17%

## Astuce 2 – Vérifier les données de e-commerce

Si vous gérez une boutique eCommerce, assurez-vous également de vérifier d'où viennent vos clients. C'est bien sûr la façon dont vous générez des revenus, donc ce sont vos visiteurs les plus importants. Ceci devrait coïncider avec votre trafic ci-dessus ; cependant, ce n'est pas toujours le cas. Si vous avez des données eCommerce ou des objectifs dans Google Analytics, vous pouvez facilement superposer ces informations aux données de géolocalisation pour prendre une décision plus éclairée. Ou vérifiez les informations de localisation stockées dans la base de données de votre plateforme de eCommerce.

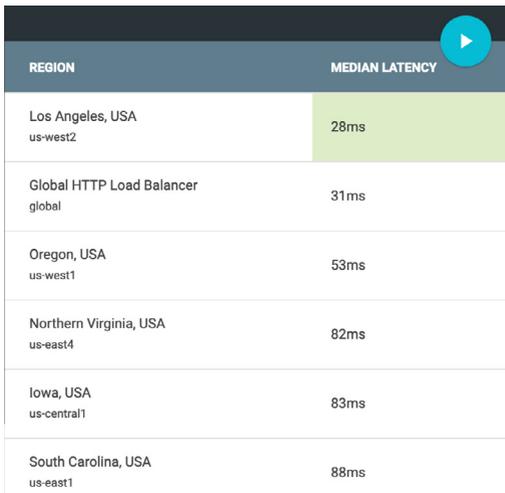
## Astuce 3 – Effectuez un test rapide de latence

Il existe de nombreux outils pratiques et gratuits pour mesurer la latence à partir de votre emplacement actuel pour différents fournisseurs de cloud computing. Cela peut vous aider à évaluer rapidement quelle région pourrait être le meilleur choix pour votre site.

- **gcping.com** : mesure de latence pour les régions de la plateforme Google Cloud, y compris les serveurs Kinsta.
- **cloudping.info** : mesure la latence dans les régions de Amazon Web Services.
- **azurespeed.com** : mesure la latence dans les régions de Azur.

Dans l'exemple ci-dessous, nous pouvons voir que l'Oregon, USA (us-ouest1) est le plus rapide d'où nous sommes situés. Cependant, si vous servez des clients dans l'ensemble des États-Unis, il serait peut-être préférable de choisir l'Iowa, États-Unis (us-central1) pour assurer une faible latence aux visiteurs de la côte ouest et de la côte est.

## Mesurer votre latence pour les régions de GCP



REGION	MEDIAN LATENCY
Los Angeles, USA us-west2	28ms
Global HTTP Load Balancer global	31ms
Oregon, USA us-west1	53ms
Northern Virginia, USA us-east4	82ms
Iowa, USA us-central1	83ms
South Carolina, USA us-east1	88ms

Chez Kinsta, nous offrons 20 centres de données différents à travers le monde. Vous pouvez facilement choisir un site qui aura à la fois une faible latence et un faible TTFB ! Cela permet également de réduire les sauts de réseau.



## Autres moyens de réduire la latence et le TTFB

En plus de choisir un emplacement de serveur proche, voici quelques autres façons de réduire la latence.

- Implémentez la mise en cache sur votre site WordPress. Dans nos tests, la mise en cache a réduit notre TTFB d'un énorme 90% !
- Utilisez un réseau de diffusion de contenu (CDN) pour servir les ressources mises en cache via des POPs du monde entier. Cela aide à annuler la latence du réseau pour les visiteurs qui pourraient ne pas être proches de votre serveur hôte.
- Profitez du protocole HTTP/2 pour minimiser le nombre d'allers-retours grâce à la parallélisation. HTTP/2 est activé sur tous les serveurs Kinsta.
- Réduisez le nombre de requêtes HTTP externes. Chacune d'entre elle peut avoir sa propre latence supplémentaire en fonction de l'emplacement de son serveur.
- Les DNS jouent un rôle dans le TTFB, vous devriez donc utiliser un fournisseur de DNS premium avec des temps de recherche rapides.

- Utilisez Prefetch et Prerender pour effectuer des tâches en coulisses pendant le chargement de la page.

Ne vous inquiétez pas, nous traiterons de toutes les recommandations mentionnées ci-dessus plus loin dans cet article.

## Vitesses du SFTP et du tableau de bord d'administration de WordPress

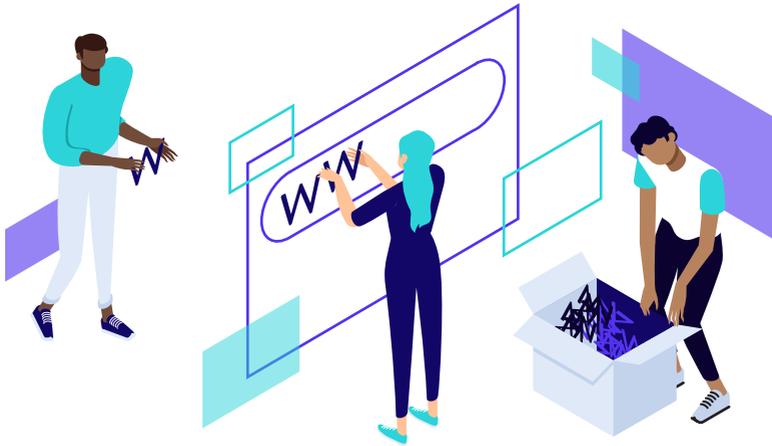
Vos visiteurs et clients devraient toujours être votre priorité. Mais un autre aspect de la performance dont beaucoup ne parlent pas est la façon dont certaines de ces décisions affectent votre travail quotidien. L'emplacement du centre de données que vous choisissez a un impact sur la vitesse de chargement et de téléchargement SFTP (transfert de fichiers avec un client FTP), ainsi que sur la réactivité de votre tableau de bord d'administration WordPress.

Ainsi, même si vous voulez vous rassurer et choisir l'emplacement qui convient le mieux à vos visiteurs, gardez à l'esprit que cela peut affecter la gestion du site. Des tâches comme le téléchargement de fichiers vers la médiathèque WordPress seront plus rapides lorsque votre site sera hébergé dans un centre de données plus proche de chez vous.

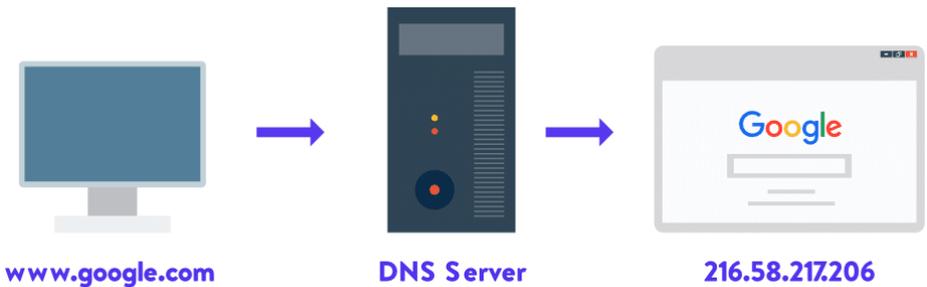
Nous entendons constamment les clients de Kinsta nous dire qu'ils sont surpris par la rapidité avec laquelle leur tableau de bord d'administration est chez nous. Il y a une multitude de facteurs qui influencent cela, mais avoir 20 centres de données différents, c'est énorme ! Choisissez un endroit qui convient à la fois à vos visiteurs et à vous ! Après tout, c'est vous qui allez probablement passer des milliers d'heures à travailler sur votre site Web.

CHAPITRE 04:

# Les DNS Premium sont meilleurs que les DNS gratuits



Les DNS, abréviation de Domain Name System, est l'un des composants les plus courants et les plus méconnus du paysage web. En termes simples, le DNS aide à diriger le trafic sur Internet en connectant les noms de domaine à des serveurs Web réels. Essentiellement, il faut une requête humaine – un nom de domaine comme `kinsta.com` – et cela se traduit en une adresse IP de serveur conviviale – comme `216.58.217.206`.



Comment Fonctionnent Les DNS

Vous pouvez trouver à la fois des DNS gratuits et des DNS premium. Tous les clients Kinsta ont accès au DNS premium via Amazon Route 53. Et en général, nous croyons que les DNS premium sont une nécessité dans le monde d'aujourd'hui.

L'une des principales raisons de choisir un DNS haut de gamme est la **vitesse et la fiabilité**. La recherche des enregistrements DNS et la direction du trafic prennent

du temps, même si ce n'est qu'une question de millisecondes.

Généralement, les DNS gratuits que vous obtiendrez de votre registraire de noms de domaine sont comparativement lents, alors que les DNS premium offrent souvent de meilleures performances. Par exemple, dans nos tests, nous avons trouvé que **les DNS gratuits de NameCheap était 33% plus lents que les DNS premium de Amazon Route 53**. De plus, les DNS premium peuvent offrir une meilleure sécurité et une meilleure disponibilité, en particulier lorsque vous êtes victime d'une attaque DDoS.

Vous pouvez utiliser un outil comme le test de vitesse SolveDNS pour vérifier vos temps de recherche DNS. DNSPerf fournit également d'excellentes données de performances sur tous les principaux fournisseurs de DNS.

Pour un bon compromis entre les DNS gratuits fournis par votre bureau d'enregistrement de domaine (Registraire) et les DNS premium, Cloudflare DNS est un service gratuit qui offre encore de nombreux avantages du DNS premium. Et ils explosent rapidement avec des temps de réponse moyens inférieurs à 20 ms dans le monde entier (comme on le voit ci-dessous).

Name Server	Result	Average (milliseconds)	Standard Deviation	95% Confidence Interval
<b>algin.ns.cloudflare.com</b>	<ul style="list-style-type: none"> <li>✓ Los Angeles: 5.65 ms</li> <li>✓ Dallas: 6.04 ms</li> <li>✓ New York: 3.47 ms</li> <li>✓ Singapore: 3.29 ms</li> <li>✓ London: 3.81 ms</li> <li>✓ Amsterdam: 4.04 ms</li> <li>✓ San Francisco: 12.4 ms</li> <li>✓ Sydney: 6.57 ms</li> <li>✓ Tokyo: 4.02 ms</li> </ul>	5.48 ms	2.69	[3.49, 7.47]
<b>heather.ns.cloudflare.com</b>	<ul style="list-style-type: none"> <li>✓ Los Angeles: 1.55 ms</li> <li>✓ Dallas: 3.98 ms</li> <li>✓ New York: 2.06 ms</li> <li>✓ Singapore: 2.11 ms</li> <li>✓ London: 2.95 ms</li> <li>✓ Amsterdam: 2.17 ms</li> <li>✓ San Francisco: 12.04 ms</li> <li>✓ Sydney: 1.51 ms</li> <li>✓ Tokyo: 2.62 ms</li> </ul>	3.44 ms	3.12	[1.13, 5.75]

Cependant, une mise en garde avec Cloudflare est qu'il a aussi plus de temps d'indisponibilité que beaucoup d'autres fournisseurs. Si vous servez principalement des visiteurs aux États-Unis, DNS Made Easy est un autre excellent fournisseur de DNS haut de gamme que vous voudrez peut-être consulter. Ils ont la réputation d'avoir fourni certains des meilleurs temps de disponibilité DNS au cours de la dernière décennie.

Au cours des 30 derniers jours, DNSPerf affiche le temps de disponibilité suivant de ces fournisseurs :

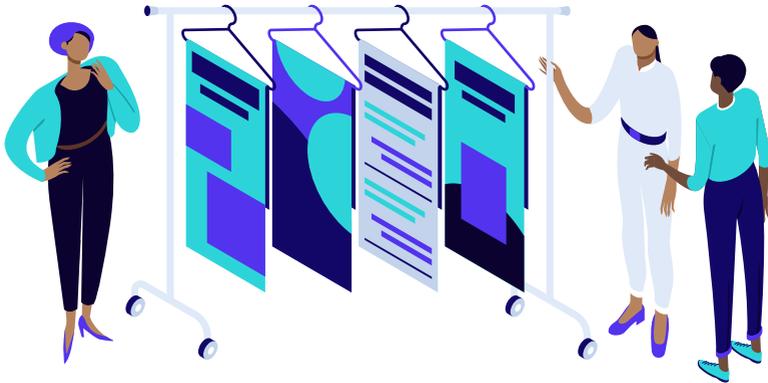
- DNS Made Easy : 99,99% ce qui équivaut à un 4m 23,0s de temps d'indisponibilité mensuel.
- Amazon Route 53 : 99,88% ce qui équivaut à 52m 35,7s de temps d'indisponibilité mensuel.
- Cloudflare : 99.85% ce qui équivaut à 1h 5m 44.6s de temps d'indisponibilité mensuel.

Les temps d'indisponibilité sont-ils si importants pour les fournisseurs DNS ?

La réponse à cette question est vraiment oui et non. Les DNS est généralement mis en cache avec les FAI en utilisant la valeur TTL (Time to live) sur l'enregistrement DNS. Par conséquent, si un fournisseur DNS tombe en panne pendant 10 minutes, vous ne remarquerez probablement rien. Le temps d'indisponibilité est cependant important si le fournisseur a constamment des pannes plus longues et fréquentes, ou si votre FAI et les enregistrements DNS utilisent tous les deux des valeurs TTL très basses.

CHAPITRE 05:

# Votre thème WordPress est important



**Tout le monde aime avoir un thème WordPress flambant neuf, mais soyez prudent avant de sortir et de vous emparer de celui avec toutes les nouvelles fonctionnalités brillantes. Tout d’abord, vous devriez consulter notre article sur les différences entre les thèmes gratuits et payants. En ce qui concerne la performance, chaque élément que vous voyez dans un thème a un impact sur la vitesse globale de votre site Web. Et malheureusement, avec des milliers de thèmes dans la nature, il y en a des bons et des mauvais.**

*Votre thème WordPress est important pour la performance. Choisissez le bon dès le départ.*

Alors comment êtes-vous censé savoir lequel choisir ? Nous vous recommandons d’opter pour l’une des deux options suivantes :

- Un thème WordPress rapide et léger qui est construit avec seulement les fonctionnalités dont vous avez besoin, rien de plus.
- Un thème WordPress plus riche en fonctionnalités, mais vous pouvez désactiver les fonctionnalités qui ne sont pas utilisées.

Des choses comme les polices Google, les icônes Font Awesome, les curseurs, les galeries, les scripts vidéo et parallax, etc. Ce ne sont là que quelques-unes des nombreuses choses que vous devriez pouvoir désactiver si vous ne les utilisez pas. Vous ne voulez pas essayer de les modifier manuellement après coup. Et nous n'allons pas vous montrer 50 façons différentes de dépouiller les choses. Au lieu de cela, vous devriez commencer ou passer à un thème WordPress qui est soit léger depuis le début ou vous donne ces options.

Vous trouverez ci-dessous quelques thèmes WordPress que nous recommandons et avec lesquels vous ne pouvez pas vous tromper ! Faites-nous confiance, vous nous remercirez plus tard.

Chaque thème mentionné ci-dessous est entièrement compatible avec WooCommerce et Easy Digital Downloads, WPML, BuddyPress et bbPress. Nous effectuons quelques tests de vitesse avec chaque thème en utilisant la configuration suivante :

- Hébergé sur Kinsta, sous WordPress 4.9.8
- PHP 7.3 et SSL (HTTPS)
- CDN Kinsta
- Imagify a été utilisé pour compresser automatiquement les images.

## GeneratePress

GeneratePress est un thème WordPress rapide, léger (moins de 1 Mo zippé), responsive pour les mobiles, conçu pour la vitesse, le référencement et la convivialité. Construit par Tom Osborne, un développeur du Canada. Il est activement mis à jour et bien soutenu. Même quelques membres de l'équipe Kinsta utilisent GeneratePress pour leurs projets.

Il existe une version gratuite et une version premium. Si vous jetez un coup d'œil au dépôt WordPress, la version gratuite compte actuellement plus de 200,000 installations actives, plus de 2 millions de téléchargements et une note impressionnante de 5 étoiles sur 5 (plus de 850 personnes lui ont attribué 5 étoiles).

L'un des avantages de GeneratePress est que toutes les options utilisent le Customizer natif de WordPress, ce qui signifie que vous pouvez voir chaque changement que vous faites instantanément avant d'appuyer sur le bouton de publication. Cela signifie également que vous n'avez pas besoin d'apprendre un nouveau panneau de contrôle de thème.

**GeneratePress** HOME PREMIUM SITE LIBRARY INSTALL HELP ▾ LOG IN

The perfect **lightweight** theme for your next project.

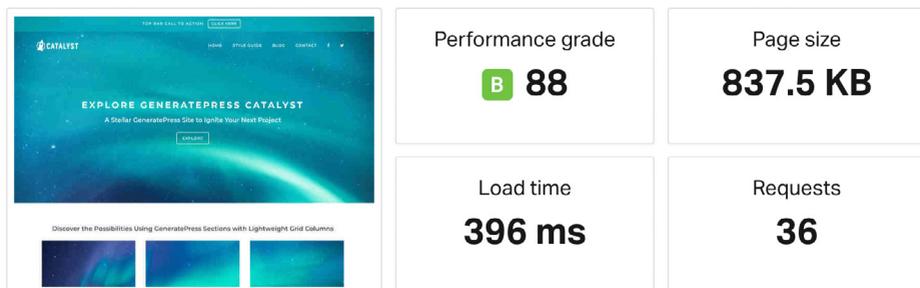
Install Premium

1,643,956+ Downloads 700+ ★★★★★ 100,000+ Active websites 40,000+ Happy customers

Mais c'est rapide à quel point ? Nous avons fait une nouvelle installation de GeneratePress, effectué cinq tests de vitesse dans Pingdom, et pris la moyenne. Le temps de chargement total était de **305 ms** avec une **taille de page totale de seulement 16,8 Ko**. Il est toujours bon d'avoir un test de base pour voir de quoi le thème est capable en termes de performance brute.

	Performance grade <b>A 97</b>	Page size <b>16.8 KB</b>
	Load time <b>305 ms</b>	Requests <b>7</b>

Nous avons ensuite effectué une autre série de tests avec l'un des thèmes pré-construits de la bibliothèque de sites GeneratePress. Il contient des images, des fonds d'écran, de nouvelles sections, etc. Un des avantages de GeneratePress est qu'il a beaucoup de thèmes pré-construits qui ne nécessitent pas de plugin de création de page. Vous pouvez voir qu'il a encore été chronométré à moins de **400 ms**.



Maintenant, bien sûr, dans un environnement réel, vous pouvez avoir d'autres choses en cours d'exécution comme Google Analytics, Facebook remarketing pixel, Hotjar, etc. Mais vous devriez pouvoir viser sous la barre des 1 seconde facilement. Jetez un coup d'œil à l'examen approfondi de GeneratePress sur Woorup.

Nous allons vous montrer plus de façons d'optimiser et d'accélérer WordPress plus loin.

## OceanWP

Le thème OceanWP est léger et extensible. Il vous permet de créer presque n'importe quel type de site web, tel qu'un blog, un portfolio, un site web d'affaires ou une vitrine WooCommerce avec un design beau et professionnel.

Construit par Nicolas Lecocq, il est aussi activement mis à jour et bien soutenu.

Tout comme avec GeneratePress, il existe une version gratuite et une version Premium. Si vous jetez un coup d'œil au dépôt WordPress, la version gratuite compte actuellement plus de 400,000 installations actives, et une autre note impressionnante de 5 étoiles sur 5 (plus de 2,600 personnes lui ont attribué 5 étoiles).

# Create Beautiful Websites

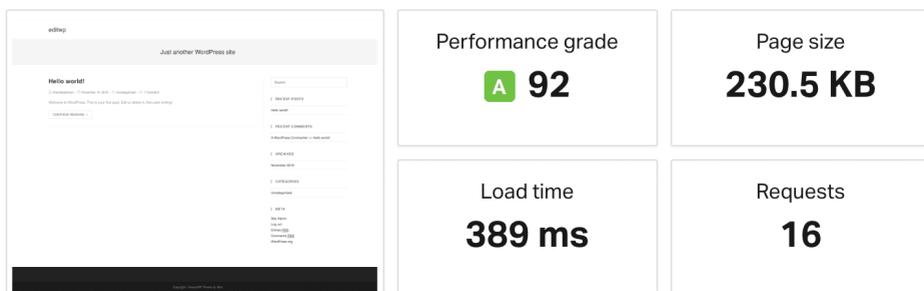
With **899,712** downloads, OceanWP is the fastest growing theme for WordPress. It's the favorite choice of thousands of developers and hobby-users.

 FREE DOWNLOAD

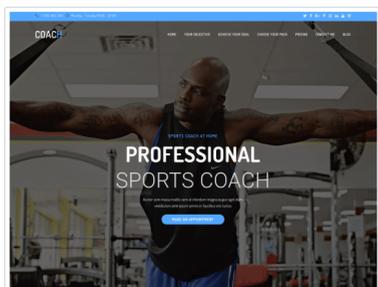
DEVELOPER

HOBBYIST

Mais c'est rapide à quel point ? Nous avons fait une nouvelle installation d'OceanWP, effectué cinq tests de vitesse dans Pingdom, et pris la moyenne. Le temps de chargement total était de **389 ms** avec une **taille de page totale de seulement 230,8 Ko**. Les scripts d'OceanWP sont un peu plus gros, mais il n'y a rien d'alarmant.



Nous avons ensuite effectué une autre série de tests avec l'un des thèmes de démonstration de la bibliothèque de sites OceanWP. Il contient des images, des arrière-plans, de nouvelles sections et nécessite le plugin de création de page Elementor. Vous pouvez voir qu'il a encore été chronométré à moins de **600 ms**.



Performance grade

**B 84**

Page size

**941.1 KB**

Load time

**548 ms**

Requests

**48**

## Astra

Astra est un thème rapide, entièrement personnalisable et magnifique qui convient aux blogs, aux portfolios, aux sites Web d'entreprise et aux boutiques WooCommerce. Il est très léger (moins de 50 Ko en front-end) et offre une vitesse inégalée. Construit par l'équipe de Brainstorm Force, il est activement mis à jour et bien soutenu. Vous les reconnaîtrez peut-être comme les créateurs du populaire plugin All In One Schema Rich Snippets (disponible sur WordPress.org) qui existe depuis de nombreuses années.

**ASTRA**

STARTER SITES PRO FEATURES WOOCOMMERCE PRICING [DOWNLOAD](#)

# Perfect Theme For Any Website

Stylish, Lightning Fast & Easily Customizable

[DOWNLOAD NOW](#)

Stunning Business Website

Broad Video Home Service Great Video

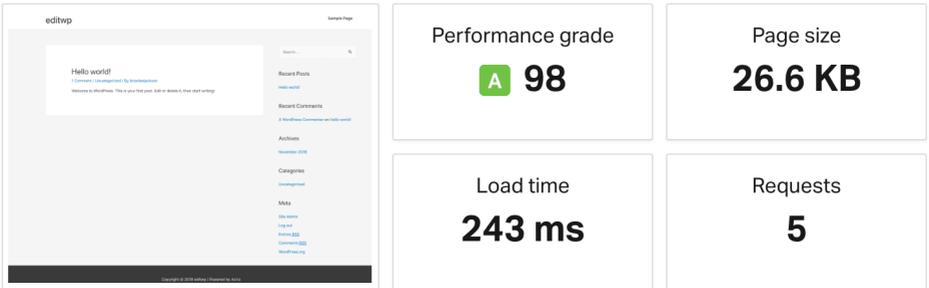
100 Grand Hotels, New York  
*Elizabeth's Makeup Studio*

LET'S BEAUTIFY YOUR GARDEN TOGETHER!

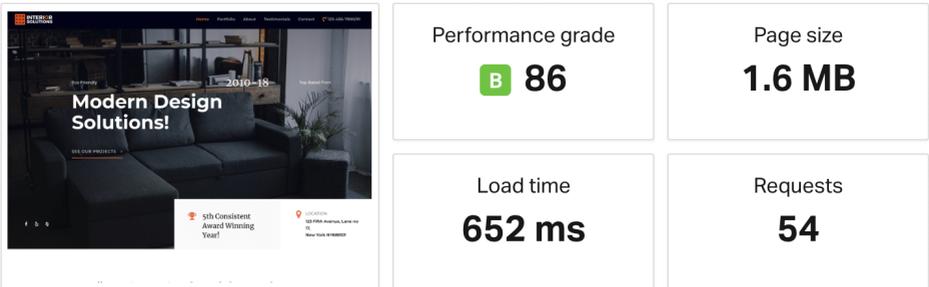
WE START HAPPY. YOU SMILE.

Tout comme pour GeneratePress et OceanWP, il existe une version gratuite et une version premium. Si vous jetez un coup d'œil au dépôt WordPress, vous constaterez que la version gratuite compte actuellement plus de 400,000 installations actives, plus de 1.6 millions de téléchargements et 5 étoiles sur 5 (plus de 2,500 personnes lui ont attribué 5 étoiles).

C'est rapide à quel point ? Nous avons fait une nouvelle installation d'Astra, fait cinq tests de vitesse dans Pingdom, et pris la moyenne. Le temps de chargement total était de **243 ms** avec une **taille de page totale de seulement 26,6 Ko**.



Nous avons ensuite effectué une autre série de tests avec l'un des thèmes de démonstration de la bibliothèque de sites du kit de démarrage Astra. Il contient des images, des arrière-plans, de nouvelles sections et nécessite le plugin de création de page Elementor. Vous pouvez voir qu'il a encore été chronométré à moins de **700 ms**. Note : les images de cette démo ont été entièrement compressées, mais elles ont été choisies en très haute résolution dès le début.



Il est important de prendre avec des pincettes les différences entre les tests de vitesse avec ces trois thèmes. Le problème, c'est qu'il est presque impossible d'effectuer une

comparaison côte à côte complètement exacte. La chose importante que nous voulions vous montrer est que tous ces thèmes WordPress sont ultra-rapides, à la fois prêts à l'emploi et ils proposent des démos complètes !

## Avertissement concernant les constructeurs de pages

Comme vous l'avez probablement remarqué, OceanWP et Astra ont tous deux demandé aux constructeurs de pages d'utiliser les thèmes de leur bibliothèque de sites. Voici quelques points à garder à l'esprit lors de l'utilisation d'un plugin de constructeur de page :

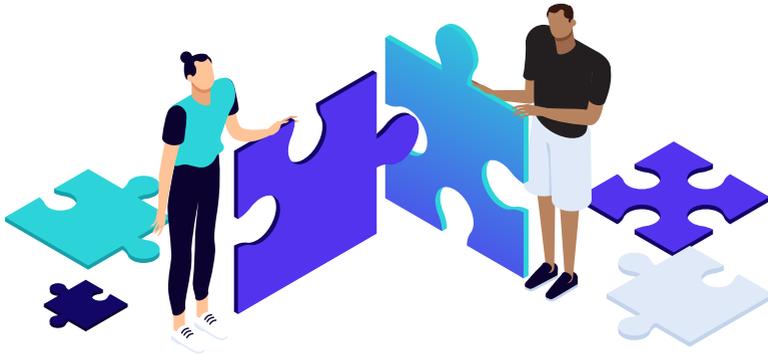
- Certains constructeurs de pages peuvent augmenter le temps de chargement sur votre site. C'est parce qu'ils doivent charger des CSS et JS supplémentaires pour que les choses fonctionnent pour vous sans code. C'est ainsi que la magie opère ! Nous recommandons toujours de tester rapidement votre site WordPress avant et après l'installation d'un constructeur de pages.
- Vous êtes en train de vous engager et de vous enfermer dans ce constructeur de page pour la conception. Assurez-vous d'en choisir un qui est régulièrement mis à jour et qui a tout ce dont vous avez besoin pour le long terme.

Cela étant dit, nous sommes toujours de grands fans des constructeurs de pages comme Elementor et Beaver Builder. Dans la plupart des cas, ils sont conçus pour la performance et n'ajoutent que peu de ressources additionnelles. Pour la plupart, la fonctionnalité et la convivialité en valent la peine, car ces plugins vous permettent de créer tout ce que vous pouvez imaginer ! Ils peuvent aussi être plus rapides dans certains cas car ils peuvent remplacer 5+ autres plugins que vous auriez dû utiliser autrement.

Cependant, si vous n'avez pas besoin d'un plugin de construction de page, ne vous contentez pas d'en installer un pour vous amuser. Il sera également intéressant de voir comment le nouvel éditeur de Gutenberg jouera un rôle dans la conception de site au cours des deux prochaines années.

CHAPITRE 06:

# Le point sur les plugins WordPress



**Passons maintenant au scoop sur les plugins WordPress. On vous a peut-être dit de ne pas installer trop de plugins, sinon cela ralentit votre site WordPress. Bien que cela soit parfois vrai, ce n'est pas le facteur le plus important. Le nombre de plugins n'est pas aussi important que leur qualité. Voilà, on l'a dit.**

Tout comme pour les thèmes, il est important de savoir comment le plugin est développé et s'il a été construit dans un souci de performance. Nous avons de nombreux clients chez Kinsta qui exécutent de 30 à 40 plugins et leurs sites se chargent encore en moins d'une seconde.

Bien qu'il soit amusant d'ajouter du code à votre site, ce n'est pas toujours pratique pour les raisons suivantes :

- 1.** Vous devez maintenir le code vous-même et le mettre à jour au fur et à mesure que les normes changent. Les gens sont occupés, pourquoi ne pas compter sur les développeurs fantastiques qui connaissent les standards mieux que la plupart ?
- 2.** La plupart du temps, un plugin bien codé ne va pas introduire beaucoup plus de temps de chargement que le code lui-même.
- 3.** Vous devez vous rappeler qu'une majorité de la communauté WordPress n'est pas aussi au fait de la technologie que les développeurs. Les plugins sont des solutions qui aident à résoudre des problèmes.

Croyez-nous, nous avons vu le pire du pire chez Kinsta. La plupart, mais pas tous, des plugins que nous bannissons chez Kinsta causent des problèmes de performances.

Cependant, on ne peut pas ignorer que l'une des choses que les gens aiment à propos de WordPress est sa bibliothèque massive de plugins tiers. Mais avec plus de 56 000 plugins gratuits listés sur WordPress.org et des milliers d'autres listés ailleurs, il peut être difficile de trouver le plugin qu'il vous faut. Vous parlez d'une aiguille dans une botte de foin !

Nous essayons seulement de partager les choses que nous utilisons quotidiennement. Et oui, nous utilisons des plugins WordPress sur notre site comme vous tous. Plusieurs membres de l'équipe de Kinsta développent et vendent même des plugins.

## Un gros problème avec les plugins WordPress

Un gros problème avec les plugins WordPress est le processus de désinstallation. Chaque fois que vous installez un plugin ou un thème WordPress, il stocke les données dans la base de données. Le problème est que lorsque vous supprimez un plugin en utilisant l'une des méthodes standard, il laisse généralement derrière lui des tables et des lignes dans votre base de données. Avec le temps, cela peut s'ajouter et représenter beaucoup de données et même commencer à ralentir votre site. Dans notre exemple, nous avons désinstallé le plugin de sécurité Wordfence, et il a laissé 24 tables dans notre base de données (voir ci-dessous). C'est encore pire si elles sont derrière les données de votre table `wp_options`.

<input type="checkbox"/>	<a href="#">wp_wfBadLeechers</a>	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	0	InnoDB	latin1_swedish_ci	16	KiB
<input type="checkbox"/>	<a href="#">wp_wfBlockedIPLog</a>	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	35	InnoDB	utf8_general_ci	16	KiB
<input type="checkbox"/>	<a href="#">wp_wfBlocks</a>	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	32	InnoDB	utf8_general_ci	32	KiB
<input type="checkbox"/>	<a href="#">wp_wfBlocksAdv</a>	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	0	InnoDB	utf8_general_ci	16	KiB
<input type="checkbox"/>	<a href="#">wp_wfConfig</a>	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	156	InnoDB	utf8_general_ci	16	KiB
<input type="checkbox"/>	<a href="#">wp_wfCrawlers</a>	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	10	InnoDB	latin1_swedish_ci	16	KiB
<input type="checkbox"/>	<a href="#">wp_wfFileMods</a>	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	0	InnoDB	utf8_general_ci	16	KiB
<input type="checkbox"/>	<a href="#">wp_wfHits</a>	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	1,907	InnoDB	latin1_swedish_ci	1.3	MiB
<input type="checkbox"/>	<a href="#">wp_wfHoover</a>	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	0	InnoDB	utf8_general_ci	32	KiB
<input type="checkbox"/>	<a href="#">wp_wfIssues</a>	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	0	InnoDB	utf8_general_ci	16	KiB
<input type="checkbox"/>	<a href="#">wp_wfKnownFileList</a>	★	<a href="#">Browse</a>	<a href="#">Structure</a>	<a href="#">Search</a>	<a href="#">Insert</a>	<a href="#">Empty</a>	<a href="#">Drop</a>	0	InnoDB	utf8_general_ci	16	KiB

<input type="checkbox"/>	wp_wfLeechers	★	Browse	Structure	Search	Insert	Empty	Drop	156,922	InnoDB	latin1_swedish_ci	13.5 MiB
<input type="checkbox"/>	wp_wfLockedOut	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	16 KiB
<input type="checkbox"/>	wp_wfLocs	★	Browse	Structure	Search	Insert	Empty	Drop	4	InnoDB	utf8_general_ci	16 KiB
<input type="checkbox"/>	wp_wfLogins	★	Browse	Structure	Search	Insert	Empty	Drop	5	InnoDB	utf8_general_ci	48 KiB
<input type="checkbox"/>	wp_wfNet404s	★	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8_general_ci	32 KiB
<input type="checkbox"/>	wp_wfNotifications	★	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8_general_ci	16 KiB
<input type="checkbox"/>	wp_wfPendingIssues	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	16 KiB
<input type="checkbox"/>	wp_wfReverseCache	★	Browse	Structure	Search	Insert	Empty	Drop	7	InnoDB	latin1_swedish_ci	16 KiB
<input type="checkbox"/>	wp_wfScanners	★	Browse	Structure	Search	Insert	Empty	Drop	618	InnoDB	latin1_swedish_ci	64 KiB
<input type="checkbox"/>	wp_wfSNIPCache	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	64 KiB
<input type="checkbox"/>	wp_wfStatus	★	Browse	Structure	Search	Insert	Empty	Drop	40	InnoDB	utf8_general_ci	48 KiB
<input type="checkbox"/>	wp_wfThrottleLog	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	32 KiB
<input type="checkbox"/>	wp_wfVulnScanners	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	latin1_swedish_ci	16 KiB

Tables WordFence

Et en plus de la base de données, de nombreux plugins laissent aussi derrière eux des dossiers et des fichiers supplémentaires. D'après notre expérience, cela se voit couramment avec les plugins de sécurité et de mise en cache qui créent des répertoires supplémentaires pour les logs. Par exemple, après la suppression du plugin Wordfence, il nous restait un dossier « wflogs » dans notre répertoire wp-content. Et nous n'essayons pas de nous en prendre à Wordfence, la majorité des plugins et des thèmes sur le marché fonctionnent de cette façon.

Remote site: /www/ /public/wp-content/wflogs			
	?	upgrade	
	?	uploads	
	📁	wflogs	
Filename	File...	Filetype	Last modified
📁 ..			
📄 .htaccess	133	HTACCESS File	7/24/2017 10:54:49 AM
📄 attack-data.php	40,0...	PHP File	7/24/2017 10:54:49 AM
📄 config.php	594,...	PHP File	7/24/2017 10:57:30 AM
📄 ips.php	51	PHP File	7/24/2017 10:54:50 AM
📄 rules.php	101,...	PHP File	7/24/2017 10:54:59 AM
📄 wafRules.rules	44,2...	RULES File	7/24/2017 10:54:59 AM

Logs WordFence

## Pourquoi les développeurs font-ils cela ?

Donc vous vous demandez probablement pourquoi les développeurs n'ont pas d'options d'auto-nettoyage lorsque vous désinstallez et supprimez un plugin ? Eh bien, c'est le cas. Mais, voici quelques raisons pour lesquelles elles ne sont probablement pas aussi évidentes d'emblée.

- 1. Ils veulent conserver les paramètres pour l'utilisateur.** Si vous supprimez un plugin WordPress et décidez de le réessayer plus tard, tous vos paramètres et données seront toujours là. Bien que cela soit très pratique, ce n'est pas le moyen le plus efficace.
- 2. Ils se fichent de la performance.** Certains développeurs pourraient soutenir que le fait de ne pas tenir compte des tables n'a pas d'impact sur la performance. Mais imaginez un site sur dix ans, ayant utilisé des centaines de plugins, qui ont généré peut-être des milliers de lignes ou de tables. Les requêtes de base de données ont un impact significatif sur les performances de votre site WordPress, et les plugins peuvent faire beaucoup de ces requêtes si le développeur n'a pas fait attention. En général, un plugin bien écrit ne devrait interroger que les tables ou les lignes auxquelles il est lié, mais ce n'est pas toujours le cas. Nous l'avons vu chez Kinsta, de longues requêtes de base de données ont amené un site à ramer à cause des données chargées automatiquement dans la table `wp_options` qui ont été laissées derrière.
- 3. Ils ont fait une erreur.** Le manuel du plugin WordPress dit même que "les développeurs moins expérimentés font parfois l'erreur d'utiliser le hook de désactivation pour cela".

La bonne nouvelle ? Il existe des moyens de nettoyer et de se débarrasser correctement d'un plugin. Des liens vers les tutoriels approfondis suivants sont disponibles sur notre site Web sur <https://kinsta.com/fr/apprendre/accellerer-wordpress/>.

- Comment désinstaller un plugin WordPress (la bonne façon)
- Comment nettoyer manuellement les tables laissées derrière

CHAPITRE 07:

# Les paramètres WordPress optimaux



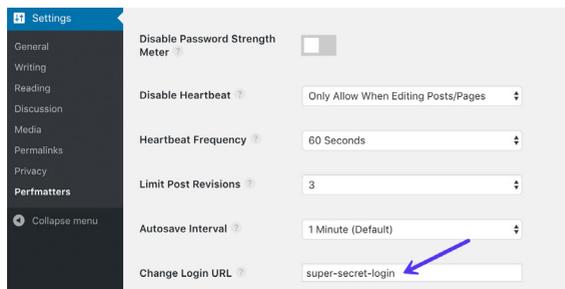
**Passons maintenant aux paramètres optimaux de WordPress. Voici quelques changements que vous pouvez apporter pour accélérer votre site WordPress. Beaucoup de ces changements sont très subtils, mais tout aide !**

## **Modifier l'URL de connexion WordPress**

Par défaut, l'URL de connexion de votre site WordPress est `domaine.com/wp-admin/`. Un des problèmes avec ceci est que tous les robots, les pirates et les scripts le savent aussi. En changeant l'URL, vous pouvez moins devenir une cible, mieux vous protéger contre les attaques par force brute, et diminuer la bande passante utilisée par les robots qui frappent cette URL à plusieurs reprises.

Changer votre URL de connexion WordPress peut aussi aider à prévenir les erreurs courantes comme "429 Too Many Requests". Ce n'est pas une solution toute faite, c'est simplement une petite astuce qui peut aider à vous protéger et à diminuer la charge sur cette page.

Pour changer votre URL de connexion WordPress, nous vous recommandons d'utiliser l'un des plugins suivants :



Changer l'URL de connexion WordPress dans Perfmatters

- **WPS Hide Login** (Gratuit, disponible sur [WordPress.org](https://WordPress.org))
- **Perfmatters** (Premium, mais inclut d'autres paramètres d'optimisation des performances. Développé par un membre de l'équipe de Kinsta et disponible sur [perfmatters.io](https://perfmatters.io))

## Désactiver ou modifier les mises à jour des plugins et des thèmes

Les tableaux de bord d'administration lents de WordPress peuvent être affectés par le réseau, l'emplacement du centre de données et même les versions PHP. Mais un autre facteur dont peu de gens parlent est le vérificateur de mise à jour WordPress qui fonctionne en arrière-plan. C'est un exemple où le fait d'avoir beaucoup de plugins et de thèmes WordPress pourrait vous nuire. WeFoster a un excellent article de blog à ce sujet où ils inventent la phrase "Third Party Plugin Update Check Syndrome" ou TPPUCS.

Essentiellement, le problème est que le vérificateur de mise à jour WordPress intégré fait une requête GET externe en arrière plan (<https://third-party-plugin/update-check.php>). Parfois, cela peut être périodique ou très fréquent. Si cela se produit tout le temps, cela pourrait rendre votre tableau de bord d'administration lent.

C'est plus un problème avec la façon dont le vérificateur de mise à jour dans WordPress est construit. Si vous souffrez de la lenteur des temps de chargement du tableau de bord d'administration de WordPress, vous devrez peut-être essayer ceci. Le remède est de désactiver les mises à jour automatiques. Attention : Ne le faites que si vous avez l'intention de vérifier les mises à jour manuellement. De nombreuses mises à jour incluent des corrections de sécurité et de bugs.

Pour désactiver les mises à jour, nous vous recommandons d'utiliser l'un des plugins suivants (les deux sont disponibles sur WordPress.org):

- **Disable All WordPress Updates** : Entièrement gratuit, sans aucun réglage. Il fait bien ce qu'il dit.
- **Easy Updates Manager** : Offre plus de contrôle sur les mises à jour sélectives. La version de base est gratuite.

Vous pouvez facilement définir vous-même un rappel via un calendrier, désactiver le plugin une fois par semaine, vérifier les mises à jour, puis le réactiver.

## Désactiver les Pingbacks (rétroliens)

Un pingback est un commentaire automatisé qui est créé lorsqu'un autre blog vous relie. Il peut également y avoir des self-pingbacks qui sont créés lorsque vous créez un lien vers un article dans votre propre blog.

Nous vous recommandons de les désactiver simplement car ils génèrent des requêtes inutiles et des spams supplémentaires sur votre site. Rappelez-vous que moins votre site WordPress reçoit d'appels, mieux c'est, surtout sur les sites très fréquentés. Sans parler du fait qu'un pingback sur votre propre site web est tout simplement ennuyeux. Suivez les étapes ci-dessous pour désactiver les pingbacks

### Étape 1 – Désactiver les Pingbacks d'autres blogs

Dans votre tableau de bord WordPress, cliquez sur "Paramètres / Discussion". Dans la section Paramètres de discussion, décochez l'option "Autoriser les notifications de liens d'autres blogs (pingbacks et trackbacks) sur les nouveaux articles".

#### Discussion Settings

##### Default article settings

Attempt to notify any blogs linked to from the article

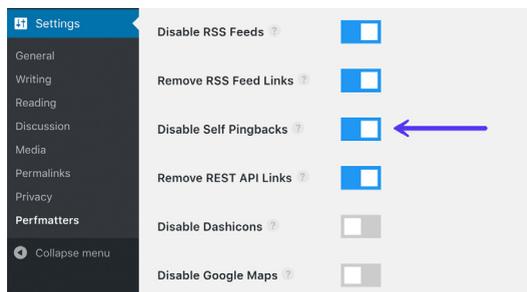


Allow link notifications from other blogs (pingbacks and trackbacks) on new articles

Allow people to post comments on new articles

*(These settings may be overridden for individual articles.)*

## Étape 2 – Désactiver le self-pingback



Lorsqu'il s'agit de désactiver les self-pingbacks, vous avez plusieurs options. Vous pouvez utiliser le plugin gratuit No Self Pings. Ou vous pouvez utiliser un plugin premium comme Perfmatters (disponible sur [perfmatters.io](https://perfmatters.io)).

Désactiver les self-pingbacks avec Perfmatters

Alternativement, vous pouvez aussi désactiver le self-pingback en ajoutant le code suivant au fichier `functions.php` de votre thème WordPress. Attention, éditer le code source d'un thème WordPress peut casser votre site s'il n'est pas fait correctement. Astuce, vous pouvez facilement ajouter des bouts de PHP comme ceci avec le plugin gratuit Code Snippets (disponible sur [WordPress.org](https://WordPress.org)). Cela signifie que vous n'avez jamais à toucher à votre thème.

```
function wpsites_disable_self_pingbacks( &$links ) {  
    foreach ( $links as $l => $link )  
        if ( 0 === strpos( $link, get_option( 'home' ) ) )  
            unset($links[$l]);  
}  
  
add_action( 'pre_ping', 'wpsites_disable_self_pingbacks' );
```

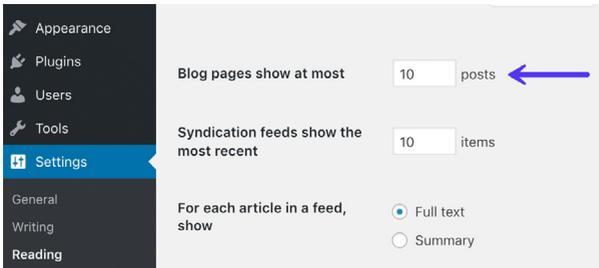
## Limiter les articles sur votre flux de Blog

Que votre flux de blog soit défini comme votre page d'accueil ou comme une autre page de votre site, vous n'avez pas besoin de 50 vignettes toutes chargées en même temps. Pour ceux qui ont des blogs très fréquentés, votre page d'accueil est la page la plus importante de votre site, et vous voulez que cela se charge rapidement. Moins il y a de requêtes et de médias, mieux c'est en termes de performances.

C'est précisément pour cette raison que la pagination a été inventée (voir ci-dessous). La pagination est ce que vous voyez à la fin des flux de blogs qui vous permettent de passer à la page suivante. Généralement ce sont des nombres, ou vous pouvez utiliser les textes « suivant/précédent ». Votre thème WordPress aura très probablement déjà une pagination personnalisée intégrée.



WordPress fixe par défaut la limite des nouvelles installations WordPress à 10, mais nous avons vu cela changé tellement de fois que nous avons perdu le compte. Assurez-vous donc de bien vérifier la valeur que vous utilisez. Nous recommandons entre 8 et 12. Si vous êtes curieux, nous en utilisons 12 sur notre page d'accueil du blog Kinsta.



Vous trouverez cette option dans votre tableau de bord d'administration WordPress sous "Paramètres / Lecture". Vous pouvez alors changer la valeur pour "Les pages du site doivent afficher au plus".

CHAPITRE 08:

# Pourquoi le cache est-il si important



**La mise en cache est de loin l'un des moyens les plus importants et les plus faciles d'accélérer WordPress ! Mais avant de vous montrer comment utiliser la mise en cache, il est essentiel de comprendre son fonctionnement et les différents types de mise en cache disponibles.**

## Qu'est-ce que la mise en cache ?

En bref, chaque page Web visitée sur votre site WordPress nécessite une requête au serveur, le traitement par ce serveur (y compris les requêtes de base de données), puis un résultat final envoyé du serveur au navigateur de l'utilisateur. Le résultat est votre site Web, complet avec tous les fichiers et les éléments qui le font paraître tel qu'il est.

Par exemple, vous pouvez avoir un en-tête, des images, un menu et un blog. Comme le serveur doit traiter toutes ces requêtes, il faut un certain temps pour que la page Web complète soit livrée à l'utilisateur, surtout s'il s'agit d'un site Web lourd ou volumineux.

C'est là qu'un plugin de mise en cache WordPress entre en jeu ! La mise en cache demande au serveur de stocker certains fichiers sur le disque ou en RAM, selon

la configuration. Par conséquent, il peut se rappeler et dupliquer le même contenu qu'il servait dans le passé. Fondamentalement, cela réduit la quantité de travail nécessaire pour générer une vue de page. En conséquence, **vos pages Web se chargent beaucoup plus rapidement, directement à partir du cache.**

D'autres avantages de la mise en cache incluent :

- **Votre serveur utilise moins de ressources** – Cela permet d'augmenter la vitesse, car moins il y a de ressources, plus le site est plus rapide. Cependant, cela met aussi moins de pression sur votre serveur. C'est très important lorsqu'il s'agit de sites très dynamiques, comme les sites d'adhésion, et pour déterminer ce que vous pouvez et ne pouvez pas servir à partir du cache.
- **Vous verrez un TTFB plus faible** – La mise en cache est l'un des moyens les plus faciles de réduire votre TTFB. En fait, dans nos tests, la mise en cache réduit généralement le TTFB jusqu'à 90% !

## Types de mise en cache

Quand il s'agit de types de mise en cache, il existe deux approches différentes couramment utilisées :

1. La mise en cache au niveau du serveur
2. La mise en cache avec un plugin

### 1. Mise en cache au niveau du serveur

La mise en cache au niveau du serveur est de loin l'une des approches les plus simples pour l'utilisateur final. Cela signifie que le fournisseur d'hébergement WordPress s'en charge pour vous. Chez Kinsta, nous utilisons les **quatre types de cache** suivants, qui se font tous automatiquement au niveau logiciel ou serveur :

- Cache de bytecode

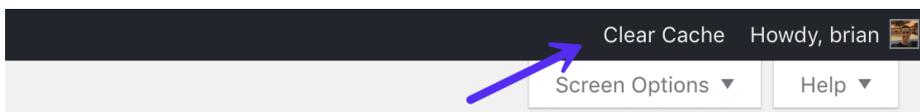
- Cache d'objets
- Cache des pages
- Cache CDN

Cela signifie que vous n'avez pas à vous soucier de travailler avec des plugins de mise en cache compliqués et déroutants. Vous pouvez arrêter de chercher sur Google les “meilleurs plugins de cache” et vous concentrer sur des tâches plus productives.

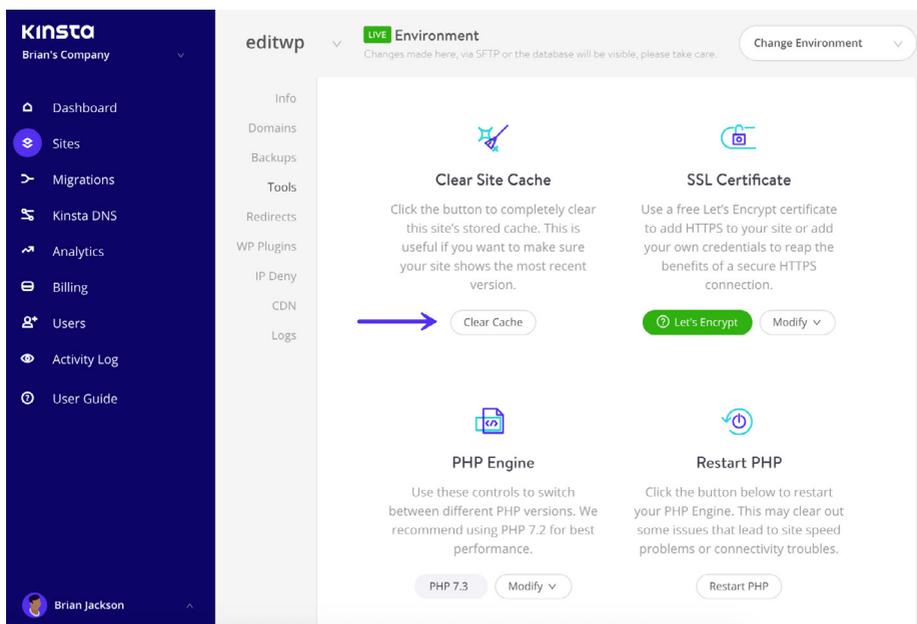
Le cache de page est configuré pour fonctionner immédiatement avec un WordPress standard. Vous n'avez rien à faire ! Lancez simplement votre site WordPress et la mise en cache des pages commencera.

Nous avons également mis en place des règles de mise en cache pour les sites d'e-commerce tels que WooCommerce et Easy Digital Downloads. Par défaut, certaines pages qui ne devraient jamais être mises en cache, comme le panier, mon compte et checkout, sont exclues de la mise en cache. Les utilisateurs contournent automatiquement le cache lorsque le cookie `woocommerce_items_in_cart` ou `edd_items_in_cart` sont détectés pour assurer un processus de paiement fluide et synchronisé.

Vous pouvez facilement effacer le cache de votre site WordPress à tout moment à partir de la barre d'outils de l'administration.



C'est également intégré dans notre tableau de bord MyKinsta. Il suffit de cliquer sur Outils et de cliquer sur “Effacer le cache”.



## 2. Mise en cache avec un plugin

Si votre hébergeur ne fournit pas de cache, vous pouvez utiliser un plugin de cache WordPress tiers. D'après notre expérience, nous recommandons l'un des suivants :

- **WP Rocket** (premium, disponible sur [wp-rocket.me](https://wp-rocket.me))
- **Cache Enabler** (gratuit, disponible sur [WordPress.org](https://WordPress.org))
- **W3 Total Cache** (gratuit, disponible sur [WordPress.org](https://WordPress.org))

Nous supportons également pleinement WP Rocket chez Kinsta ! Nous n'autorisons généralement pas les plugins de cache dans notre environnement car ils entrent en conflit avec notre solution de cache intégrée. Cependant, à partir de WP Rocket 3.0, la fonctionnalité de mise en cache des pages sera automatiquement désactivée lors de leur exécution sur les serveurs Kinsta.

Cela permet aux clients de Kinsta d'utiliser notre cache rapide au niveau du serveur tout en profitant des fantastiques fonctionnalités d'optimisation que WP Rocket a à offrir.

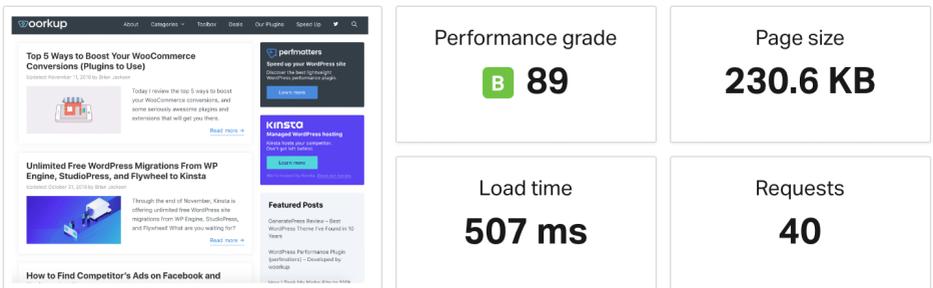
## Pas de cache vs avec cache

Dans quelle mesure la mise en cache aide-t-elle ?

Nous avons effectué quelques tests de vitesse avec le cache au niveau du serveur de Kinsta pour que vous puissiez voir la différence que cela fait, tant en termes de vitesse globale que de TTFB.

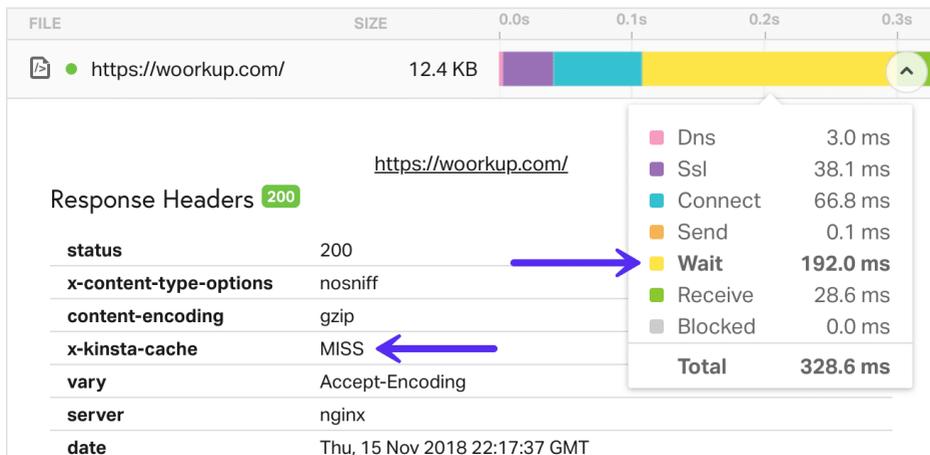
## Pas de cache

Nous avons d'abord effectué cinq tests sur Pingdom sans activer le cache et avons pris la moyenne.



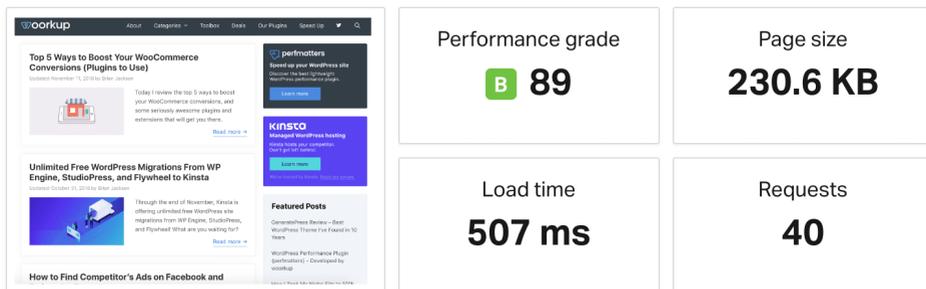
## TTFB sans le cache

Il est également important de noter la différence entre le TTFB sans et avec le cache. Le TTFB de Pingdom est représenté par la barre jaune "en attente". Comme vous pouvez le voir, le TTFB sans cache est de 192 ms. Vous pouvez voir qu'il n'est pas servi à partir du cache car l'en-tête `x-kinsta-cache` affiche un `MISS`.



## Avec le cache activé

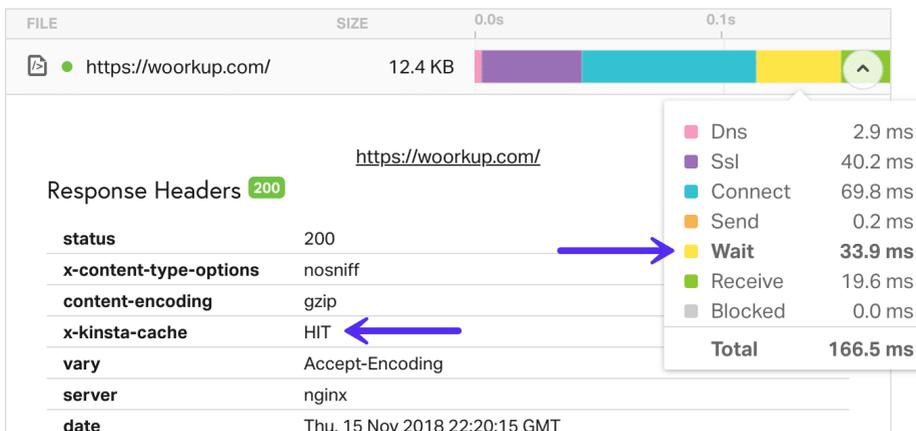
Nous avons ensuite activé la mise en cache au niveau du serveur et effectué cinq tests sur Pingdom et pris la moyenne.



Comme vous pouvez le constater, **le cache au niveau du serveur a réduit le temps de chargement de nos pages de 33,77% !** Et c'est sans travail supplémentaire. Ce site que nous avons testé est également assez optimisé, ainsi les sites non optimisés de plus grande taille sont susceptibles de présenter des différences encore plus grandes.

## TTFB avec le cache activé

Maintenant, si nous regardons le TTFB avec le cache activé, nous pouvons voir qu'il est inférieure à 35 ms. Vous pouvez voir qu'il est servi à partir du cache car l'en-tête `x-kinsta-cache` affiche un `HIT`.



Le cache CDN est également aussi important que le cache de votre hébergeur WordPress. Nous nous pencherons plus en détails sur les CDNs plus loin dans ce qui suit.

*La mise en cache de WordPress peut facilement réduire le temps de chargement de votre page de plus de 33% !*

## Problèmes avec le cache et les sites d'adhésion

Les sites d'adhésion contiennent beaucoup de **contenu non cachable** et de pages qui changent continuellement. Des éléments tels que la page de connexion pour les membres de la communauté (qui pourrait être consultée constamment en fonction de la taille du site), les pages de paiement pour les biens numériques ou les cours, et les forums de discussion sont des coupables communs et des points faibles, car ceux-ci

ne peuvent généralement pas être mis en cache.

Mais cela ne s'arrête pas là. Sur les sites WordPress standard, le tableau de bord WordPress n'est pas non plus mis en cache pour les **utilisateurs "connectés"**. C'est très bien quand vous n'avez que quelques auteurs et administrateurs, mais quand vous avez soudainement des milliers de membres qui utilisent le tableau de bord, cela provoque immédiatement des problèmes de performances car rien de tout cela ne peut être servi depuis le cache sur le serveur. Cela signifie que vous avez besoin de la puissance et de l'architecture en coulisses pour gérer cela. Les fournisseurs d'hébergement mutualisé sont généralement paralysés dans ces circonstances.

## Mise en cache d'objets pour les sites hautement dynamiques

Quand il s'agit de sites d'adhésion WordPress, vos configurations de mise en cache communes ne sont généralement pas suffisantes car elles ne tirent pas toujours pleinement parti de ce service. C'est ici que **la mise en cache d'objets entre en jeu**.

Le cache d'objet stocke les résultats des requêtes de la base de données de sorte que la prochaine fois que ce bit particulier de données est nécessaire, il peut être livré à partir du cache sans interroger la base de données. Cela accélère les temps d'exécution de PHP et réduit la charge sur votre base de données. Cela devient extrêmement important avec les sites d'adhésion ! Avec WordPress, vous pouvez implémenter la mise en cache d'objets de plusieurs façons différentes :

1. Une solution de mise en cache tierce telle que W3 Total Cache
2. **Redis (recommandé)**
3. Memcached

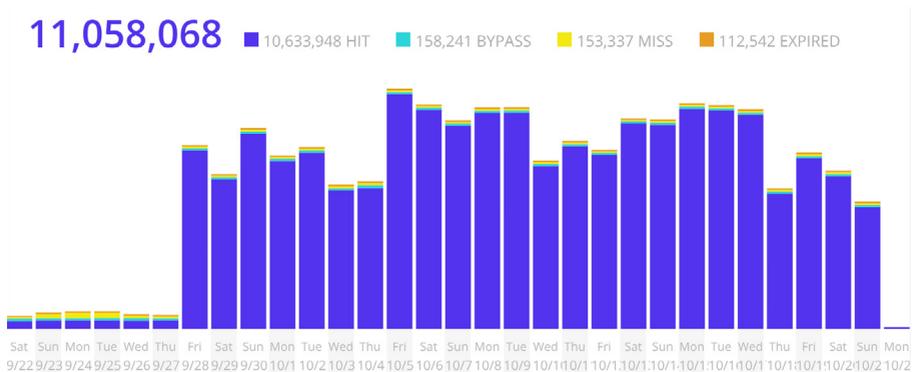
Nous offrons Redis comme option supplémentaire chez Kinsta pour que vous puissiez profiter pleinement de la mise en cache d'objets persistants pour vos sites d'adhésion.

## Analyse du cache

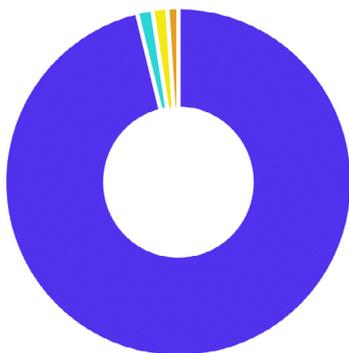
Vous vous souvenez de l'en-tête `x-kinsta-cache` mentionnée plus haut ? Selon votre hébergeur ou votre solution de mise en cache, l'en-tête peut porter un nom légèrement différent. Chaque fois qu'une requête est faite à partir de votre site WordPress, cet en-tête a une valeur, telle que HIT, BYPASS, MISS, et EXPIRED. Cela vous permet de voir comment fonctionne votre cache.

Augmenter le taux de consultation du cache de votre site WordPress est important parce que vous voulez qu'une grande partie de votre site soit servie à partir du cache autant que possible. Chez Kinsta, vous pouvez analyser les données via notre outil d'analyse MyKinsta et les logs de cache kinsta pour déterminer s'il existe des requêtes GET qui contournent le cache qui pourraient être mises en cache ou des requêtes POST qui pourraient être éliminées.

**La pile de composants du cache** (comme indiqué ci-dessous) vous permet de voir l'état de chaque requête, qu'elle soit HIT, BYPASS, MISS, ou EXPIRED. Vous pouvez filtrer les données par 24 heures, 7 jours ou 30 jours.



**Le diagramme des composants du cache** vous donne un aperçu de votre taux de mise en cache. Plus vous servez de requêtes à partir du cache, mieux c'est. Comme vous pouvez le voir dans l'exemple ci-dessous, ce site WordPress possède un ratio de 96,2% de HIT pour le cache. Ce qui est bien !



Total  
11,058,068

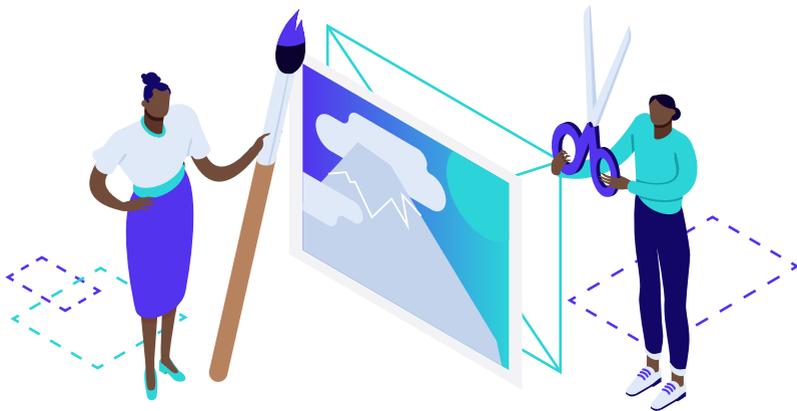
- HIT 10,633,948 (96.2%)
- BYPASS 158,241 (1.4%)
- MISS 153,337 (1.4%)
- EXPIRED 112,542 (1.0%)

La section des plus nombreuses requêtes qui **contournent le cache** vous permet de voir quelles requêtes ne sont pas servies à partir du cache (Bypass). Généralement, il peut s'agir de tâches CRON, de requêtes admin-ajax, de pages de commande ecommerce, de query strings, de paramètres UTM, etc.

PATH	REQUESTS
/wp-cron.php?server_triggered_cronjob	28,977
/wp-admin/admin-ajax.php?action=edd_jilt_get_cart_data	22,279
?ref=1	796
/	667
/sitemap_index.xml	490
?ajax=1&ht-kb-search=1&s=	435
/account/	372
/checkout/	329
?ref=19&campaign=bw	292

CHAPITRE 09:

# L'optimisation des images est indispensable

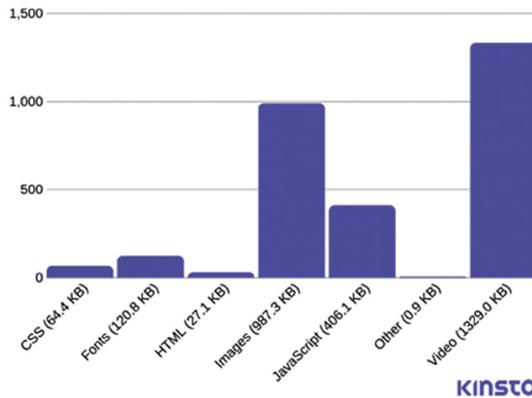


**L'optimisation d'images est une autre chose simple que vous pouvez faire et qui a un impact significatif sur le temps de chargement global de vos pages. Ce n'est pas facultatif ; tous les sites devraient le faire !**

Les grosses images ralentissent vos pages Web, ce qui crée une expérience utilisateur moins optimale. L'optimisation des images consiste à réduire la taille de leur fichier, à l'aide d'un plugin ou d'un script, ce qui accélère le temps de chargement de la page. La compression lossy et la compression lossless sont deux méthodes couramment utilisées.

Selon HTTP Archive, en août 2019, les images représentaient en moyenne **34% du poids total d'une page Web**. Ainsi, après les vidéos, qui sont beaucoup plus difficiles à optimiser, les images sont de loin le premier endroit où vous devriez commencer ! C'est plus important que le JavaScript, le CSS et les Fonts (polices d'écriture). Et ironiquement, un bon flux de travail d'optimisation d'image est l'une des choses les plus faciles à mettre en œuvre, mais beaucoup de propriétaires de sites Web négligent cela.

## Bytes moyens par page (KB)



Les images représentaient en moyenne 54% du poids total d'une page en décembre 2017. Il semble donc que le web dans son ensemble s'améliore dans l'optimisation des images ! Mais 34 %, c'est toujours un chiffre que l'on ne peut ignorer. Si vous n'avez pas de contenu vidéo sur votre site Web, les images restent probablement votre point faible n°1 pour le poids des pages.

*Les images représentent en moyenne 34% du poids total d'une page web. Optimisez-les !*

## Trouver l'équilibre (taille et qualité du fichier)

L'objectif principal de l'optimisation de vos images est de trouver **l'équilibre entre la taille de fichier la plus petite et une qualité acceptable**. Il existe plus d'une façon d'effectuer presque toutes ces optimisations. L'un des moyens les plus simples est de les compresser avant de les télécharger vers WordPress. Habituellement, cela peut être fait dans un outil comme Adobe Photoshop ou Affinity Photo. Ou en utilisant la nouvelle application gratuite Squoosh de Google. Cependant, ces tâches peuvent aussi être effectuées automatiquement à l'aide de plugins, dont nous parlerons plus loin.

Les deux principales choses à considérer sont le **format de fichier** et le **type de compression** que vous utilisez. En choisissant la bonne combinaison de format de fichier et de type de compression, vous pouvez réduire jusqu'à 5 fois la taille de votre image. Vous devrez expérimenter avec chaque format d'image ou de fichier pour voir ce qui fonctionne le mieux.

Avant de commencer à modifier vos images, assurez-vous d'avoir choisi le meilleur type de fichier. Il existe plusieurs types de fichiers que vous pouvez utiliser :

- **PNG** – produit des images de meilleure qualité, mais a également une plus grande taille de fichier. A été créé comme un format d'image "lossless", bien qu'il puisse aussi être "lossy".
- **JPEG** – utilise l'optimisation "lossy" ou "lossless". Vous pouvez ajuster le niveau de qualité pour obtenir un bon équilibre entre la qualité et la taille du fichier.

Idéalement, vous devriez utiliser JPEG (ou JPG) pour les images avec beaucoup de couleurs et PNG pour les images simples.

Qu'en est-il des GIFs ? Les GIFs animés sont toujours amusants, mais ils tuent la performance web. Beaucoup de GIFs ont une taille supérieure à 1 Mo. Nous vous recommandons de les conserver pour les médias sociaux et Slack. S'il y en a un dont vous ne pouvez pas vous passer dans votre blog, jetez un coup d'oeil à la façon dont vous pouvez compresser les GIFs animés.

## Qualité de la compression vs Taille

Voici un exemple de ce qui peut arriver si vous compressez trop une image. La première montre l'utilisation d'un taux de compression très faible, ce qui permet d'obtenir la meilleure qualité (mais une taille de fichier plus grande). La seconde illustre l'utilisation d'un taux de compression très élevé, ce qui donne une image de très mauvaise qualité (mais avec une taille de fichier plus petite). Remarque : L'image originale intacte est de 2,06 Mo.

Comme vous pouvez le voir, la première image ci-dessus est de 590 KB. C'est assez gros pour une photo ! Il est généralement préférable que vous puissiez garder le poids total d'une page Web à moins de 1 ou 2 MB en taille. 590 KB serait déjà un quart de ce chiffre. La deuxième image a l'air horrible, mais alors c'est seulement 68 KB. Ce que vous voulez faire, c'est trouver un juste milieu entre votre taux de compression (qualité) et la taille du fichier.



Basse compression (haute qualité) JPG - 590 KB



Haute compression (faible qualité) JPG - 68 KB



Compression moyenne (grande qualité) JPG - 151 KB

Nous avons donc repris l'image avec un taux de compression moyen, et comme vous pouvez le voir ci-dessous, la qualité est bonne maintenant, et la taille du fichier est de 151 Ko, ce qui est acceptable pour une photo haute résolution. C'est presque 4x petit que la photo originale avec une faible compression. Nous essayons de garder la plupart de nos images sous

la barre des 100 KB pour une meilleure performance.

## Optimisation Lossy vs Lossless

Il est également important de comprendre qu'il existe deux types de compressions que vous pouvez utiliser, lossy ou lossless.

La compression lossy implique l'**élimination de certaines données de votre image.**

Pour cette raison, cela signifie que vous pourriez voir une dégradation (réduction de la qualité ou ce que certains appellent une pixellisation). Il faut donc faire attention de combien vous réduisez votre image. Non seulement en raison de la qualité, mais aussi parce que vous ne pouvez pas inverser le processus. Bien sûr, l'un des grands avantages de la compression lossy et pourquoi c'est l'une des méthodes de compression les plus populaires est que vous pouvez **réduire la taille du fichier de manière considérable**.



La compression lossless, contrairement à la compression lossy, **ne réduit pas la qualité de l'image**. Comment est-ce possible ? Cela se fait généralement en supprimant les métadonnées inutiles (données générées automatiquement par l'appareil qui capture l'image). Cependant, le plus grand inconvénient de cette méthode est que **vous ne verrez pas une réduction significative de la taille du fichier**. En d'autres termes, cela occupera beaucoup d'espace disque avec le temps.

Vous devrez expérimenter avec ce qui fonctionne le mieux pour vous. Mais pour la majorité des utilisateurs, **nous recommandons l'utilisation de la compression lossy** car vous pouvez facilement compresser une image bien au-delà de 70% (parfois même plus de 90% !) sans perte de qualité importante. Multipliez ceci par 15 images sur une page, et cela jouera un rôle important dans la réduction du temps de chargement de votre site.

## Plug-ins de compression d'images

La grande nouvelle, c'est qu'il existe des plug-ins de compression d'images WordPress excellents que vous pouvez utiliser pour automatiser l'ensemble du processus. Voici

quelques plugins que nous recommandons :

- **Imagify** (lossy et lossless – optimise les images depuis l'extérieur)
- **WP Smush** (lossy et lossless – optimise les images depuis l'extérieur)
- **Optimole** (lossy et lossless – optimise les images depuis l'extérieur)
- **EWWW Cloud** (lossy et lossless – optimise les images depuis l'extérieur)
- **ShortPixel** (lossy et lossless – optimise les images depuis l'extérieur)

La chose la plus importante lors du choix d'un plugin d'optimisation d'image est d'**utiliser un plugin qui compresse et optimise les images en externe sur leurs serveurs**. Ceci, à son tour, réduit la charge sur votre site. Tous ceux qui précèdent le font.

Si vous êtes curieux, nous utilisons le plugin Imagify (disponible sur WordPress.org) sur le site Kinsta. Il compresse automatiquement les images lorsque nous les téléchargeons dans la médiathèque WordPress. On n'a jamais à s'inquiéter de quoi que ce soit. Avec le temps, vous pouvez vous faire une idée du niveau de compression d'image que vous souhaitez utiliser. Il offre Normal, Agressif, et Ultra.

**Nous utilisons le mode Agressif** chez Kinsta et nous réalisons généralement des économies de **60 à 70% selon l'image**. Note : nous utilisons beaucoup plus de PNG que de JPEG car la plupart de nos images sont des icônes et des illustrations, pas des photos.

Image Name	Author	Article Title	Date	New File Size	Original Saving
bedrock-trellis.png	Brian Jackson	How to Use Bedrock and Trellis at Kinsta (WordPress Development)	2018/04/10	6.16 KB	74.68%
kinsta-architecture-2.png	Brian Jackson	Does Kinsta Offer Shared, VPS, or Dedicated Server Hosting?	2018/04/09	82.47 KB	62.41%

À quel point votre site WordPress sera-t-il plus rapide si vous utilisez la compression d'images ? Tout dépend de la taille de vos images originales et après compression. Cependant, nous avons fait des tests de vitesse et avons constaté qu'une solution de compression d'image de qualité peut **réduire le temps de chargement des pages de plus de 80 % !**

## Le Lazy Loading

Si vous avez beaucoup d'images, vous pourriez envisager de les charger en lazy load. Il s'agit d'une technique d'optimisation qui charge le contenu visible mais retarde le téléchargement et le rendu du contenu qui apparaît sous la ligne de flottaison.

Le Lazy loading fonctionne comme ceci :

- Le navigateur construit le DOM de la page web sans télécharger les images et précharger les vidéos.
- Le JavaScript est utilisé pour déterminer les images à télécharger et les vidéos à précharger en fonction du contenu qui est initialement visible lorsque la page se charge. Ces images et vidéos sont téléchargées et préchargées comme il convient.
- Le téléchargement et l'affichage des vidéos supplémentaires sont retardés jusqu'à ce qu'un visiteur du site fasse défiler la page vers le bas et que du contenu supplémentaire soit affiché.

Le résultat final est que les images ne sont pas téléchargées et les vidéos ne sont pas chargées jusqu'à ce qu'elles soient réellement nécessaires. Cela peut fournir une augmentation significative des performances pour les sites qui incluent beaucoup d'images en haute résolution et des vidéos intégrées.

Heureusement, il existe de nombreux plugins WordPress gratuits qui facilitent le chargement en Lazy Load (tous sont disponibles sur [WordPress.org](https://WordPress.org)) :

- **a3 Lazy Load**

- **Lazy Load by WP Rocket**
- **Lazy Load for Videos**

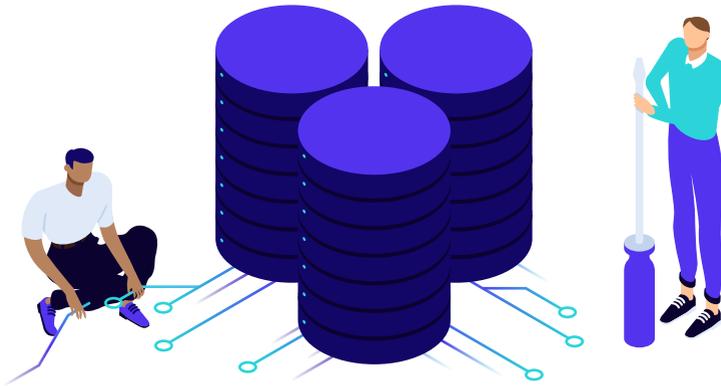
## **Autres conseils d'optimisation d'images**

Voici quelques derniers conseils d'optimisation d'images à suivre :

- Les jours de téléchargement d'images uniquement dimensionnées à la largeur de la colonne ou de la DIV sont révolus. Les images responsives sont disponibles dans WordPress (à partir de la version 4.4) et affichent automatiquement des images de plus petite taille pour les utilisateurs mobiles.
- Les SVGs peuvent être une autre alternative géniale à l'utilisation des images. Toutes les illustrations dessinées à la main que vous voyez sur le site de Kinsta sont des SVGs (vectoriels). Les SVGs sont généralement beaucoup plus petits en taille de fichier, mais pas toujours.
- Utilisez des polices d'icônes au lieu de placer du texte dans les images – elles sont plus belles au redimensionnement et prennent moins de place. Et si vous utilisez un générateur de polices, vous pouvez les optimiser encore plus. nous avons réduit la taille de 97.59% de notre fichier police d'icônes en utilisant un générateur de polices.
- Vous pouvez également utiliser le WebP, un nouveau format d'image de Google qui gagne rapidement en popularité.

CHAPITRE 10:

# Optimiser votre base de données



**Voici quelques conseils sur la façon d’optimiser votre base de données WordPress. Tout comme une voiture, votre base de données a besoin d’être entretenue car avec le temps, elle peut grossir.**

Les sites d’adhésion rendent cela particulièrement délicat, car ils **génèrent généralement des requêtes plus complexes**, ce qui à son tour ajoute une latence supplémentaire dans la récupération des informations de la base de données MySQL. Cela est dû en grande partie à toutes les pièces mobiles supplémentaires et aux grandes quantités de données dont disposent ces sites. Cela peut également être causé par des sites qui dépendent fortement des requêtes de recherche pour la navigation ou qui utilisent `WP_Query`.

Sans compter que vous avez également un grand nombre d’utilisateurs simultanés qui interrogent la base de données en permanence.

## Utiliser le moteur de stockage MySQL InnoDB

Beaucoup d’anciens sites utilisent encore le moteur de stockage MyISAM dans leur base de données. Au cours des dernières années, InnoDB s’est montrée plus performant et plus fiable.

*InnoDB est comme du carburant synthétique,  
alors que MyISAM est un carburant de base.*

Voici quelques avantages d'InnoDB par rapport à MyISAM :

- InnoDB est doté d'un **verrouillage au niveau des lignes**. MyISAM ne dispose que d'un verrouillage complet au niveau de la table. Cela permet d'accélérer le traitement de vos requêtes.
- InnoDB possède ce qu'on appelle l'intégrité référentielle qui implique la prise en charge des **clés étrangères** (RDBMS) et des contraintes relationnelles, ce que MyISAM ne fait pas (DMBS).
- InnoDB supporte les transactions, ce qui signifie que vous pouvez les valider et les annuler. Ce n'est pas le cas de MyISAM.
- InnoDB est plus fiable car il utilise les logs de **transactions** pour la récupération automatique. Ce n'est pas le cas de MyISAM.

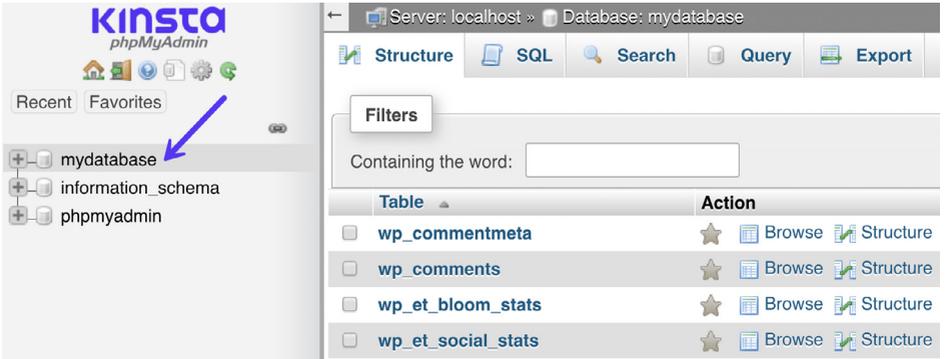
Vous vous demandez peut-être maintenant si vous utilisez InnoDB ou MyISAM ?

Si vous utilisez un site WordPress relativement nouveau, il y a de fortes chances que vous utilisiez déjà le moteur de stockage MySQL InnoDB. Mais avec les anciens sites WordPress, vous pourriez vouloir faire une vérification rapide. Certains sites peuvent même les avoir mélangés et fait correspondre les tables MyISAM et InnoDB, ainsi vous pourrez voir des améliorations en les convertissant partout.

Suivez les étapes simples ci-dessous pour vérifier.

## Étape 1

Connectez-vous à phpMyAdmin et cliquez sur votre base de données MySQL.

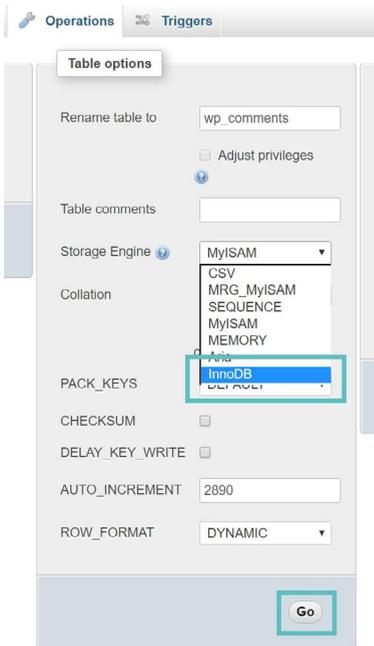


## Étape 2

Faites un balayage rapide ou un tri de la colonne “Type”, et vous pouvez voir quels types de moteurs de stockage vos tables utilisent. Dans l’exemple ci-dessous, vous pouvez voir que deux des tables utilisent toujours MyISAM.

Table	Type	Charset	Collation	Size
wp_commentmeta	InnoDB	utf8	unicode_ci	64 K <i>B</i>
wp_comments	InnoDB	utf8	unicode_ci	48 K <i>B</i>
wp_et_bloom_stats	InnoDB	utf8	unicode_ci	64 K <i>B</i>
wp_et_social_stats	InnoDB	utf8	unicode_ci	80 K <i>B</i>
	InnoDB	utf8	unicode_ci	48 K <i>B</i>
	InnoDB	utf8mb4	unicode_ci	32 K <i>B</i>
	MyISAM	utf8mb4	unicode_ci	80 K <i>B</i>
	MyISAM	utf8mb4	unicode_ci	208 K <i>B</i>
	InnoDB	utf8mb4	unicode_ci	48 K <i>B</i>
	InnoDB	utf8mb4	unicode_ci	16 K <i>B</i>

Si vous en avez trouvé, il est probablement temps de les convertir en InnoDB. Nous vous recommandons toujours de contacter votre hébergeur et de lui demander s’il peut le faire pour vous. Chez Kinsta, les tables de base de données de chaque client sont automatiquement converties en InnoDB par notre équipe de migration.



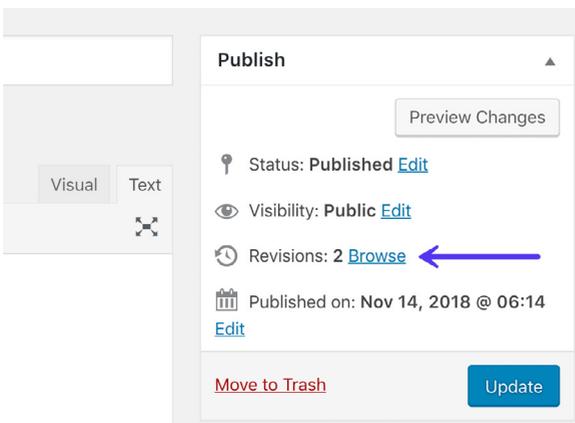
Mais vous pouvez aussi les convertir manuellement avec phpMyAdmin. Cliquez simplement sur la table MyISAM, cliquez dans l'onglet "Opérations" et changez le moteur de stockage.

Si vous êtes un développeur, une autre alternative serait de les convertir avec WP-CLI.

## Supprimer et limiter les révisions de pages et d'articles

Chaque fois que vous sauvegardez une page ou un article dans WordPress, cela crée

ce qu'on appelle une révision. Cela se produit à la fois dans les brouillons et dans les articles déjà publiés qui sont mis à jour. Les révisions peuvent être utiles au cas où vous auriez besoin de revenir à une version précédente de votre contenu.

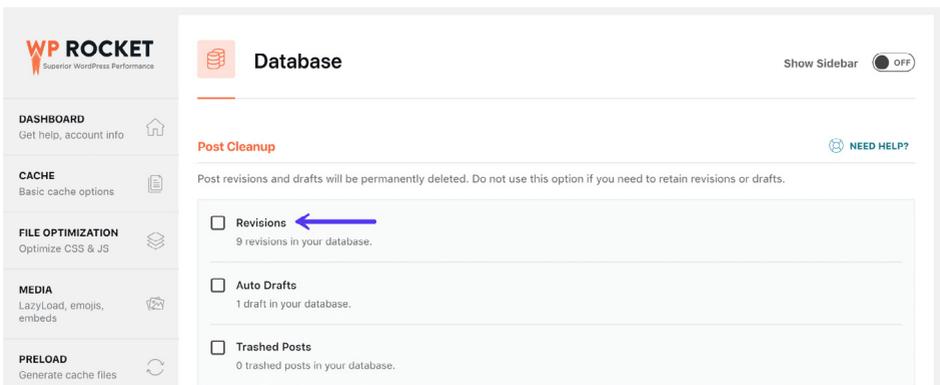


Cependant, les révisions peuvent également nuire aux performances de votre site WordPress. Sur les grands sites, cela peut ajouter très rapidement des milliers de lignes dans votre base de données qui ne sont pas forcément nécessaires. Et plus vous avez de lignes, plus votre base de données est volumineuse, ce qui occupe de l'espace de stockage. Bien que des index aient été créés à cette fin, nous avons encore vu ce problème paralyser des sites WordPress. Il y a deux ou trois choses que vous pouvez faire.

## 1. Supprimer les anciennes révisions

Si vous avez un ancien site WordPress avec beaucoup de pages et d'articles, il est peut-être temps de faire un nettoyage rapide et de supprimer ces anciennes révisions. Vous pouvez le faire avec MySQL, mais avec tous les mauvais bouts de code que l'on retrouve sur le web, nous vous recommandons de faire une sauvegarde de votre site et d'utiliser un plugin gratuit comme WP-Sweep.

Un autre de nos plugins préférés, WP Rocket, possède également une fonction d'optimisation de base de données pour éliminer les révisions.



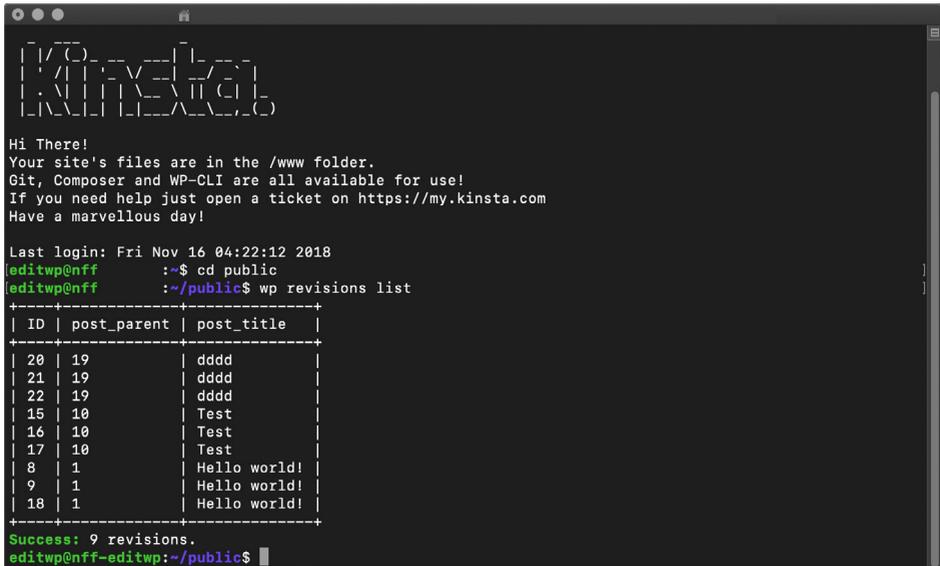
The screenshot shows the WP Rocket plugin interface. On the left is a sidebar with navigation options: DASHBOARD, CACHE, FILE OPTIMIZATION, MEDIA, and PRELOAD. The main content area is titled 'Database' and features a 'Post Cleanup' section. Below this, there is a warning: 'Post revisions and drafts will be permanently deleted. Do not use this option if you need to retain revisions or drafts.' A list of items to be cleaned up is shown with checkboxes: 'Revisions' (9 revisions in your database), 'Auto Drafts' (1 draft in your database), and 'Trashed Posts' (0 trashed posts in your database). A blue arrow points to the 'Revisions' checkbox.

Si vous êtes à l'aise avec WP-CLI, il y a quelques commandes que vous pouvez utiliser pour cela.

Connectez-vous à votre serveur via SSH et exécutez la commande suivante pour

obtenir et voir le nombre de révisions actuellement dans la base de données.

```
wp revisions list
```



```

  ____
 |  /  \ |
 | /    \|
 |:  _  |:
 | \  /  |
 |  \  /  |
 |   \  /  |
 |____\_/____|

Hi There!
Your site's files are in the /www folder.
Git, Composer and WP-CLI are all available for use!
If you need help just open a ticket on https://my.kinsta.com
Have a marvellous day!

Last login: Fri Nov 16 04:22:12 2018
editwp@nff :~$ cd public
editwp@nff :~/public$ wp revisions list
+-----+-----+-----+
| ID | post_parent | post_title |
+-----+-----+-----+
| 20 | 19           | dddd       |
| 21 | 19           | dddd       |
| 22 | 19           | dddd       |
| 15 | 10           | Test       |
| 16 | 10           | Test       |
| 17 | 10           | Test       |
|  8 | 1            | Hello world! |
|  9 | 1            | Hello world! |
| 18 | 1            | Hello world! |
+-----+-----+-----+
Success: 9 revisions.
editwp@nff-editwp:~/public$
```

Si vous obtenez une erreur, vous devrez peut-être d'abord installer le paquet wp-revisions-cli avec la commande suivante :

```
wp package install trepmal/wp-revisions-cli
```

Vous pouvez ensuite exécuter la commande suivante pour nettoyer les révisions :

```
wp revisions clean
```

## 2. Limiter les révisions

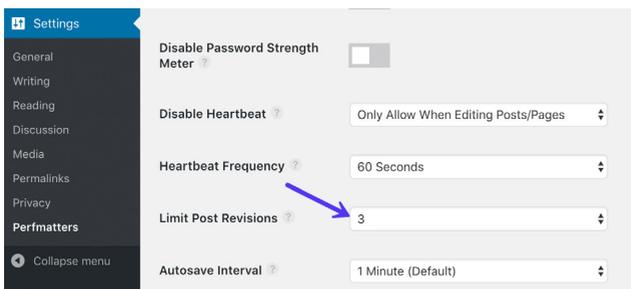
Une autre bonne stratégie que nous utilisons chez Kinsta est de limiter le nombre de révisions qui peuvent être stockées par article ou page. Même en le réglant à quelque chose comme dix, vous éviterez que les révisions ne deviennent incontrôlables, surtout si vous faites beaucoup de mises à jour.

Pour limiter les révisions, vous pouvez ajouter le code suivant à votre fichier `wp-config.php`. Le code ci-dessous doit être inséré au-dessus du 'ABSPATH' sinon il ne fonctionnera pas. Vous pouvez changer le nombre de révisions que vous voulez conserver dans votre base de données.

```
define( 'WP_POST_REVISIONS', 10 );
```

```
60 /**
61  * WordPress Database Table prefix.
62  *
63  * You can have multiple installations in one database if you give each
64  * a unique prefix. Only numbers, letters, and underscores please!
65  */
66 $table_prefix = 'wp_';
67
68 define( 'WP_POST_REVISIONS', 10 ); ←
69
70 /* That's all, stop editing! Happy blogging. */
71
72 /** Absolute path to the WordPress directory. */
73 if ( ! defined( 'ABSPATH' ) )
74     define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76 /** Sets up WordPress vars and included files. */
77 require_once ABSPATH . 'wp-settings.php';
78
```

Line 28, Column 1



Limiter les révisions avec Perfmatters

Ou vous pouvez utiliser un plugin comme Perfmatters pour limiter les révisions.

### 3. Désactiver les révisions

Enfin, vous pouvez également désactiver complètement les révisions sur votre site. Si vous empruntez cette voie, nous vous recommandons fortement de suivre la première option ci-dessus pour supprimer les révisions, puis de les désactiver par

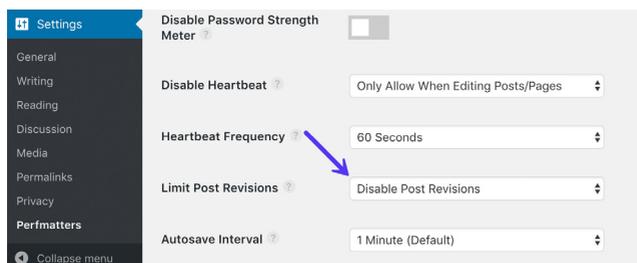
la suite. De cette façon, votre base de données est complètement libérée de toutes les anciennes révisions et aucune nouvelle ne sera ajoutée à l'avenir.

Pour désactiver les révisions, vous pouvez ajouter le code suivant à votre fichier `wp-config.php`. Le code ci-dessous doit être inséré au-dessus du `'ABSPATH'` sinon il ne fonctionnera pas.

```
define( 'WP_POST_REVISIONS', false );
```

```
60 /**
61  * WordPress Database Table prefix.
62  *
63  * You can have multiple installations in one database if you give each
64  * a unique prefix. Only numbers, letters, and underscores please!
65  */
66 $table_prefix = 'wp_';
67
68 define( 'WP_POST_REVISIONS', false ); ←
69
70 /* That's all, stop editing! Happy blogging. */
71
72 /** Absolute path to the WordPress directory. */
73 if ( ! defined( 'ABSPATH' ) )
74     define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76 /** Sets up WordPress vars and included files. */
77 require_once ABSPATH . 'wp-settings.php';
78
```

Line 33, Column 34



Ou vous pouvez utiliser un plugin comme Perfmatters pour désactiver les révisions.

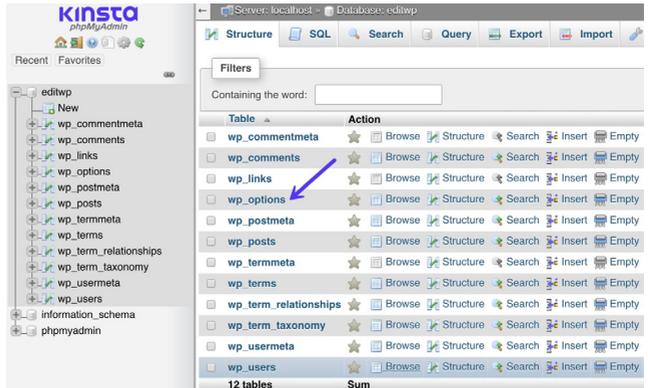
Désactiver les révisions avec Perfmatters

## Nettoyer votre table `wp_options` et vos données chargées automatiquement

La table `wp_options` est souvent négligée lorsqu'il s'agit de la performance globale

de WordPress et de la base de données. Surtout sur les sites plus anciens et plus grands, cela peut facilement être la cause de la lenteur des requêtes sur votre site en raison des données chargées automatiquement qui sont laissées derrière par des plugins et des thèmes tiers. Faites-nous confiance, nous le constatons tous les jours !

La table `wp_options` contient toutes sortes de données pour votre site WordPress telles que :



- URL du site, URL d'accueil, email de l'administrateur, catégorie par défaut, articles par page, format horaire, etc.
- Paramètres des plugins, thèmes, widgets
- Données temporairement mises en cache

Cette table contient les champs (colonnes) suivants :

- `option_id`
- `option_name`
- `option_value`
- **autoload** (c'est cette colonne qui nous intéresse quand il s'agit de performances)

+ Options

	<code>option_id</code>	<code>option_name</code>	<code>option_value</code>	<code>autoload</code>
<input type="checkbox"/> Edit Copy Delete	1	siteurl	https://editwp.com	yes
<input type="checkbox"/> Edit Copy Delete	2	home	https://editwp.com	yes
<input type="checkbox"/> Edit Copy Delete	3	blogname	editwp	yes
<input type="checkbox"/> Edit Copy Delete	4	blogdescription	Just another WordPress site	yes

Une des choses importantes à comprendre à propos de la table `wp_options` est le champ **autoload**. Il contient une valeur oui ou non (Yes et No). Ceci contrôle essentiellement si elle est chargée ou non par la `wp_load_alloptions()`. Les données chargées automatiquement sont **des données qui sont chargées sur chaque page de votre site WordPress**. Tout comme nous vous avons montré comment désactiver le chargement de certains scripts à l'échelle du site, la même idée s'applique ici. L'attribut de chargement automatique est défini à "oui" par défaut pour les développeurs, mais tous les plugins ne devraient théoriquement pas charger leurs données sur chaque page.

Le problème que les sites WordPress peuvent rencontrer est quand il y a une grande quantité de données chargées automatiquement dans la table `wp_options`. C'est généralement le résultat de ce qui suit :

- Les données sont chargées automatiquement par un plugin alors qu'elles devraient être réglées sur "no". Un bon exemple de cela serait un plugin de formulaire de contact. A-t-il besoin de charger des données sur chaque page ou seulement sur la page de contact ?
- Les plugins ou thèmes ont été supprimés du site WordPress, mais leurs options sont toujours laissées dans la table `wp_options`. Cela pourrait signifier que des données inutiles sont interrogées à chaque requête.
- Les développeurs de plugins et de thèmes chargent les données dans la table `wp_options` au lieu d'utiliser leurs propres tables. Il y a des arguments des deux côtés, car certains développeurs préfèrent les plugins qui ne créent pas de tables supplémentaires. Cependant, la table `wp_options` n'a pas non plus été conçue pour contenir des milliers de lignes.

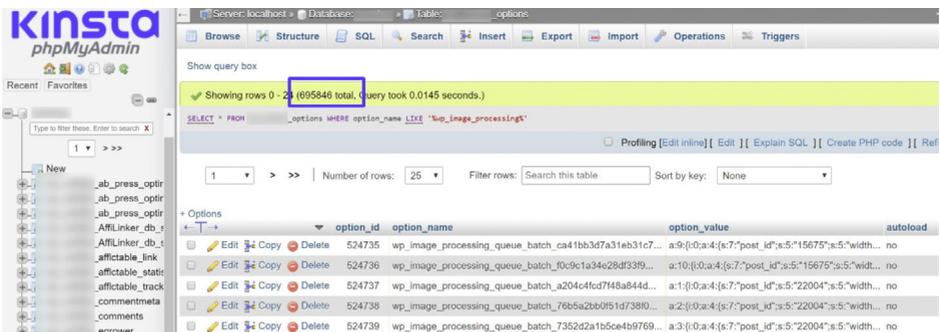
Dans quelle mesure y-a-t-il trop de données chargées automatiquement ? Cela peut varier bien sûr, mais idéalement, vous voulez que ce soit entre 300 Ko et 1 Mo. Une fois que vous commencez à vous approcher de la plage de 3 à 5 Mo ou plus, il y a très probablement des choses qui peuvent être optimisées ou supprimées du chargement automatique. Et tout ce qui dépasse 10 Mo devrait être traité immédiatement. Cela ne signifie pas toujours que cela va causer un problème, mais c'est un bon point de départ.

Chez Kinsta, nous aidons nos clients à nettoyer les données chargées automatiquement tout le temps.

*Quand avez-vous nettoyé votre table  
wp\_options pour la dernière fois ? C'est  
ce qu'on pensait. Allez-y!*

## Nettoyer les transients

Sauf si vous utilisez un cache d'objet, WordPress stocke les transients dans la table `wp_options`. En général, on leur donne un délai d'expiration et ils devraient disparaître avec le temps. Toutefois, ce n'est pas toujours le cas. Nous avons vu quelques bases de données où il y avait des milliers d'anciens enregistrements de transients. En fait, sur un site, nous avons traité des enregistrements de transients corrompus dans lesquels **plus de 695 000 lignes ont été générées** dans la table `wp_options` table. Wow !



The screenshot shows the phpMyAdmin interface. The query box contains the following SQL query:

```
SELECT * FROM `wp_options` WHERE option_name LIKE '%wp_image_processing%'
```

The result shows 695840 total rows. The table has columns: option\_id, option\_name, option\_value, and autoload. The data is as follows:

option_id	option_name	option_value	autoload
524735	wp_image_processing_queue_batch_ca41bb3d7a31eb31c7...	a:9:{i:0;a:4:{s:7:"post_id";s:5:"15675";s:5:"width...	no
524736	wp_image_processing_queue_batch_f0c9c1a34e28df339...	a:10:{i:0;a:4:{s:7:"post_id";s:5:"15675";s:5:"width...	no
524737	wp_image_processing_queue_batch_a204cfd748a844d...	a:1:{i:0;a:4:{s:7:"post_id";s:5:"22004";s:5:"width...	no
524738	wp_image_processing_queue_batch_76b5a2bb0151d7380...	a:2:{i:0;a:4:{s:7:"post_id";s:5:"22004";s:5:"width...	no
524739	wp_image_processing_queue_batch_7352d2a1b50e4b9769...	a:3:{i:0;a:4:{s:7:"post_id";s:5:"22004";s:5:"width...	no

Il est également important de noter que les transients ne sont pas chargés automatiquement par défaut. Vous pouvez utiliser une requête comme celle ci-dessous pour voir s'il y a des transients chargés automatiquement.

```
SELECT *  
FROM `wp_options`  
WHERE `autoload` = 'yes'  
AND `option_name` LIKE '%transient%'
```

Une meilleure et plus sûre option serait d'utiliser un plugin gratuit comme Transient Cleaner ou Delete Expired Transients (tous les deux disponibles sur WordPress.org) qui peut nettoyer seulement les transients de votre table `wp_options`. Cependant, il semble qu'il y ait maintenant une fonction dans WordPress, ajoutée dans la version 4.9. J'espère donc que cela est effectué automatiquement sur votre site maintenant.

WP Rocket a également la capacité de nettoyer les transients dans leurs options d'optimisation de base de données.

**ADVANCED RULES**  
Fine-tune cache rules

**DATABASE**  
Optimize, reduce bloat

**CDN**  
Integrate your CDN

**HEARTBEAT**  
Control WordPress Heartbeat API

**ADD-ONS**  
Add more features

**IMAGE OPTIMIZATION**  
Compress your images

**TOOLS**  
Import, Export, Rollback

**Comments Cleanup**  
Spam and trashed comments will be permanently deleted.

Spam Comments  
0 spam comments in your database.

Trashed Comments  
0 trashed comments in your database.

**Transients Cleanup** ←

Transients are temporary options; they are safe to remove. They will be automatically regenerated as your plugins require them.

Expired transients  
2 expired transients in your database.

All transients  
24 transients in your database.

## Nettoyer les sessions WordPress

Un autre problème courant que nous avons vu est que les tâches cron sont parfois désynchronisées ou ne fonctionnent pas correctement, et donc les sessions ne sont pas nettoyées. Vous pouvez finir par avoir des tonnes de lignes `_wp_session_` dans votre base de données. Dans cet exemple ci-dessous, le site en question a accumulé plus de 3 millions de lignes dans la table `wp_options`. Et la taille de la table était passée à plus de 600 Mo.

<input type="checkbox"/> wp_links	★	🔍 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	🚫 Drop	0	InnoDB	utf8mb4_unicode_ci	32	KiB
<input type="checkbox"/> wp_options	★	🔍 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	🚫 Drop	~3,419,339	InnoDB	utf8mb4_unicode_ci	620.9	MiB
<input type="checkbox"/> wp_postmeta	★	🔍 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	🚫 Drop	12,986	InnoDB	utf8mb4_unicode_ci	9.1	MiB
<input type="checkbox"/> wp_posts	★	🔍 Browse	📄 Structure	🔍 Search	➕ Insert	🗑 Empty	🚫 Drop	2,367	InnoDB	utf8mb4_unicode_ci	4	MiB

Vous pouvez utiliser une requête comme celle ci-dessous pour voir si vous rencontrez ce problème :

```

SELECT *
FROM `wp_options`
WHERE `option_name` LIKE '_wp_session_%'

```

+ Options

	option_id	option_name	option_value	autoload
<input type="checkbox"/> Edit Copy Delete	3301835	_wp_session_254a6d16b4400d5aabf3c48b41b72d2b	a:1:{s:10:"edd_errors";s:0:""};	no
<input type="checkbox"/> Edit Copy Delete	3301836	_wp_session_expires_254a6d16b4400d5aabf3c48b41b72d...	1520801306	no
<input type="checkbox"/> Edit Copy Delete	3301837	_wp_session_747c86aefae52149d1068a60f240249f	a:1:{s:10:"edd_errors";s:0:""};	no
<input type="checkbox"/> Edit Copy Delete	3301838	_wp_session_expires_747c86aefae52149d1068a60f24024...	1520801306	no
<input type="checkbox"/> Edit Copy Delete	3301839	_wp_session_17aa53659273873116d83993bae7cfd8	a:1:{s:10:"edd_errors";s:0:""};	no
<input type="checkbox"/> Edit Copy Delete	3301840	_wp_session_expires_17aa53659273873116d83993bae7cf...	1520801306	no
<input type="checkbox"/> Edit Copy Delete	3301841	_wp_session_f3f29f822eec97951504e4d66afcd081	a:1:{s:10:"edd_errors";s:0:""};	no
<input type="checkbox"/> Edit Copy Delete	3301842	_wp_session_expires_f3f29f822eec97951504e4d66afcd0...	1520801306	no
<input type="checkbox"/> Edit Copy Delete	3301843	_wp_session_2d4be84abb1e786ee2295061c26e60e	a:1:{s:10:"edd_errors";s:0:""};	no

Dans la plupart des cas, vous pouvez ensuite les supprimer en toute sécurité (comme une tâche cron devrait l'avoir fait) avec la commande suivante :

```

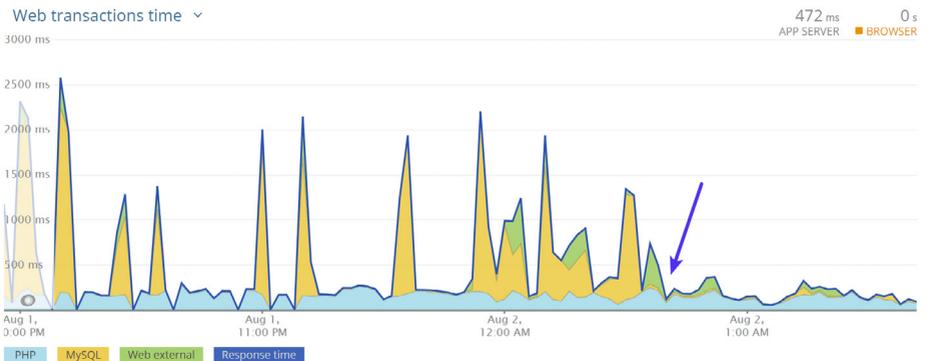
DELETE FROM `wp_options`
WHERE `option_name` LIKE '_wp_session_%'

```

Après avoir nettoyé toutes les lignes `_wp_session_` restantes, la table avait moins de 1000 lignes et a été réduite à 11 Mo.

<input type="checkbox"/> wp_links	☆	📄 Browse	🔍 Structure	🔍 Search	➕ Insert	🗑 Empty	🚫 Drop	0	InnoDB	utf8mb4_unicode_ci	32	KiB
<input type="checkbox"/> wp_options	☆	📄 Browse	🔍 Structure	🔍 Search	➕ Insert	🗑 Empty	🚫 Drop	672	InnoDB	utf8mb4_unicode_ci	11.5	MiB
<input type="checkbox"/> wp_postmeta	☆	📄 Browse	🔍 Structure	🔍 Search	➕ Insert	🗑 Empty	🚫 Drop	12,986	InnoDB	utf8mb4_unicode_ci	9.1	MiB
<input type="checkbox"/> wp_posts	☆	📄 Browse	🔍 Structure	🔍 Search	➕ Insert	🗑 Empty	🚫 Drop	2,367	InnoDB	utf8mb4_unicode_ci	4	MiB

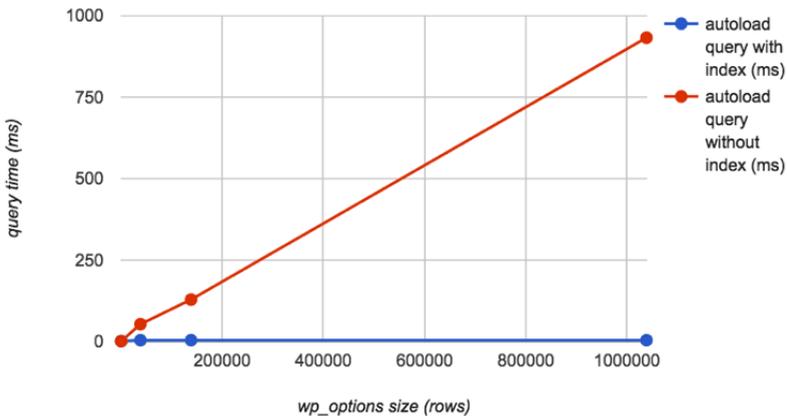
Cela a également corrigé les pics que le site recevait dans MySQL.



## Ajouter un index à Autoload

Si le nettoyage de votre table `wp_options` n'était pas suffisant, vous pouvez essayer d'ajouter un "index" au champ Autoload. Cela peut essentiellement l'aider à être recherché plus efficacement. L'équipe géniale de chez 10up a réalisé des scénarios de test sur une table `wp_options` avec un nombre typique d'enregistrements chargés automatiquement pour montrer comment l'ajout d'un index de chargement automatique aux requêtes `wp_options` peut améliorer les performances.

### Taille `wp_options` vs temps de requête



Source de l'image : 10up

## Utiliser Redis comme cache d'objet persistant pour WordPress

Redis est un magasin de structure de données clé/valeur en mémoire open source rapide. Dans le contexte de WordPress, Redis peut être utilisé pour stocker les valeurs générées par le cache d'objets natif de WordPress de façon persistante afin que les objets mis en cache puissent être réutilisés entre les chargements de pages.

L'utilisation d'un cache d'objets persistant tel que `Redis` permet de **réutiliser les objets mis en cache** plutôt que d'exiger que la base de données MySQL soit interrogée une

seconde fois pour le même objet. Le résultat est que Redis peut réduire la charge sur la base de données MySQL d'un site Web, tout en diminuant le temps de réponse du site et en augmentant la capacité du site à évoluer et à gérer le trafic supplémentaire.



Les sites Web très dynamiques (WooCommerce, sites d'adhésion, forums de discussion, blogs avec des systèmes de commentaires extrêmement actifs) qui ne peuvent pas faire bon usage de la mise en cache des pages sont des candidats potentiels pour une option de mise en cache d'objets persistants telle que Redis.

Si vous êtes un client Kinsta, nous offrons une option additionnelle Redis.

## Utiliser Elasticsearch pour accélérer la recherche dans WordPress

Elasticsearch est un moteur de recherche full-text open-source. Il est utilisé pour indexer les données et les rechercher incroyablement rapidement.

Dans le contexte de WordPress, Elasticsearch peut être utilisé pour **accélérer l'interrogation de la base de données WordPress.**



elasticsearch

Ceci est fait en construisant un index du contenu de la base de données de votre site et en utilisant Elasticsearch pour rechercher cet index beaucoup plus rapidement qu'une requête MySQL pour la même recherche.

Si vous en avez le temps et la capacité, Elasticsearch peut être intégré à un site WordPress par un développeur WordPress et Elasticsearch hautement qualifié. Si votre site fait un usage relativement standard de WP\_Query, Elasticsearch peut aussi être intégré en installant ElasticPress, un plugin WordPress gratuit de 10up, disponible sur WordPress.org, qui s'intègre automatiquement avec l'objet WP\_Query pour générer des résultats de requêtes avec Elasticsearch plutôt que MySQL.

Tout site qui fait un usage intensif de WP\_Query peut bénéficier d'Elasticsearch.

Exemples de sites qui peuvent bénéficier d'Elasticsearch :

- Sites où la recherche est le principal moyen de navigation.
- Les sites de WooCommerce avec un grand nombre de commandes où les administrateurs du site doivent pouvoir consulter régulièrement la liste des commandes.
- Tout site avec un grand nombre d'articles où les requêtes MySQL produisent des résultats d'une lenteur inacceptable.

Tout comme avec Redis, nous avons aussi une option supplémentaire Elasticsearch.

## Désactiver les fonctions non critiques intensives en base de données

Cela peut sembler un peu évident, mais cela peut faire toute la différence si vous désactivez les plugins non critiques et les fonctions de thème qui sont intenses pour la base de données.

- Les widgets et plugins populaires et ou connexes sont horribles. Ils ont généralement de lourdes requêtes à l'échelle du site.
- Les plugins d'optimisation d'images qui compressent les images à l'aide de votre serveur. Vous devriez toujours utiliser un plugin d'optimisation d'images qui optimise les images depuis l'extérieur.

Si vous visitez le blog de Kinsta et faites défiler jusqu'à la fin d'un article, vous remarquerez que nous avons ce que nous appelons des articles "choisis à la main". Ceux-ci sont sélectionnés manuellement par nos soins et affectés à l'article. Cela réduit la requête à presque rien et ne nuira pas à la performance de l'ensemble de votre site. Faut-il plus de travail ? Oui, mais cela peut être encore mieux car vous pouvez choisir ce que vous voulez que les lecteurs voient.

## Hand-picked related articles



Working Remotely –  
Everything You Need to Know  
Beforehand



The 53 Best Tools for  
Freelancers to Scale a (Real)  
Business



Where and How to Hire a  
WordPress Developer

Comment y sommes-nous parvenus ? Nous avons utilisé l'excellent plugin Advanced Custom Fields (Champs personnalisés avancés) et avons ensuite assigné ces champs à notre type d'article de blog. Cela nous permet de rechercher et d'assigner le contenu que nous voulons à chacun de nos articles de blog (voir ci-dessous).

### Post Settings

#### Related Posts

Choose hand-picked related posts here. If none are chosen (or fewer than needed for the display) we will fill the rest up with posts from within the same category/tags.

10 Best WordPress Directory Plugins to Organize Your Listings (draft)	How to Score 100/100 in Google PageSpeed Insights with WordPress
10 Best WordPress Security Plugins to Lock out the Bad Guys	10 Unique WordPress Ecommerce Strategies to Skyrocket Sales
10 High-converting WordPress Landing Page Plugins With Beauty and Optimization Combined (draft)	Conversion Rate Optimization Tips – 12 Easy Ways to Boost Sales
10 Stunning Places to Find Free Images For WordPress	
10 Things Not To Do In PHP 7	

Nous vous recommandons également de ne pas utiliser de plugins qui ajoutent un compteur de vues/articles à votre site, à moins que vous n'en ayez absolument besoin. Par exemple, évitez des choses comme « 792 messages » à côté de l'avatar d'un utilisateur dans les messages du forum ou « 5,243 vues » lorsque vous lisez les messages du forum. Lorsque vous avez une longue discussion, ces compteurs auront un impact énorme sur votre base de données. En général, minimiser l'utilisation des compteurs et ne les utiliser qu'en cas de besoin.

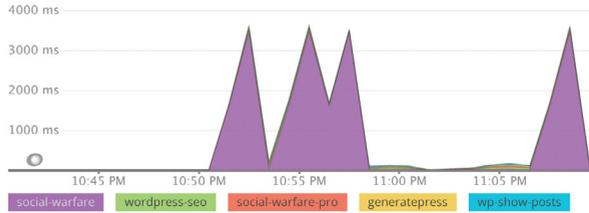
Cela vaut également pour de nombreux compteurs sociaux. Par exemple, sur ce site

ci-dessous, vous pouvez voir que le temps de réponse du populaire plugin Social Warfare est 30 fois plus long que celui du plugin suivant. La mise en cache est activée, mais évidemment, ce plugin a un coût de performances considérable. Après avoir désactivé le plugin sur le site, les temps de chargement se sont instantanément améliorés et la réactivité du tableau de bord d'administration de WordPress s'est améliorée.

Sort by

social-warfare	18,9 sec
wordpress-seo	624 ms
social-warfare-pro	449 ms
generatepress	425 ms
wp-show-posts	389 ms
aawp	356 ms
gp-premium	229 ms
nogluten	160 ms
imagify	115 ms
wp-review	107 ms

Top 5 plugins and themes  
by response time



CHAPITRE 11:

# Utiliser un réseau de diffusion de contenu (CDN)



**CDN est l'abréviation de content delivery network. Il s'agit d'un réseau de serveurs (également connus sous le nom de POPs) situés dans le monde entier. Ils sont conçus pour héberger et fournir des copies du contenu statique (et parfois dynamique) de votre site WordPress, comme des images, du CSS, du JavaScript et des flux vidéo.**

Tout d'abord, vous ne devez pas confondre un CDN avec votre hébergeur WordPress. Il s'agit de services entièrement distincts. Un CDN n'est pas un remplacement pour votre hébergeur, mais plutôt un moyen supplémentaire d'augmenter la vitesse de votre site. Alors que notre hébergement chez Kinsta est rapide, un CDN peut rendre votre site encore plus rapide.

## Comment fonctionne le CDN

Comment fonctionne exactement un CDN ? Par exemple, lorsque vous hébergez votre site Web chez Kinsta, vous devez choisir un emplacement de centre de données physique, comme les États-Unis, l'Europe, l'Asie-Pacifique ou l'Amérique du Sud.

Disons que vous choisissez US Central. Cela signifie que votre site Web est physiquement situé sur un "serveur hôte" à Council Bluffs, Iowa. Quand les gens en Europe visitent votre site web, cela va prendre plus de temps pour le charger par

rapport à quelqu'un qui le visite de Dallas au Texas par exemple.

Pourquoi ? Parce que les données doivent parcourir une plus grande distance. C'est ce qu'on appelle la latence. La latence fait référence au temps et/ou au retard impliqué dans la transmission de données sur un réseau. Plus la distance est grande, plus la latence est grande.

## Types de CDN

There are two different types of content delivery networks:

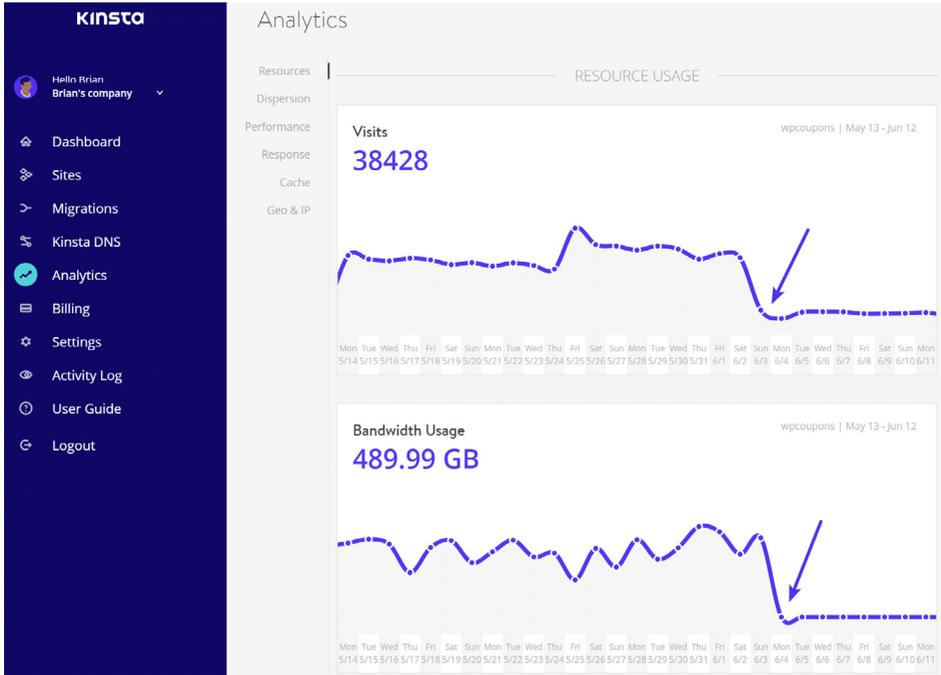
1. Le traditionnel CDN Pull
2. Le Reverse Proxy CDN

Les CDN pull traditionnels mettent en cache une copie de l'ensemble de votre contenu et de vos médias, mais une requête du client est quand même faite directement à votre fournisseur d'hébergement. KeyCDN et CDN77 sont des exemples de CDN traditionnels.

Un Reverse Proxy CDN est légèrement différent. Il agit comme un CDN, il intercepte toutes les requêtes entrantes et agit comme un serveur intermédiaire entre le client et votre hébergeur. Cloudflare et Sucuri sont des exemples de Reverse Proxy CDN. C'est l'une des raisons pour lesquelles vous devez diriger vos DNS directement vers ces fournisseurs au lieu de votre hébergeur.

L'avantage de ces derniers est qu'ils agissent en tant que serveur intermédiaire, ils peuvent fournir des pare-feux d'application web forts qui peuvent aider à bloquer le mauvais trafic pour ne pas qu'il frappe votre site WordPress et/ou fournisseur d'hébergement. L'inconvénient est qu'ils ont un peu plus de frais généraux en termes de performances par rapport à un CDN pull traditionnel. Mais avec des performances et des caractéristiques de sécurité supplémentaires, cela pourrait être considéré comme négligeable.

Voici un exemple de ce qui s'est passé après avoir activé Sucuri sur le site d'un client. Comme vous pouvez le constater, cela a eu un impact dramatique sur la quantité de mauvais trafic qui arrivait. En fin de compte, ces types de services peuvent vous aider à économiser sur vos frais d'hébergement.



## Test de vitesse du CDN

Tout à l'heure, nous avons parlé des énormes avantages de la mise en cache de WordPress. Eh bien, la mise en cache CDN est aussi super puissante. Cela s'explique par le fait que les CDNs ont généralement beaucoup plus de serveurs que les fournisseurs d'hébergement. Cela signifie qu'ils peuvent mettre en cache toutes vos ressources (images, JS, CSS) plus près de vos visiteurs et les servir à des vitesses fulgurantes.

Faisons quelques tests rapides pour voir à quel point votre site pourrait être plus rapide avec un CDN.

# Sans CDN

Notre site Web de test est hébergé chez Kinsta et est physiquement situé au centre de données de l'Iowa, aux États-Unis. Nous avons d'abord effectué cinq tests de vitesse avec Pingdom (sans le CDN activé), et avons pris la moyenne. Important : Nous utilisons l'emplacement Europe – Royaume-Uni – Londres sur Pingdom pour démontrer la réelle puissance d'un CDN. Le temps de chargement total était de **1,03 s**.



Performance grade <b>B 89</b>	Page size <b>231.0 KB</b>
Load time <b>1.03 s</b>	Requests <b>40</b>

# Avec CDN

Nous avons ensuite activé notre CDN et effectué cinq tests de vitesse supplémentaires dans Pingdom. Notre temps de chargement total est maintenant de **585 ms** depuis l'Europe – Royaume-Uni – Londres sur Pingdom. Ainsi, en utilisant le CDN, **nous avons pu réduire nos temps de chargement de page de 43,2% !** C'est énorme.

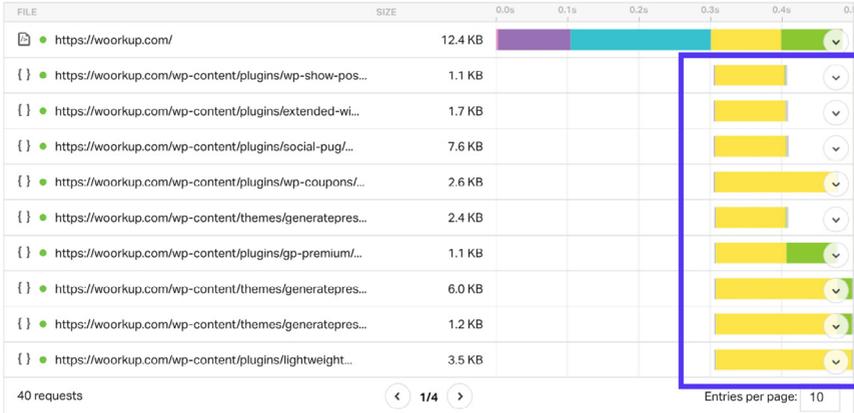


Performance grade <b>B 89</b>	Page size <b>230.6 KB</b>
Load time <b>585 ms</b>	Requests <b>40</b>

La raison d'une telle différence est que le CDN a un centre de données à Londres. Cela signifie que toutes les ressources sont mises en cache à cet emplacement et prêtes à être servies avec une latence minimale.

## TTFB sans CDN

Rappelez-vous que la barre jaune dans Pingdom représente le temps d'attente, qui est le temps du premier octet (TTFB). Lors de nos tests de vitesse sans CDN, le TTFB moyen sur les ressources était d'environ **98 ms**.



## TTFB avec CDN

Une fois le CDN activé, la moyenne du TTFB sur les ressources est tombée à **15 ms**. Ainsi, en utilisant un CDN, **notre TTFB moyen a chuté de 84,69 %**. Cela s'explique principalement par le fait que les ressources étaient servies directement à partir de la mémoire cache du CDN.



*Un CDN a réduit nos temps de chargement de page de 43,2% ! Voyez pourquoi vous devriez en utiliser un.*

## Comment activer un CDN

Activer un CDN sur votre site WordPress n'a pas besoin d'être difficile, c'est très facile ! Il suffit de suivre ces étapes.

### Étape 1

Sélectionnez un fournisseur de CDN et abonnez-vous à son service. Ceux-ci sont généralement facturés sur une base mensuelle ou en fonction de l'utilisation des données. La plupart des fournisseurs disposent d'une calculatrice pour estimer vos coûts.

- Chaque fournisseur de CDN devrait avoir de la documentation sur son site pour vous aider à démarrer.
- Nous avons également des tutoriels approfondis sur la façon d'installer Cloudflare et Sucuri disponibles sur notre site Web sur [kinsta.com](http://kinsta.com).

### Étape 2

Si vous utilisez un CDN pull traditionnel, vous pouvez utiliser un plugin gratuit comme CDN Enabler (disponible sur [WordPress.org](http://WordPress.org)), WP Rocket ou Perfmatters pour l'intégrer à votre site WordPress. Ces plugins relient automatiquement vos ressources au CDN. Aucun travail n'est nécessaire de votre part pour obtenir votre contenu sur le CDN ; c'est tout ce qu'il y a de plus facile à faire ! Les Reverse Proxy CDN n'ont généralement pas besoin de plugins, bien qu'ils en aient parfois pour activer des fonctionnalités supplémentaires.

- [Dashboard](#)
- [Kinsta Cache](#)
- [Posts](#)
- [Media](#)
- [Forms](#)
- [Pages](#)
- [Downloads](#)
- [Jilt](#)
- [Knowledge Base](#)
- [Appearance](#)
- [Plugins](#)

## Perfmatters Settings

Options
CDN
Google Analytics
Extras
License
Support

### CDN

*CDN options that allow you to rewrite your site URLs with your CDN URLs.*

**Enable CDN Rewrite** ?

**CDN URL** ?

**Included Directories** ?

**CDN Exclusions** ?

Activer le CDN dans WordPress avec Perfmatters

## Comment activer le CDN Kinsta

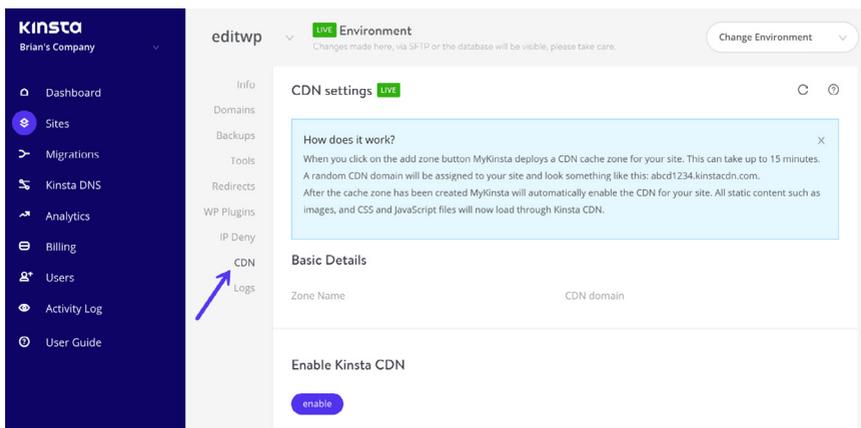
Avez-vous aimé les tests de vitesse CDN ci-dessus ? Nous utilisons KeyCDN dans ces tests. La bonne nouvelle est que notre CDN Kinsta est propulsé par KeyCDN. Il s'agit d'un réseau de diffusion de contenu HTTP/2 avec 34 POPs, pour surcharger vos ressources et médias dans le monde entier. Les régions actuellement desservies sont l'Amérique, l'Amérique du Sud, l'Europe, l'Afrique, l'Asie et l'Australie.



Si vous êtes un client Kinsta, nous incluons la bande passante CDN gratuite sur tous nos plans d'hébergement. Vous pouvez activer le CDN Kinsta en deux étapes simples.

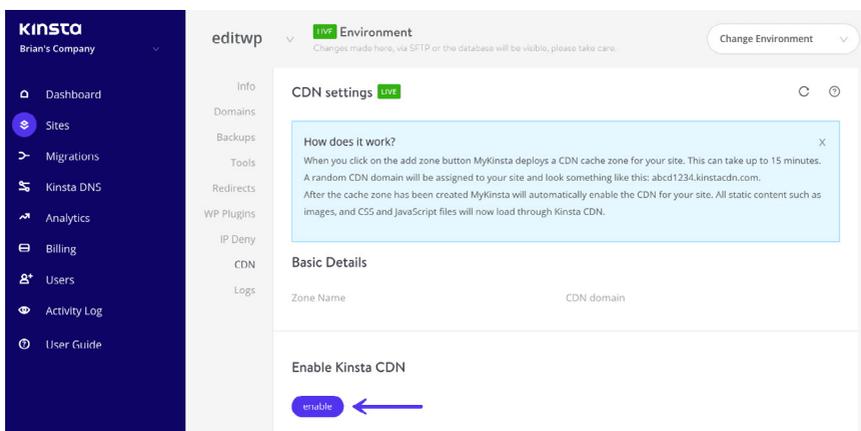
# Étape 1

Tout d'abord, connectez-vous à votre tableau de bord MyKinsta. Cliquez sur votre site, puis sur l'onglet Kinsta CDN.



# Étape 2

Cliquez ensuite sur “Activer le CDN Kinsta”. Après quelques minutes, le CDN est automatiquement déployé et vos ressources seront stockées dans le cache du monde entier. C’est tout ce qu’il y a à faire.



## Optimisations CDN additionnelles

Voici quelques autres optimisations de CDN que vous voudrez peut-être vérifier ou auxquelles vous voudrez peut-être réfléchir.

- Si vous avez beaucoup de commentaires, les avatars peuvent générer beaucoup de requêtes et charger depuis `secure.gravatar.com`. Vous pouvez plutôt charger les avatars à partir de votre CDN. Nous le faisons sur le site web de Kinsta. Si vous êtes un client Kinsta, il vous suffit de contacter notre équipe de support pour obtenir cette configuration.
- Vous pouvez également héberger des polices personnalisées, des polices Google et même votre favicon sur votre CDN. Des liens vers des tutoriels sur la façon de procéder sont disponibles sur notre site Web à l'adresse <https://kinsta.com/fr/apprendre/accellerer-wordpress/>.

CHAPITRE 12:

# Décharger vos médias et vos emails en cas de besoin



**Tout ce qui génère une requête a un impact sur la performance de votre site d'une manière ou d'une autre. Pour les sites hébergeant des centaines de milliers de fichiers ou de grands médias, il peut être judicieux de se débarrasser complètement de ces fichiers. Télécharger des médias n'est pas pareil que servir des médias à partir d'un CDN. Avec un CDN, les données originales résident toujours chez votre hébergeur, le CDN en a simplement plusieurs copies.**

Lorsque le cache expire sur vos ressources CDN, il demande à nouveau à votre hébergeur les dernières copies des fichiers. Les CDNs sont destinés à mettre en cache des fichiers pour de longues périodes de temps. Mais en raison du fait qu'il y a tant de POPs, il pourrait y avoir beaucoup de nouvelles requêtes à mesure que le cache expire dans différentes régions.

Lorsque vous téléchargez des médias ou des fichiers, cela signifie que vous devez déplacer l'emplacement physique d'origine de ces fichiers hors de votre hébergeur. Ainsi, bien qu'il puisse sembler que les fichiers sont servis à partir de votre site, ils sont en réalité situés totalement ailleurs. En plus de réduire le nombre de requêtes supplémentaires renvoyées à l'hébergeur, la raison numéro un est évidemment d'économiser également de l'espace disque.

## Décharger les médias sur Amazon S3

L'une des solutions de téléchargement les plus populaires est Amazon S3. Amazon S3 est une solution de stockage et fait partie des nombreux produits d'Amazon Web Services.

Généralement utilisé pour les grands sites qui ont besoin de sauvegardes ou qui servent de gros fichiers (téléchargements, logiciels, vidéos, jeux, fichiers audio, PDFs, etc.). Amazon a fait ses preuves en matière de fiabilité et, grâce à son infrastructure massive, peut offrir des coûts de stockage très bas. S3 compte parmi ses clients Netflix, Airbnb, SmugMug, Nasdaq, etc.



Parce qu'ils s'occupent entièrement du stockage en masse, vous pouvez presque garantir que les prix seront moins chers que votre hébergeur WordPress. Télécharger vos médias vers AWS peut être un excellent moyen d'économiser de l'argent et c'est gratuit pour votre première année (jusqu'à 5 Go de stockage). De plus, comme les requêtes pour vos médias sont servies directement depuis Amazon, cela réduit la charge sur votre site WordPress, ce qui signifie des temps de chargement plus rapides.

Nous vous recommandons d'utiliser le plugin WP Offload Media Lite for Amazon S3 (disponible sur WordPress.org) pour cela. Le plugin copie automatiquement les fichiers de votre site WordPress vers Amazon S3 au fur et à mesure qu'ils sont téléchargés dans la médiathèque. Vous pouvez également utiliser un CDN avec les médias téléchargés pour le meilleur des deux mondes.

## Télécharger des médias sur Google Cloud Storage



Une autre solution de téléchargement populaire est Google Cloud Storage. Depuis que Kinsta est propulsé par la plateforme Google Cloud Platform, nous sommes de grands fans de leur technologie et de leur infrastructure. En raison de l'infrastructure massive de Google et du fait qu'ils traitent le stockage en masse, ils peuvent offrir des coûts

de stockage très bas. Spotify, Vimeo, Coca-Cola, Philips, Evernote, et Motorola font partie de leurs clients.

Nous vous recommandons d'utiliser le plugin gratuit WP-Stateless (disponible sur WordPress.org) pour cela. Le plugin copie automatiquement les fichiers de votre site WordPress vers Google Cloud Storage lorsqu'ils sont chargés dans la médiathèque. Il offre ensuite différents modes, tels que la sauvegarde d'une copie de votre média ou même la suppression du média de WordPress pour le servir uniquement à partir de Google Cloud Storage.

## Décharger les emails transactionnels et de marketing

Que vous le pensiez ou non, les courriels ont un impact sur votre serveur et ses ressources. Avec certains hébergeurs, en particulier les hébergeurs mutualisés, abuser de cela pourrait même vous faire suspendre. Cela devient particulièrement un problème avec ceux qui essaient d'envoyer des courriels en masse. C'est la raison pour laquelle il existe des fournisseurs de messagerie transactionnelle tiers et pourquoi de nombreux hébergeurs bloquent complètement la livraison des emails sur les ports standard. **Nous ne recommandons jamais d'utiliser votre hébergeur pour le courrier électronique.**

Si vous envoyez des newsletters ou des emails en masse, nous vous recommandons toujours les alternatives suivantes pour obtenir les meilleurs résultats :

- Utiliser un logiciel d'email marketing professionnel tiers qui ne fait pas partie de WordPress
- Utiliser un fournisseur de services de messagerie transactionnelle (API HTTP ou SMTP) avec WordPress

Parmi les autres avantages de l'utilisation d'un service tiers, mentionnons les suivants :

- Meilleure délivrabilité du courrier électronique. Laissez les fournisseurs d'emails faire ce qu'ils font le mieux !

- Moins de chance d'être sur la liste noire.
- Il n'est pas toujours possible d'établir des enregistrements DMARC avec votre hébergeur.

## Outils d'Email Marketing

Quelques exemples d'emails de marketing comprennent des newsletters, des annonces de produits et de caractéristiques, des ventes, des invitations à des événements, des rappels d'intégration, etc. Voici quelques outils d'email marketing que nous vous recommandons :

- **MailChimp** – Nous utilisons MailChimp chez Kinsta.
- **MailerLite**
- **Drip**

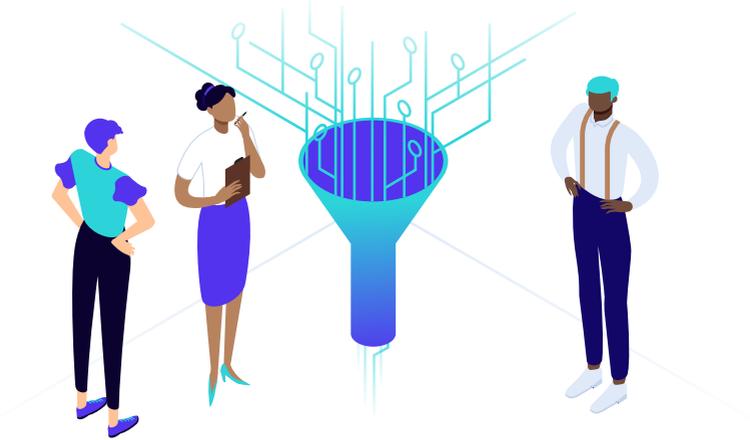
## Services d'emails transactionnels

Quelques exemples d'emails transactionnels incluent les reçus d'achat de WooCommerce ou de EDD, les avis de création de compte, les avis d'expédition, les messages d'erreur d'application, les réinitialisations de mot de passe, etc. Si vous êtes un client de Kinsta, nous comptons sur un fournisseur SMTP tiers pour assurer une haute délivrabilité. Mais en fonction de votre volume, nous vous recommandons toujours de mettre cela hors du site. Voici quelques services d'email transactionnel que nous recommandons :

- **SendGrid** – Nous utilisons SendGrid chez Kinsta.
- **Mailgun**
- **SparkPost**

CHAPITRE 13:

# Comment trouver les blocages et les plugins lents

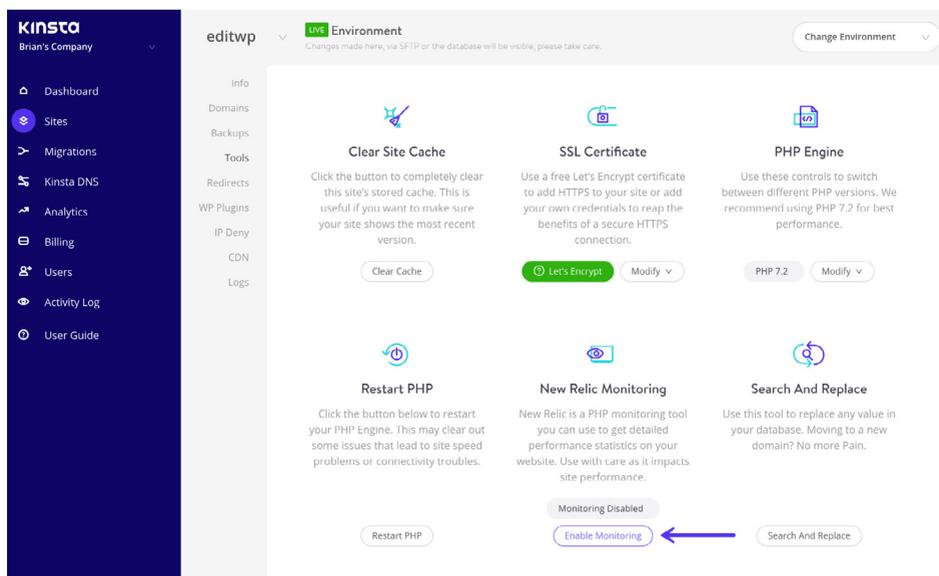


**Nous allons maintenant vous donner quelques conseils sur la façon de trouver les blocages sur votre site WordPress et ce que vous pouvez faire pour y remédier.**

## **Utiliser New Relic pour identifier les plugins et les requêtes de base de données lentes**

Il existe d'excellents outils sur le marché qui peuvent vous aider à repérer et à identifier les requêtes de base de données lentes et les plugins qui prennent beaucoup de temps. Nous sommes de grands fans de New Relic chez Kinsta et nous l'utilisons quotidiennement. New Relic est un outil de surveillance PHP que vous pouvez utiliser pour obtenir des statistiques de performances détaillées sur votre site Web.

Si vous êtes un client Kinsta, vous pouvez même ajouter votre propre clé de licence New Relic sur notre tableau de bord MyKinsta.



Cependant, utilisez New Relic avec prudence car il a un impact sur les performances du site. Il ajoute du JavaScript à votre site Web. Nous vous recommandons de l'activer lorsque vous avez besoin de dépanner la performance et de désactiver par la suite.

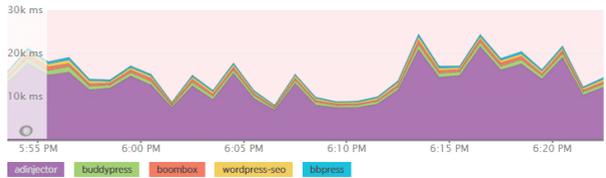
## Trouver les plugins lents

Lorsqu'un plugin WordPress cause une lenteur générale, les symptômes varient en fonction de l'activité du plugin. Cependant, dans de nombreux cas, vous constaterez qu'un plugin lent affectera chaque page d'un site WordPress. Dans le cas du site dont vous voyez les données dans l'image ci-dessous, la lenteur générale a été observée sur chaque page publique du site. Voici ce que New Relic avait à dire sur la performance des plugins du site.

Sort by Most time consuming

adinjector	390 sec
buddypress	24.4 sec
boombox	17.6 sec
wordpress-seo	15.6 sec
bbpress	12.6 sec
easy-social-share-buttons3	4.34 sec
zombify	4.22 sec

Top 5 plugins and themes  
by response time



Immédiatement vous pouvez voir que le plugin adinjector consomme plus de 15 fois plus de temps que le plugin le plus lent suivant.

Quand vous voyez des données de ce type, il peut être tentant de rejeter immédiatement la faute sur le plugin en disant qu'il est mal codé ou d'une manière ou d'une autre inefficace. Bien que ce soit parfois le cas, ce n'est pas toujours le cas. Une mauvaise configuration des plugins, la lenteur de la base de données ou des ressources externes lentes à réagir peuvent entraîner une perte de temps considérable pour un plugin.

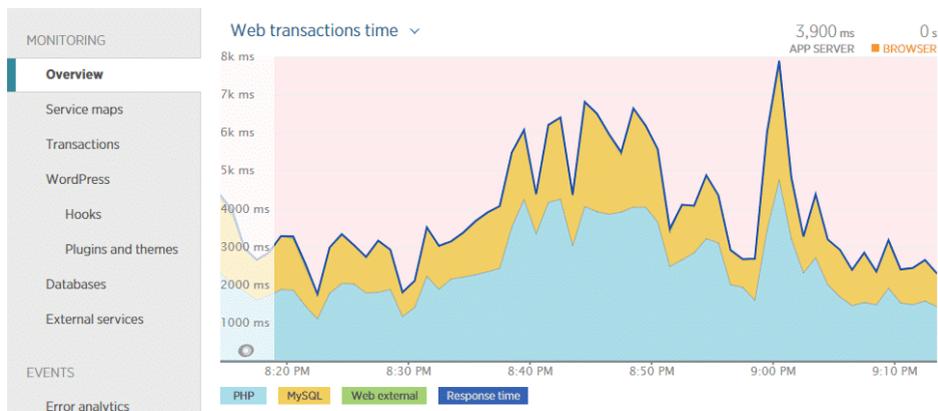
Ainsi, lorsque vous voyez un plugin qui répond lentement, c'est une bonne idée de vérifier plusieurs autres écrans dans New Relic pour trouver des informations supplémentaires. Les transactions, les bases de données et les ressources externes doivent toutes être vérifiées avant de décider que la désactivation du plugin est la meilleure ou la seule façon de progresser.

## Lenteur globale causée par une base de données submergée

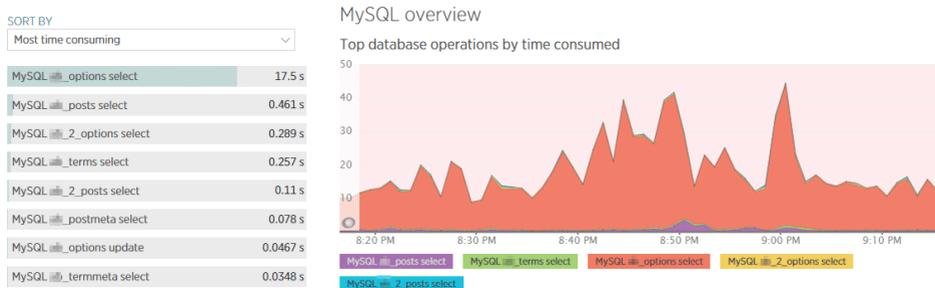
Une base de données mal optimisée peut entraîner une lenteur générale sur un site WordPress. Tout à l'heure, nous avons passé en revue beaucoup de choses différentes que vous pouvez faire pour régler ce problème. Dans New Relic, cette lenteur liée à la base de données apparaîtra très probablement à deux endroits :

- Tout d'abord, vous verrez une quantité démesurée d'activités MySQL dans l'aperçu.
- Deuxièmement, vous verrez une ou plusieurs tables de la base de données qui consomment beaucoup de temps dans l'onglet bases de données.

En commençant par l'écran de synthèse, un site avec une base de données en difficulté pourrait ressembler à ceci :



Pour avoir une meilleure idée de la table de base de données ou de la requête à l'origine du problème, allez à l'onglet des bases de données.



L'onglet Bases de données indique la table et le type de requête qui consomme le plus de temps. Si vous sélectionnez l'une des entrées de la liste, vous pouvez voir plus de détails, y compris des exemples de requêtes.

SORT BY  
Most time consuming

MySQL <code>wp_options</code> select	17.5 s
MySQL <code>wp_posts</code> select	0.461 s
MySQL <code>wp_2_options</code> select	0.289 s
MySQL <code>wp_terms</code> select	0.257 s
MySQL <code>wp_2_posts</code> select	0.11 s
MySQL <code>wp_postmeta</code> select	0.078 s
MySQL <code>wp_options</code> update	0.0467 s
MySQL <code>wp_termmeta</code> select	0.0348 s
MySQL <code>wp_2_options</code> update	0.0309 s
MySQL other	0.0281 s
MySQL <code>wp_sitemeta</code> select	0.0115 s

MySQL `wp_options` select

Slow query trace 2,510 ms /archive-product

9:14 PM MAX TIME ACTION

**Query**  
`SELECT option_name, option_value FROM wp_options WHERE autoload = ?`

**Database instance**  
`wp`

**Database name**  
`wp`

**Stack trace**

```

mysql_query called at /www/www-data/wordpress_4.7.4/public/wp-includes/wp-db.php (1887)
_db::do_query called at /www/www-data/wordpress_4.7.4/public/wp-includes/wp-db.php (1775)
in wpdb::query called at /www/www-data/wordpress_4.7.4/public/wp-includes/wp-db.php (2398)
_wpget_results called at /www/www-data/wordpress_4.7.4/public/wp-includes/option.php (189)
_lloptions called at /www/www-data/wordpress_4.7.4/public/wp-includes/functions.php (1364)
blog_installed called at /www/www-data/wordpress_4.7.4/public/wp-includes/load.php (533)
in wp_not_installed called at /www/www-data/wordpress_4.7.4/public/wp-settings.php (140)
in require_once called at /www/www-data/wordpress_4.7.4/public/wp-config.php (65)
in require_once called at /www/www-data/wordpress_4.7.4/public/wp-load.php (37)
in require_once called at /www/www-data/wordpress_4.7.4/public/wp-blog-header.php (13)
in require called at /www/www-data/wordpress_4.7.4/public/index.php (17)

```

Dans ce cas, les données pointent du doigt les données chargées automatiquement dans la table `wp_options`. Souvenez-vous, on en a déjà parlé. Bien sûr, une analyse rapide de la table `wp_options` confirme que près de 250 Mo de données sont chargées automatiquement à partir de cette table, faisant de ce site un candidat évident pour la maintenance et l'optimisation de bases de données.

Nous avons également des tutoriels supplémentaires disponibles sur [kinsta.com](https://kinsta.com) sur la façon d'utiliser New Relic pour déboguer les problèmes de performances sur votre site WordPress.

## Utiliser le plugin gratuit Query Monitor

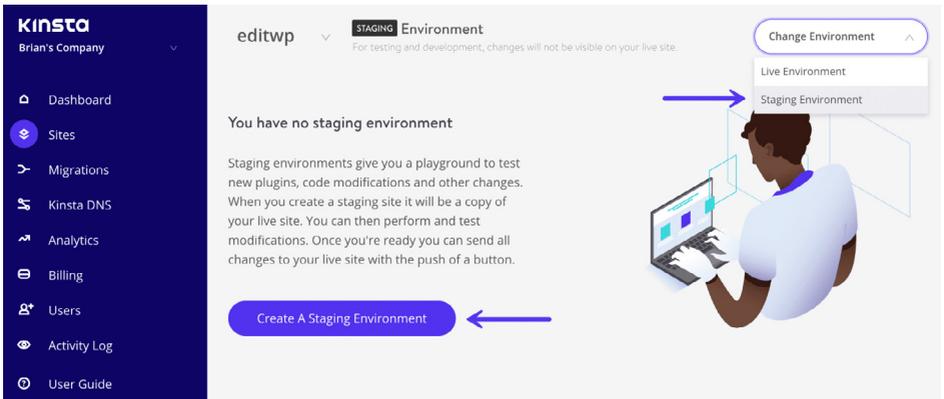
Vous pouvez également utiliser le plugin WordPress gratuit Query Monitor (disponible sur [WordPress.org](https://WordPress.org)). Utilisez-le pour identifier et déboguer les requêtes lentes des bases de données, les appels AJAX, les requêtes REST API, et bien plus encore. De plus, le plugin rapporte les détails du site Web tels que les dépendances de script et les hooks WordPress qui se déclenchent pendant la génération des pages, les détails de l'environnement d'hébergement, les balises de requête conditionnelles rencontrées par la page courante, et beaucoup plus.

Query Monitor				
Page generation time	Peak memory usage	Database query time	Database queries	Object cache
0.0321 0.0% of 0s limit	2,048 kB 0.0% of 16,777,216 kB limit	0.0022	SELECT: 3	99.9% hit rate External object cache: true
\$wpdb Queries				
Query	Caller	Component	Rows	Time
1 SELECT user_id, meta_key, meta_value FROM wp_usermeta WHERE user_id IN (1) ORDER BY umeta_id ASC	update_meta_cache() wp-includes/meta.php:826	+ Plugin: coming-soon-builder	39	0.0004
2 SELECT wp_posts.* FROM wp_posts WHERE 1=1 AND (wp_posts.ID = '2') AND wp_posts.post_type = 'page' ORDER BY wp_posts.post_date DESC	WP_Query->get_posts() wp-includes/query.php:3655	+ Core	1	0.0017
3 SELECT * FROM wp_posts WHERE (post_type = 'page') AND post_status = 'publish') AND post_parent = 2 ORDER BY wp_posts.post_title ASC LIMIT 0,1	get_pages() wp-includes/post.php:4598	+ Core	0	0.0001
Total Queries: 3				0.0022

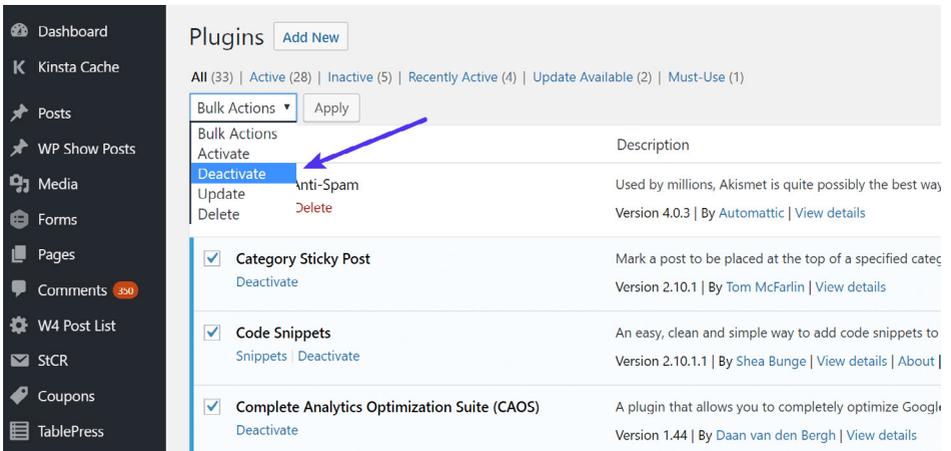
Le plugin a été développé par John Blackburn, un committer du cœur de WordPress qui est actuellement développeur chez Human Made et qui était auparavant employé par WordPress.com VIP – en d’autres termes, quelqu’un qui connaît bien WordPress. Query Monitor a été ajouté au répertoire des plugins WordPress en 2013 et compte actuellement plus de 10 000 installations actives – une somme impressionnante pour un plugin de développement. L’évaluation de cinq étoiles sur cinq du plugin par les utilisateurs aide à expliquer sa popularité auprès des développeurs.

## Utilisation des sites de développement sans toucher à la production

Nous ne savons pas ce que nous ferions sans les environnements de développement (staging). Ces outils peuvent être d’une valeur inestimable lorsqu’il s’agit de dépanner les problèmes de performances. Heureusement, Kinsta dispose d’environnements de développement activables en un clic. Si votre hébergeur WordPress n’offre pas d’environnements de staging, vous pouvez aussi utiliser un plugin comme WP Staging (disponible sur WordPress.org), bien que ce ne soit pas aussi facile.

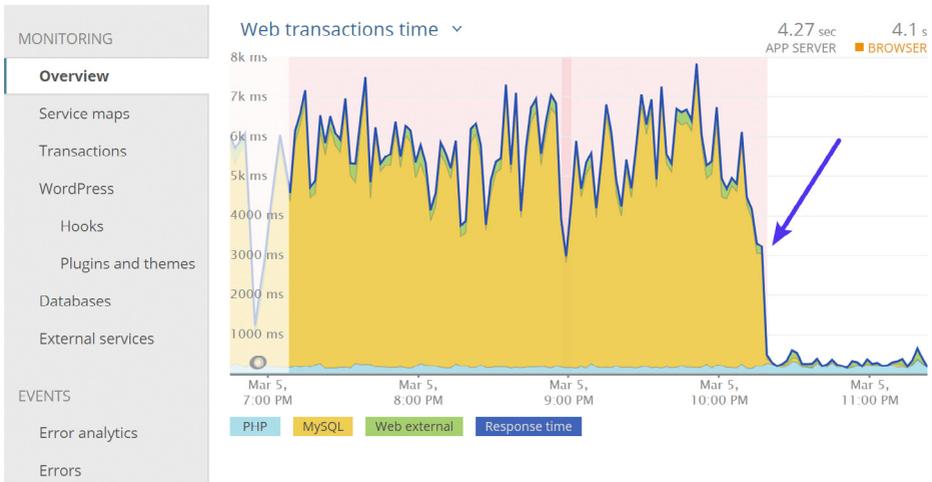


Une fois que vous avez un site de développement opérationnel, la première chose que vous pouvez faire est de désactiver tous vos plugins. Puisqu'il s'agit d'une copie de votre site en production, vous n'avez pas à vous soucier de casser quoi que ce soit. C'est de loin l'un des moyens les plus faciles de réduire les problèmes. Il vous suffit d'aller dans Plugins, de les sélectionner tous et de choisir "Désactiver" parmi les options en masse.



Après avoir fait cela, vous pouvez surveiller les temps de réponse dans New Relic ou Query Monitor et voir ce qui se passe. Dans cet exemple ci-dessous, les temps de réponse sont immédiatement revenus à la normale sur le site, nous savons donc que c'était l'un des plugins qui causait un problème. Vous pouvez ensuite les réactiver un par

un, en répétant le même processus jusqu'à ce que vous trouviez le coupable.



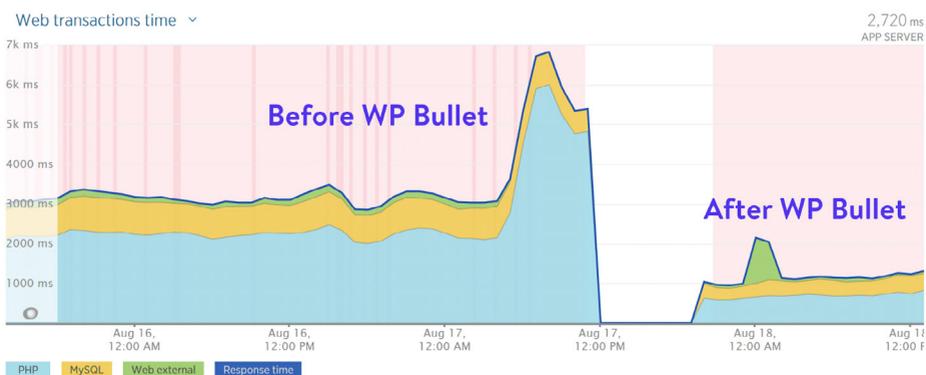
Voici un exemple de ce qui s'est passé lorsque nous avons activé le plugin qui causait le problème. Les temps de chargement (temps de transaction Web) ont immédiatement augmenté.



Que faire après avoir trouvé le plugin qui cause la lenteur ? Voici ce que nous vous conseillons :

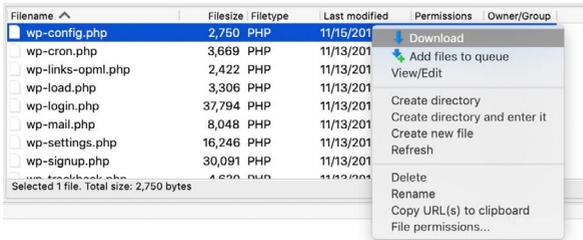
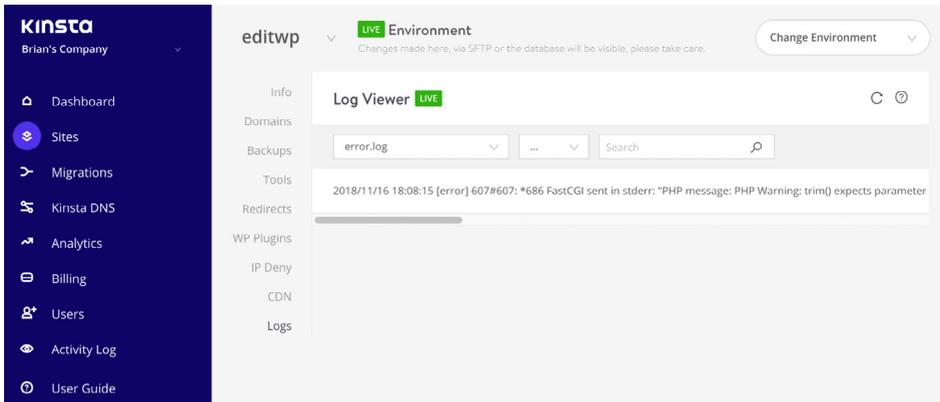
1. Mettez à jour vos plugins et thèmes avec la dernière version si vous ne l'avez pas déjà fait.
2. Contactez le développeur du plugin ou du thème et demandez-lui de l'aide.
3. Trouvez un plugin alternatif qui peut offrir la même fonctionnalité.
4. Peut-être que votre version de PHP cause un problème. Changez votre moteur PHP pour une version inférieure et voyez si le plugin ou le thème fonctionne.

Vous pouvez également engager un développeur WordPress pour résoudre le problème. Si c'est lié à la performance, nous devons donner un message personnel à Mike Andreason de WP-Bullet.com. C'est un développeur Codeable à temps plein spécialisé dans l'optimisation des performances, qui a aidé de nombreux clients ici chez Kinsta avec des installations complexes à faire passer leur site à un niveau supérieur.



## Vérifier vos logs d'erreurs

Vérifier les logs d'erreurs n'est jamais amusant, mais peut révéler beaucoup de choses sur les problèmes de performances avec les plugins WordPress. Si vous êtes un client Kinsta, vous pouvez facilement visualiser vos logs d'erreurs, vos logs de cache et vos logs d'accès directement depuis le tableau de bord MyKinsta.



Vous pouvez également activer les logs d'erreurs en ajoutant du code à votre fichier `wp-config.php`. Tout d'abord, vous devrez vous connecter à votre site via SFTP.

Téléchargez ensuite votre

`wp-config.php` pour pouvoir l'éditer. Note : Faites toujours une sauvegarde de ce fichier d'abord !

Trouvez la ligne qui dit `/* That's all, stop editing! Happy blogging. */` et juste avant, ajoutez ce qui suit (comme indiqué ci-dessous) :

```
define( 'WP_DEBUG', true );
```

```

37
38
39
40 define( 'WP_DEBUG', true );
41
42 /* That's all, stop editing! Happy blogging. */
43
44 /** Absolute path to the WordPress directory. */
45 if ( !defined( 'ABSPATH' ) )
46     define( 'ABSPATH', dirname( __FILE__ ) . '/' );
47
48 /** Sets up WordPress vars and included files. */
49 require_once( ABSPATH . 'wp-settings.php' );

```

Si le code ci-dessus existe déjà dans votre fichier `wp-config.php` mais est réglé sur “false”, changez-le simplement en “true”. Ceci activera le mode de débogage.

Remarque : Vous verrez peut-être également des avertissements ou des erreurs dans votre administration WordPress.

Vous pouvez alors activer le journal de débogage pour envoyer toutes les erreurs à un fichier en ajoutant le code suivant juste après la ligne `WP_DEBUG` (comme indiqué ci-dessous) :

```
define( 'WP_DEBUG_LOG', true );
```

```
38
39
40 define( 'WP_DEBUG', true );
41 define( 'WP_DEBUG_LOG', true );
42
43 /* That's all, stop editing! Happy blogging. */
44
45 /** Absolute path to the WordPress directory. */
46 if ( !defined('ABSPATH') )
47     define('ABSPATH', dirname(__FILE__) . '/');
48
49 /** Sets up WordPress vars and included files. */
50 require_once(ABSPATH . 'wp-settings.php');
```

Sauvegardez vos modifications et envoyez-les sur votre serveur. Les erreurs seront alors enregistrées dans le fichier `debug.log` dans votre dossier `/wp-content/`. Si pour une raison quelconque vous ne voyez pas ce fichier, vous pouvez toujours en créer un.

## Utiliser MyKinsta Analytics

Si vous êtes un client Kinsta, vous pouvez tirer parti des informations sur les performances que nous avons intégrées dans notre outil MyKinsta Analytics.

Sous la section de surveillance des performances, vous pouvez voir votre temps de réponse moyen PHP + MySQL, le débit PHP, l'utilisation d'AJAX, le top des temps de Upstream moyen et maximaux.





# Top des temps de réponse moyens PHP + MySQL

Cette liste montre le top des temps de réponse moyens de PHP et MySQL. Ces nombres peuvent être des pics ponctuels, il est donc conseillé de comparer cette liste avec le “Top des temps maximum de Upstream”.

PATH	REQUESTS	TIME
/wp-admin/update.php?action=upload-plugin	1	2.19 s
/wp-cron.php?server_triggered_cronjob	93	1.97 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&plugin_status=all&paged=1&s&_wpnonce=70f7d2faca	1	1.47 s
/wp-admin/plugins.php?activate=true&plugin_status=all&paged=1&s=	2	1.37 s
/wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=729c628974	1	1.07 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=70f7d2faca	1	0.91 s
/wp-cron.php?doing_wp_cron=1542348999.8110280036926269531250	1	0.67 s
/wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=113bcb7711	1	0.53 s
/wp-admin/plugins.php	3	0.49 s
/wp-admin/plugin-install.php	1	0.48 s

## Top des temps maximum de Upstream

Le top des temps maximum de Upstream est le temps total nécessaire à NGINX

(et aux serveurs en amont) pour traiter une requête et envoyer une réponse. Le temps est mesuré en secondes, avec une résolution en millisecondes.

PATH	REQUESTS	TIME
/wp-cron.php?server_triggered_cronjob	93	180.00 s
/wp-admin/plugins.php?activate=true&plugin_status=all&paged=1&s=	2	2.60 s
/wp-admin/update.php?action=upload-plugin	1	2.19 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&plugin_status=all&paged=1&s&_wpnonce=70f7d2faca	1	1.47 s
/hello-world/	5	1.46 s
/wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=729c628974	1	1.07 s
/	83	0.93 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=70f7d2faca	1	0.91 s
/wp-admin/plugins.php	3	0.85 s
/wp-cron.php?doing_wp_cron=1542348999.8110280036926269531250	1	0.67 s

## Votre site pourrait être piraté

Si vous rencontrez des problèmes de performances, il se peut très bien que votre site soit piraté, infecté par des malwares ou victime d'une attaque DDoS. Cela peut avoir un impact sur la vitesse de votre site et même sur la réactivité de votre tableau de bord d'administration WordPress. Dans ces cas, nous recommandons ce qui suit :

1. Implémenter un serveur proxy et WAF tel que Cloudflare ou Sucuri.
2. Bloquer les mauvaises adresses IP en utilisant les services ci-dessus ou si vous êtes un client Kinsta vous pouvez également bloquer les adresses IP depuis notre tableau de bord MyKinsta.
3. Vous pouvez également implémenter le blocage géographique. Certains pays sont vraiment mauvais quand il s'agit de la qualité du trafic qu'ils génèrent. Si vous êtes attaqué, vous devrez peut-être bloquer le pays en entier, temporairement ou définitivement.

## Dépannage avec les codes d'erreur (codes d'état HTTP)

Les codes d'état HTTP sont comme une courte note du serveur Web qui est collée en haut d'une page Web. Ça ne fait pas partie de la page web. Il s'agit plutôt d'un message du serveur vous informant de l'évolution de la situation lorsque le serveur a reçu la demande de consultation de la page. Ils peuvent être d'une valeur inestimable lorsqu'il s'agit de dépannage !

Bien qu'il existe plus de 40 codes d'état différents, voici les plus courants auxquels les utilisateurs de WordPress sont confrontés. Nous avons des tutoriels disponibles sur [kinsta.com](https://kinsta.com) avec des instructions sur la façon de corriger chacun des codes d'erreur suivants :

**429: "Too many requests."** Générée par le serveur lorsque l'utilisateur a envoyé trop de requêtes dans un laps de temps donné (limitation de débit).

Cela peut parfois se produire à partir de robots ou de scripts qui tentent d'accéder à votre site.

Dans ce cas, vous pouvez essayer de changer votre URL de connexion WordPress.

### 429 Too Many Requests

nginx

**500: "There was an error on the server and the request could not be completed."** Un code générique qui signifie simplement « erreur interne du serveur ». Quelque chose a mal tourné sur le serveur, et la ressource demandée n'a pas été livrée. Ce code est généralement généré par des

plugins tiers, un PHP défectueux, ou même une rupture de connexion à la base de données. **Error establishing a database connection**

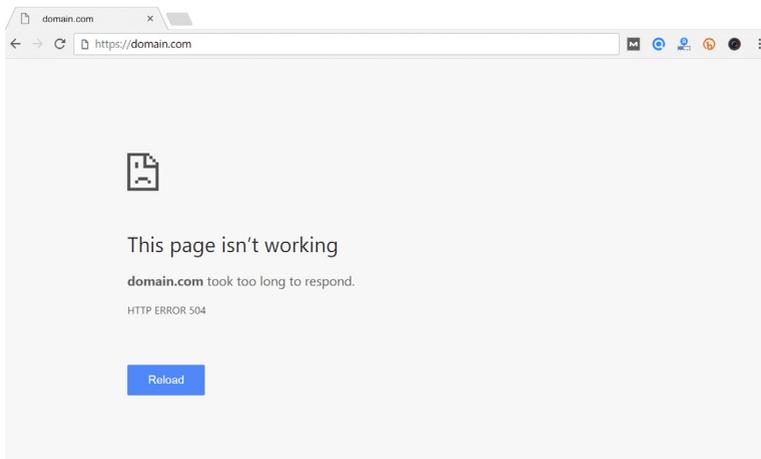
**502: “Bad Gateway.”** Ce code d’erreur signifie généralement qu’un serveur a reçu une réponse invalide d’un autre serveur. Il arrive qu’une requête prenne trop de temps, qu’elle soit annulée ou supprimée par le serveur et que la connexion à la base de données soit interrompue.



**503: “The server is unavailable to handle this request right now.”**

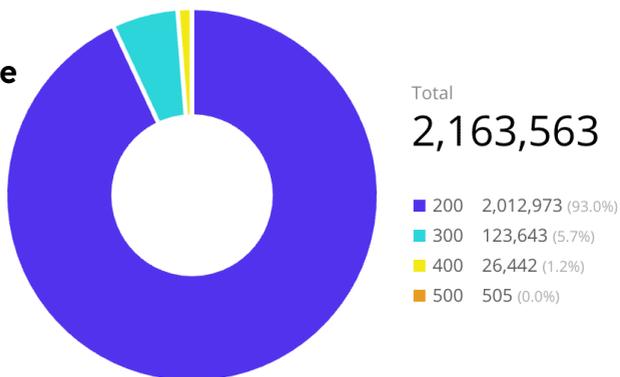
La demande ne peut être traitée pour l’instant. Ce code peut être renvoyé par un serveur surchargé qui n’est pas en mesure de traiter des requêtes supplémentaires.

**504: “The server, acting as a gateway, timed out waiting for another server to respond.”** Ce code est retourné lorsqu’il y a deux serveurs impliqués dans le traitement d’une requête, et que le premier serveur attend la réponse du second serveur.



Vous pouvez également consulter ces codes de réponse HTTP dans notre outil MyKinsta Analytics. Notre rapport de répartition des codes de réponse vous permet de voir un aperçu de la répartition des codes d'état HTTP servis pour les ressources demandées.

## Répartition des codes de réponse



Le rapport des statistiques de réponse vous permet de voir le nombre total de redirections en cours, les erreurs, le taux de réussite et le taux d'erreurs. Chaque site WordPress aura généralement un petit taux d'erreur, ce qui est tout à fait normal.

## Statistiques de réponse

**513,037**  
Redirects

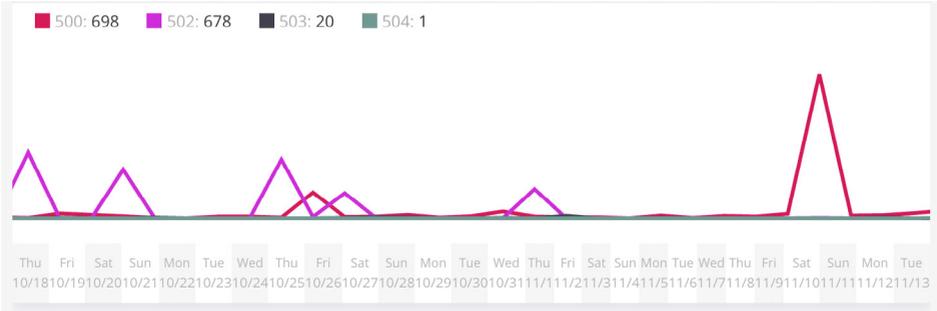
**119,796**  
Errors

**99.3%**  
Success Rate

**0.7%**  
Error Ratio

Il existe ensuite des rapports de panne pour chaque type de code d'erreur, tels que les erreurs 500, 400, les redirections, etc.

# Répartition des erreurs 500



CHAPITRE 14:

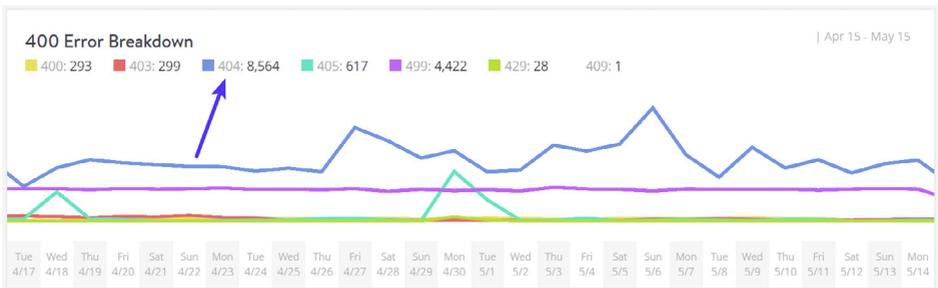
# Conseils sur l'optimisation du Back-End



Nous allons maintenant nous pencher sur les moyens d'accélérer WordPress en optimisant le back-end. Le back-end implique généralement tout ce qui est entièrement géré par le serveur, comme PHP, les en-têtes de cache HTTP, la compression GZIP, etc.

## Créer une page 404 légère

Nous avons vu que les sites très dynamiques génèrent généralement beaucoup d'erreurs 404. Votre site Web peut en générer plus que vous ne le pensez ! Notre outil d'analyse MyKinsta peut vous aider à déterminer la quantité exacte (voir ci-dessous).



La raison pour laquelle ces erreurs sont mauvaises est que de nombreuses pages 404

sont très gourmandes en ressources. Pour un site WordPress très dynamique, vous devrez éviter une page 404 lourde. Créez un modèle de 404 simple qui évite si possible d'interroger la base de données. Et bien sûr, passez un peu de temps à corriger les erreurs 404, car ce n'est pas seulement exigeant en ressources, c'est tout simplement mauvais pour l'expérience utilisateur.

## Augmenter le nombre de workers PHP

Les workers PHP sont peut-être un terme dont vous n'avez jamais entendu parler, mais il y a de nombreux hébergeurs, y compris Kinsta, qui gèrent la limitation des requêtes (plutôt que de vous limiter par CPU ou RAM, ce qui est généralement le cas des fournisseurs d'hébergement mutualisé).

Les workers PHP déterminent **combien de requêtes simultanées votre site peut traiter à un moment donné**. En d'autres termes, chaque requête non mise en cache pour votre site Web est traitée par un worker PHP. Par exemple, si vous avez 4 requêtes qui arrivent sur votre site exactement en même temps et que votre site a 2 workers PHP, deux de ces requêtes seront traitées tandis que les deux autres devront attendre que les deux premières aient fini de les traiter.

Rappelez-vous que nous avons discuté plus tôt que l'un des plus gros problèmes avec les sites WordPress d'adhésion est toutes ces demandes non mises en cache. C'est pourquoi les workers PHP deviennent très importants car ils doivent travailler pour chaque demande. Par conséquent, ces sites auront généralement besoin de workers PHP supplémentaires pour s'assurer que chaque demande est traitée sans délai et traitée avec succès.

Que se passe-t-il si vous atteignez continuellement le maximum du nombre de vos workers PHP ? Fondamentalement, la file d'attente commence à repousser les demandes plus anciennes, ce qui pourrait entraîner des erreurs 500 sur votre site. Chacun des plans d'hébergement de Kinsta inclut un nombre prédéfini de workers PHP. Si vous avez de la difficulté à estimer les besoins de votre site, vous pouvez toujours discuter avec notre équipe de vente ou de support.



pratique pour le hotlinker, mais c'est en fait du vol car il utilise les ressources du site qui en est victime. C'est comme si on prenait notre voiture et qu'on s'en allait avec l'essence qu'on siphonnait de la voiture de notre voisin.

*Le Hotlinking, c'est comme partir en voiture avec de l'essence que vous avez siphonnée dans la voiture de votre voisin.*

Le hotlinking peut représenter **un énorme gaspillage de ressources pour le serveur cible**. Imaginez si vous êtes sur un hébergeur WordPress mutualisé et que le Huffington Post utilise soudainement vos images. Vous pourriez passer de quelques centaines de requêtes par heure sur votre site à quelques centaines de milliers. Cela pourrait même entraîner la suspension de votre compte d'hébergement. C'est une raison pour non seulement utiliser un hébergeur haute performance (qui peut gérer des imprévus comme celui-ci), mais aussi activer la protection hotlink, afin que cela ne se produise pas.

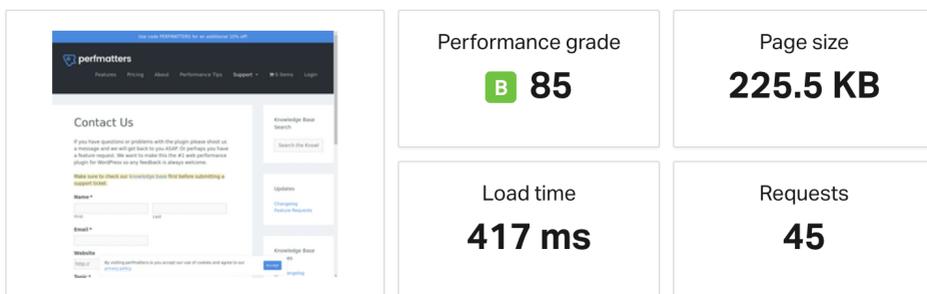
## Minimiser les redirections et les ajouter au niveau du serveur

Ne pas avoir trop de redirections, c'est toujours quelque chose dont il faut faire attention. Des redirections simples comme une simple redirection 301, HTTP vers HTTPS, ou www vers non-www (vice versa) sont acceptables. Et souvent ces dernières sont nécessaires dans certaines parties de votre site Web. Cependant, chacune a un coût sur la performance de votre site. Et si vous commencez à empiler les redirections les unes sur les autres, il est important de réaliser leur impact sur votre site. Cela s'applique aux redirections de pages et d'articles, aux redirections d'images, à tout.

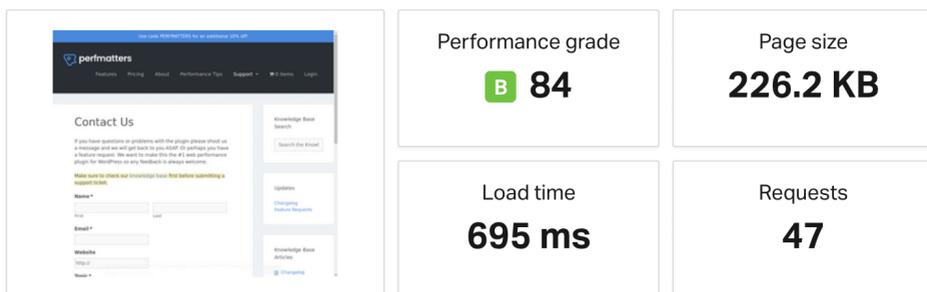
Une redirection générera un code d'état 301 ou 302 dans l'en-tête de réponse.

FILE	SIZE	0.0s	0.1s	0.2s
<a href="http://www.perfmatters.io/contact">http://www.perfmatters.io/contact</a>	238.0 B	<div style="width: 100%; height: 10px; background-color: #00a0c0;"></div>	<div style="width: 100%; height: 10px; background-color: #ffeb3b;"></div>	<div style="width: 100%; height: 10px; background-color: #ffeb3b;"></div>
Response Headers <b>301</b> ← <a href="http://www.perfmatters.io/contact">http://www.perfmatters.io/contact</a>				
<b>content-length</b>	162			
<b>x-content-type-options</b>	nosniff			

Dans quelle mesure les redirections ont-elles un impact sur votre site ? Faisons un petit test. Tout d'abord, nous effectuons un test de vitesse sur notre page Contactez-nous <https://perfmatters.io/contact/>. Comme vous pouvez le voir ci-dessous, nous obtenons un temps de chargement total de **417 ms**.

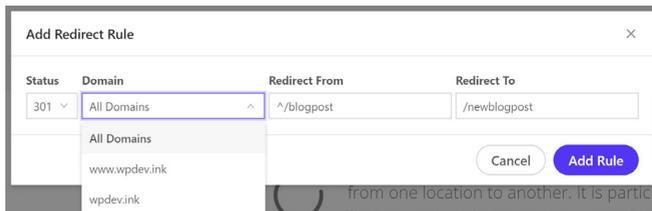


Nous modifions ensuite légèrement l'URL et effectuons un autre test de vitesse pour voir l'impact des redirections multiples. <http://www.perfmatters.io/contact>. Comme vous pouvez le voir, il faut maintenant **695 ms** pour charger la même page. C'est une augmentation de 66%. Wow !



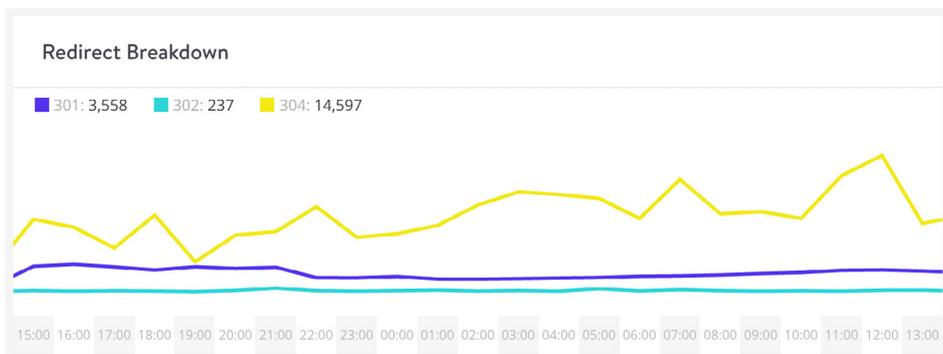
L'utilisation de plugins WordPress gratuits pour implémenter des redirections peut parfois causer des problèmes de performances car la plupart d'entre eux utilisent

la fonction `wp_redirect`, qui nécessite une exécution de code et des ressources supplémentaires. Certains d'entre eux ajoutent aussi des données chargées automatiquement à votre table `wp_options`, ce qui augmente le gonflement de la base de données. Les ajouter au niveau du serveur est l'endroit où cela devrait être toujours fait. C'est ce que nous vous permettons de faire chez MyKinsta avec notre outil de règles de redirection.



Status	Domain	Redirect From	Redirect To
301	All Domains	^/blogpost	/newblogpost

Vous pouvez également voir une répartition complète du nombre de redirections sur vos sites dans notre outil MyKinsta Analytics. Voir le nombre total de 301, 302 et 304.



## Ne laissez pas les tâches cron devenir incontrôlables

Les tâches CRON (WP-Cron) sont utilisées pour planifier des tâches répétitives pour votre site WordPress. Cependant, avec le temps, elles peuvent devenir incontrôlables et causer des problèmes de performances. Vous pouvez utiliser le plugin gratuit WP Control (disponible sur WordPress.org) pour vérifier et gérer toutes les tâches Cron sur votre site.

Nous avons également constaté des problèmes de performances avec le gestionnaire

de Cron intégré de WordPress : WP-Cron. Si un site n'a pas assez de workers PHP, il arrive parfois que lorsqu'une requête arrive, WordPress génère le cron, mais le cron doit attendre le worker, et donc il reste là. Une meilleure approche est de désactiver WP-Cron et d'utiliser le cron système à la place. Ceci est même recommandé dans le manuel officiel du Plugin.

Pour désactiver WP-Cron, ajoutez ce qui suit à votre fichier `wp-config.php` juste avant la ligne qui dit "That's all, stop editing! Happy blogging." Note : Ceci le désactive pour qu'il ne s'exécute pas lors du chargement de la page, ou lorsque vous l'appellez directement `wp-cron.php`.

```
define('DISABLE_WP_CRON', true);
```

```
60  /**
61   * WordPress Database Table prefix.
62   *
63   * You can have multiple installations in one database if you give each
64   * a unique prefix. Only numbers, letters, and underscores please!
65   */
66  $table_prefix = 'wp_';
67
68  define('DISABLE_WP_CRON', true); ←
69
70  /* That's all, stop editing! Happy blogging. */
71
72  /** Absolute path to the WordPress directory. */
73  if ( ! defined( 'ABSPATH' ) )
74      define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76  /** Sets up WordPress vars and included files. */
77  require_once ABSPATH . 'wp-settings.php';
78
```

Line 68, Column 33

Vous devrez ensuite planifier `wp-cron.php` à partir de votre serveur. Si vous êtes un client Kinsta, les crons du système sont déjà activés et fonctionnent toutes les 15 minutes par défaut. Vous pouvez également faire augmenter la fréquence en vous adressant à notre équipe de support. Si vous êtes familier avec SSH, vous pouvez gérer les crons de serveur depuis la ligne de commande.

## Ajout des en-têtes Cache-Control et Expires Headers (détermine la longueur du cache)

Chaque script de votre site WordPress doit être accompagné d'un en-tête de cache HTTP (ou devrait l'être). Ceci **détermine quand le cache d'un fichier expire**. Pour résoudre ce problème, assurez-vous que votre hébergeur WordPress possède les configurations pour les en-têtes `cache-control` et `expires`. Si ce n'est pas le cas, vous verrez très probablement des avertissements sur la nécessité d'ajouter des en-têtes d'expiration ou d'utiliser la mise en cache du navigateur dans les outils de test de vitesse.

Alors que l'en-tête `cache-control` active la mise en cache côté client et définit l'âge maximum d'une ressource, l'en-tête `expires` est utilisé pour spécifier un moment précis où la ressource n'est plus valide. Bien que les deux en-têtes puissent être utilisés ensemble, vous n'avez pas nécessairement besoin d'ajouter les deux en-têtes. `cache-control` est plus récent et est généralement la méthode recommandée.

Kinsta ajoute automatiquement des en-têtes de cache HTTP sur toutes les requêtes du serveur, et si vous utilisez un CDN, il ajoute probablement ces en-têtes pour vous également.

Name	Headers	Preview	Response	Timing
<input type="checkbox"/> editwp.com				
<input type="checkbox"/> admin-bar.min.js?ver=5.0-beta3-43867				
<input type="checkbox"/> skip-link-focus-fix.js?ver=20151215				
<input type="checkbox"/> wp-embed.min.js?ver=5.0-beta3-43867				
<input checked="" type="checkbox"/> be9a4e9519964b933dea6723d480c95a?s=52&d=mm&r=g				
<input checked="" type="checkbox"/> be9a4e9519964b933dea6723d480c95a?s=128&d=mm&r=g				
<input checked="" type="checkbox"/> wp-emoji-release.min.js?ver=5.0-beta3-43867				
<input type="checkbox"/> data:application/fo...				
<input type="checkbox"/> url-metrics				
<input type="checkbox"/> tpc-check.html				
<input type="checkbox"/> postmessage.js				
<input type="checkbox"/> tpc-check-add-cookie.js				
12 requests   6.5 KB transferred   Finish: 1.19 s   DOMContentLoaded: 452 ms   ...				

▼ Response Headers

**access-control-allow-origin:** \*

**cache-control:** max-age=315360000 ←

**content-encoding:** gzip

**content-type:** application/javascript; charset=UTF-8

**date:** Wed, 07 Nov 2018 03:17:34 GMT

**etag:** W/"5bdc7194-2efa"

**expires:** Thu, 31 Dec 2037 23:55:55 GMT ←

**last-modified:** Fri, 02 Nov 2018 15:47:32 GMT

**server:** nginx

**status:** 200

**vary:** Accept-Encoding

**x-content-type-options:** nosniff

Si il manque ces en-têtes à votre serveur, vous pouvez les ajouter manuellement.

## Ajout de l'en-tête Cache-Control dans Nginx

Vous pouvez ajouter des en-têtes `cache-control` dans Nginx en ajoutant ce qui suit à l'emplacement ou au bloc serveur de votre configuration du serveur.

```
location ~* \.(js|css|png|jpg|jpeg|gif|svg|ico)$ {
    expires 30d;
    add_header Cache-Control "public, no-transform";
}
```

## Ajout d'un en-tête Expires dans Nginx

Vous pouvez ajouter des en-têtes `expires` dans Nginx en ajoutant ce qui suit à votre bloc serveur. Dans cet exemple, vous pouvez voir comment spécifier différents temps d'expiration en fonction des types de fichiers.

```
location ~* \.(jpg|jpeg|gif|png|svg)$ {
    expires 365d;
}

location ~* \.(pdf|css|html|js|swf)$ {
    expires 2d;
}
```

## Ajout de l'en-tête Cache-Control dans Apache

Vous pouvez ajouter des en-têtes `cache-control` dans Apache en ajoutant ce qui suit à votre fichier `.htaccess`. Des bouts de code peuvent être ajoutés en haut ou en bas du fichier (avant `# BEGIN WordPress` ou après `# END WordPress`).

```
<filesMatch ".(ico|pdf|flv|jpg|jpeg|png|gif|svg|js|css|swf)$">
Header set Cache-Control "max-age=84600, public"
</filesMatch>
```

# Ajout d'un en-tête Expires dans Apache

Vous pouvez ajouter les en-têtes `expires` dans Apache en ajoutant ce qui suit à votre fichier `.htaccess`.

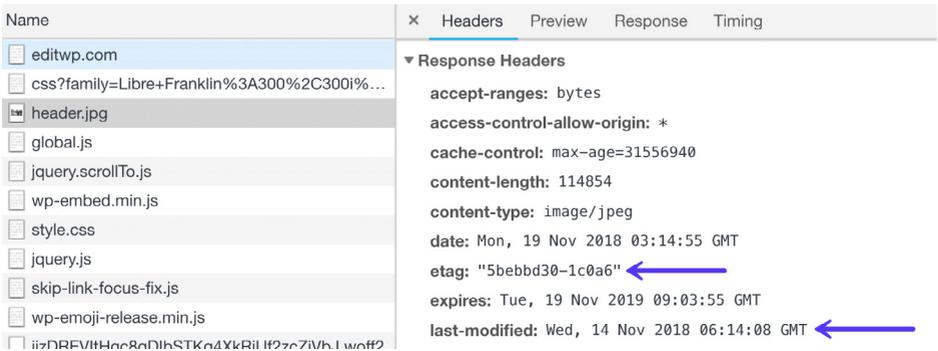
```
## EXPIRES HEADER CACHING ##
<IfModule mod_expires.c>
ExpiresActive On
ExpiresByType image/jpg "access 1 year"
ExpiresByType image/jpeg "access 1 year"
ExpiresByType image/gif "access 1 year"
ExpiresByType image/png "access 1 year"
ExpiresByType image/svg "access 1 year"
ExpiresByType text/css "access 1 month"
ExpiresByType application/pdf "access 1 month"
ExpiresByType application/javascript "access 1 month"
ExpiresByType application/x-javascript "access 1 month"
ExpiresByType application/x-shockwave-flash "access 1 month"
ExpiresByType image/x-icon "access 1 year"
ExpiresDefault "access 2 days"
</IfModule>
## EXPIRES HEADER CACHING ##
```

Il est également important de noter que vous ne pouvez ajouter des en-têtes de cache HTTP que sur les ressources de votre serveur. Si vous recevez un avertissement à ce sujet, vous avez peut-être besoin d'exploiter la mise en cache du navigateur sur une requête tierce, il n'y a rien que vous puissiez faire, car vous n'avez pas d'accès sur leur serveur. Les coupables les plus courants sont le script Google Analytics et les pixels marketing, comme Facebook et Twitter.

Si vous essayez de résoudre ce problème avec le script Google Analytics, vous pouvez l'héberger localement ou sur votre CDN (bien que cela ne soit pas officiellement supporté) avec un plugin comme Perfmatters (disponible sur [perfmatters.io](https://perfmatters.io)) ou WP Rocket (disponible sur [wp-rocket.me](https://wp-rocket.me)).

# Ajout des en-têtes Last-Modified et ETag (Validation du cache)

Ensuite, nous avons deux autres ensembles d'en-têtes, `last-modified` et `etag`. Les en-têtes `cache-control` et `expires` aident le navigateur à déterminer si le fichier a changé depuis la dernière fois qu'il a été demandé. Ou plutôt, ils valident le cache. Si ceux-ci ne sont pas correctement réglés, vous verrez peut-être un avertissement "Specify a cache validator".



Ces en-têtes doivent être inclus dans chaque réponse du serveur d'origine, car **ils valident et définissent la longueur du cache**. Si les en-têtes ne sont pas trouvés, une nouvelle requête pour la ressource sera générée à chaque fois, ce qui augmente la charge sur votre serveur. L'utilisation d'en-têtes de mise en cache garantit que les requêtes ultérieures n'ont pas à être chargées à partir du serveur, ce qui permet d'économiser de la bande passante et d'améliorer les performances de l'utilisateur.

Kinsta ajoute automatiquement les en-têtes ci-dessus sur toutes les requêtes serveur, et si vous utilisez un CDN, il ajoute probablement ces en-têtes pour vous également. Tout comme avec `cache-control` et `expires`, vous ne pouvez pas définir manuellement ces en-têtes HTTP sur des ressources externes.

## En-tête Last-Modified

L'en-tête **Last-modified** est généralement envoyé automatiquement depuis le serveur.

**C'est un en-tête que vous n'aurez généralement pas besoin d'ajouter manuellement.**

Il est envoyé pour voir si le fichier dans le cache du navigateur a été modifié depuis la dernière fois qu'il a été demandé. Vous pouvez consulter la requête d'en-tête dans Pingdom ou utiliser Chrome DevTools pour voir la valeur du dernier en-tête modifié.

## En-tête Etag

L'en-tête **Etag** est également très similaire au dernier en-tête modifié. Il sert également à valider le cache d'un fichier. Si vous utilisez Apache 2.4 ou supérieur, l'en-tête ETag est déjà automatiquement ajouté en utilisant la directive FileETag. Et en ce qui concerne NGINX, l'en-tête ETag est activé par défaut depuis 2016.

Vous pouvez activer manuellement l'en-tête ETag dans Nginx en utilisant le code suivant.

```
etag on
```

## Ajouter un en-tête Vary: Accept-Encoding

L'en-tête `vary: Accept-Encoding` doit être inclus dans chaque réponse du serveur d'origine, car il indique au navigateur si le client peut ou non gérer les versions compressées du contenu. Si ce n'est pas correctement réglé, vous verrez peut-être un message d'avertissement indiquant "Specify a Vary: Accept-Encoding Header".

Par exemple, disons que vous avez un vieux navigateur sans compression gzip et un navigateur moderne avec. Si vous n'utilisez pas l'en-tête `vary: Accept-Encoding` votre serveur Web ou CDN peut mettre en cache la version non compressée et la transmettre par erreur au navigateur moderne, ce qui nuit à la performance de votre site WordPress. En utilisant l'en-tête, vous pouvez vous assurer que votre serveur web et/ou CDN fournit la version appropriée.

Name	× Headers	Preview	Response	Timing
editwp.com	▼ Response Headers			
css?family=Libre+Franklin%3A300%2C300i%...	<b>content-encoding:</b> gzip			
header.jpg	<b>content-type:</b> text/html; charset=UTF-8			
global.js	<b>date:</b> Mon, 19 Nov 2018 03:14:57 GMT			
jquery.scrollTo.js	<b>server:</b> nginx			
wp-embed.min.js	<b>status:</b> 200			
style.css	<b>vary:</b> Accept-Encoding ←			
jquery.js	<b>x-content-type-options:</b> nosniff			
skip-link-focus-fix.js	<b>x-kinsta-cache:</b> HIT			

Kinsta ajoute automatiquement les en-têtes ci-dessus sur toutes les requêtes serveur, et si vous utilisez un CDN, il ajoute probablement ces en-têtes pour vous également. Tout comme pour les autres en-têtes de cache dont nous avons parlé plus haut, vous ne pouvez pas définir manuellement cet en-tête sur des ressources externes.

## Ajouter l'en-tête vary: Accept-Encoding dans Apache

Vous pouvez ajouter l'en-tête `vary: Accept-Encoding` dans Apache en ajoutant ce qui suit à votre fichier `.htaccess`.

```
<IfModule mod_headers.c>
  <FilesMatch ".(js|css|xml|gz|html)$">
    Header append Vary: Accept-Encoding
  </FilesMatch>
</IfModule>
```

## Ajouter l'en-tête Vary: Accept-Encoding dans Nginx

Vous pouvez ajouter l'en-tête `vary: Accept-Encoding` dans Nginx en ajoutant le code suivant à votre fichier de configuration. Tous les fichiers de configuration Nginx se trouvent dans le répertoire `/etc/nginx/`. Le fichier de configuration principal est `/etc/nginx/nginx.conf`.

```
gzip_vary on
```

## Modifier la limite de mémoire de WordPress dans wp-config.php

Comme indiqué dans le Codex WordPress, avec WordPress Version 2.5, l'option `WP_MEMORY_LIMIT` vous permet de spécifier la quantité maximale de mémoire qui peut être utilisée par PHP. Ce réglage peut être nécessaire dans le cas où vous recevez un message tel que "Allowed memory size of xxxxxx bytes exhausted".

Par défaut, WordPress tentera d'augmenter la mémoire allouée par PHP à 40 Mo pour un site unique et 64 Mo pour le multisite. Ils définissent les limites de mémoire dans le fichier (source).

Vous avez alors aussi le `memory_limit` de PHP sur le serveur par votre hébergeur. Ce sont deux choses différentes. Chez Kinsta, nous fixons la `memory_limit` par défaut à 256M. Si vous rencontrez l'erreur d'épuisement de la mémoire, vous pouvez essayer d'augmenter la limite de mémoire PHP dans WordPress.

Ajoutez ce qui suit à votre fichier `wp-config.php` file , juste avant la ligne qui dit "That's all, stop editing! Happy blogging".

```
define( 'WP_MEMORY_LIMIT', '256M' );
```

Jan Reilink a également un excellent article de blog sur son site qui décrit plus en détails la question de la limite de mémoire de WordPress. Il donne également une variante du code que vous pourriez utiliser. Au lieu de régler la limite manuellement, vous pouvez la mettre à la valeur `memory_limit` de PHP.

```
define( 'WP_MEMORY_LIMIT', ini_get( 'memory_limit' ) );
```

CHAPITRE 15:

# Conseils sur l'optimisation du Front-End



**Nous allons maintenant nous pencher sur les moyens d'accélérer WordPress en optimisant le front-end. Le front-end implique généralement tout ce qui est entièrement géré par le navigateur côté client, tel que le CSS, le JavaScript, les images, etc. Cela inclut également l'analyse des services externes que vous avez chargés sur votre site et l'impact qu'ils ont sur votre temps de chargement global.**

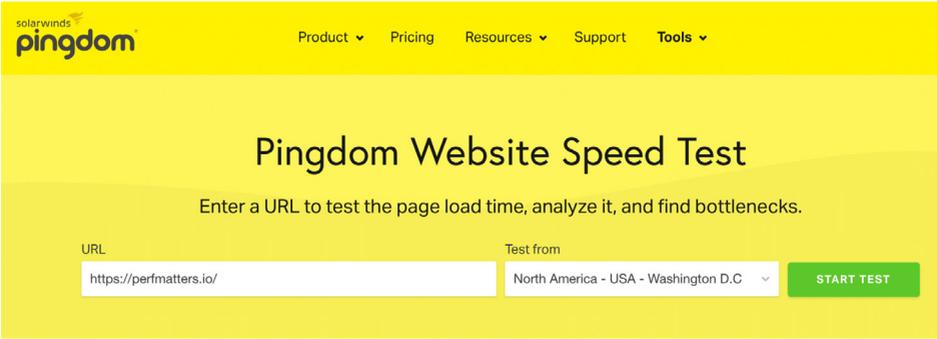
Deux des objectifs les plus importants que vous devriez avoir quand il s'agit d'optimisation du front-end sont :

- **Réduire la taille globale de votre page Web.** La taille de votre CSS, JavaScript, des images, est importante. Un site Web de 4 Mo va généralement se charger beaucoup plus lentement qu'un site Web de 1 Mo. Cependant, Paul Calvano a un excellent article sur l'impact du poids de la page sur le temps de chargement et à quel point il est important de s'assurer que ce n'est pas la seule chose que vous suivez, car cela peut parfois être trompeur.
- **Réduire les requêtes HTTP et les services externes.** Avec HTTP/2, plusieurs requêtes et réponses peuvent maintenant être envoyées en même temps en utilisant une seule connexion TCP. Bien que ce soit un excellent moyen

d'améliorer les performances, la réduction des requêtes HTTP peut toujours aider à accélérer votre site WordPress. Cela comprend également la réduction du nombre total de requêtes et de services externes. Chacune d'entre elles ajoute des délais supplémentaires tels que les recherches DNS, les connexions TLS et la latence du réseau.

## Tester la vitesse de votre site WordPress pour obtenir une base de référence

Quand il s'agit d'optimiser le front-end de votre site, il est toujours bon de commencer par un point de référence. Cela signifie généralement que vous devez effectuer un test de vitesse. Il y a une multitude de façons de le faire. Nous avons des guides détaillés sur la façon d'utiliser correctement Pingdom et GTmetrix disponibles sur notre site Web sur [kinsta.com](https://www.kinsta.com).



### Your Results:

[DOWNLOAD HAR](#)[SHARE RESULT](#)

Performance grade

**B 88**

Page size

**243.0 KB**

Load time

**398 ms**

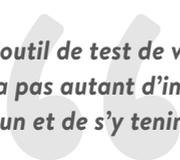
Requests

**36**

Voici quelques points à garder à l'esprit lors des tests de vitesse :

# 1. Choisir un outil et s'y tenir

Nous sommes de grands fans de Pingdom, GTmetrix, WebPageTest, PageSpeed Insights, et Chrome DevTools. Cependant, ce n'est pas tant l'outil de test de vitesse que vous utilisez qui importe, c'est votre cohérence. Ils ont tous différents moyens de mesurer et de quantifier la vitesse, alors choisissez un outil et suivez-le tout au long de vos tests et de vos optimisations. Même Google dit d'en choisir un.



*L'outil de test de vitesse que vous choisissez  
n'a pas autant d'importance que d'en choisir  
un et de s'y tenir pendant tous vos tests.*

# 2. Ne soyez pas obsédés par un score parfait

Beaucoup d'outils tels que Google PageSpeed Insights ont tous un certain type de score de vitesse ou de performances. Il est important de se rappeler que le score n'a pas toujours autant d'importance que la vitesse de votre site Web et la performance perçue par l'utilisateur. Le score est là pour vous aider à évaluer vos résultats. Mais l'obsession d'une note parfaite de 100/100 ou d'un A dans certains cas pourrait être une perte de temps. Et les sites plus grands avec beaucoup de scripts et de publicités externes n'obtiendront jamais un score parfait, ce qui est tout-à-fait correct.

# 3. L'emplacement de votre test est important

L'endroit que vous choisissez pour le test de vitesse est important. Comme nous l'avons mentionné dans une section précédente, la raison en est que tout cela est relatif à l'emplacement du centre de données que vous avez choisi. Le TTFB, la latence du réseau, tout entre en jeu. Testez donc votre site à la fois à partir d'un endroit proche de votre centre de données et d'un endroit éloigné. Cela vous aidera également à voir l'impact qu'un CDN peut avoir sur votre site WordPress.

## 4. Tester plusieurs fois à cause de la mise en cache

Comme nous l'avons vu plus haut dans la section sur la mise en cache, si le cache a été récemment effacé ou est expiré sur votre hébergeur WordPress ou CDN, il va enregistrer un "MISS" sur l'en-tête HTTP. Cela signifie que votre site Web ou votre ressource n'est pas servie à partir du cache.

FILE	SIZE	0.0s	0.1s	0.2s
 <a href="https://perfmatters.io/">https://perfmatters.io/</a>	9.6 KB			
<a href="https://perfmatters.io/">https://perfmatters.io/</a>				
Response Headers <span style="background-color: #c8e6c9; border-radius: 3px; padding: 2px;">200</span>				
<b>status</b>	200			
<b>x-content-type-options</b>	nosniff			
<b>content-encoding</b>	gzip			
<b>set-cookie</b>	PHPSESSID=d33c7f3bfa8f6c0a9ba55f3a6804e1...			
<b>x-kinsta-cache</b>	MISS 			
<b>expires</b>	Thu, 19 Nov 1981 08:52:00 GMT			

Pour voir correctement la vitesse de l'ensemble de votre site, vous avez besoin de voir que tout à été chargé depuis le cache, votre page initiale, et toutes les ressources doivent afficher un "HIT". Cela nécessite parfois l'exécution de votre test de vitesse plusieurs fois. Vous pouvez alors prendre la moyenne.

FILE	SIZE	0.0s	0.1s	0.2s
 <a href="https://perfmatters.io/">https://perfmatters.io/</a>	9.6 KB			
<a href="https://perfmatters.io/">https://perfmatters.io/</a>				
Response Headers <span style="background-color: #c8e6c9; border-radius: 3px; padding: 2px;">200</span>				
<b>status</b>	200			
<b>x-content-type-options</b>	nosniff			
<b>content-encoding</b>	gzip			
<b>set-cookie</b>	PHPSESSID=d33c7f3bfa8f6c0a9ba55f3a6804e1...			
<b>x-kinsta-cache</b>	HIT 			
<b>expires</b>	Thu, 19 Nov 1981 08:52:00 GMT			

Passons maintenant à quelques optimisations du front-end que vous pouvez faire sur votre site WordPress.

## Supprimer les chaînes de requête (Query Strings)

Un avertissement ou une recommandation courante que les gens voient dans les outils de test de vitesse est que vous devriez supprimer les chaînes de requête (Query Strings). De quoi s'agit-il ? Eh bien, fondamentalement, vos fichiers CSS et JavaScript ont généralement la version du fichier à la fin de leurs URLs, comme `https://domaine.com/fichier.min.css?ver=4.5.3`. Certains serveurs et serveurs proxy sont incapables de mettre en cache les chaînes de requête. Ainsi, en les supprimant, vous pouvez parfois améliorer votre mise en cache.

Vous pouvez les supprimer manuellement en plaçant le code suivant dans le fichier `functions.php` de votre thème.

```
function remove_query_strings() {
    if(!is_admin()) {
        add_filter('script_loader_src', 'remove_query_strings_split', 15);
        add_filter('style_loader_src', 'remove_query_strings_split', 15);
    }
}

function remove_query_strings_split($src){
    $output = preg_split("/(&ver|\?ver)/", $src);
    return $output[0];
}
add_action('init', 'remove_query_strings');
```

Ou vous pouvez utiliser un plugin premium comme Perfmatters (disponible sur [perfmatters.io](https://perfmatters.io)) pour supprimer les chaînes de requête en un seul clic.

## Avec des chaînes de requête

Voici un exemple de chargement de scripts avec des chaînes de requête.

Name	Status
 editwp.com	200
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600%2C600i...	200
 style.css?ver=4.9.8 ←	200
 jquery.js?ver=1.12.4	200
 header.jpg	200
 skip-link-focus-fix.js?ver=1.0	200
 global.js?ver=1.0	200
 jquery.scrollTo.js?ver=2.1.2	200
 jizDREVIthGc8qDlbSTKq4XkRiUf2zcZiVbJ.woff2	200

## Sans chaînes de requête (Après le code)

Name	Status
 editwp.com	200
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600%2C600i...	200
 style.css ←	200
 jquery.js	200
 header.jpg	200
 skip-link-focus-fix.js	200
 global.js	200
 jquery.scrollTo.js	200
 jizDREVIthGc8qDlbSTKq4XkRiUf2zcZiVbJ.woff2	200

Voici un exemple de scripts après avoir supprimé des chaînes de requête.

Cependant, avant d'effacer immédiatement les chaînes de requête sur votre site, il est important de savoir pourquoi les chaînes de requête sont utilisées. Le versioning sur fichiers est généralement utilisé par les développeurs WordPress pour contourner les problèmes de cache.

Par exemple, si un développeur de plugin pousse une mise à jour et change le fichier `style.css` de `?ver=4.6` à `?ver=4.7`, ce sera traité comme une URL entièrement nouvelle et ne sera pas mis en cache. Si vous supprimez les chaînes de requête et mettez à jour un plugin, la version mise en cache pourrait continuer à être servie. Dans certains cas, cela pourrait casser l'apparence de votre site jusqu'à ce que la ressource en cache expire ou que le cache soit complètement vidé.

De plus, certains CDNs peuvent mettre en cache des chaînes de requête. Le CDN Kinsta peut le faire et le fait par défaut. Ainsi, si vous êtes un client Kinsta, les chaînes de requête sont déjà mises en cache sur vos ressources.

## Éliminer le JavaScript et le CSS bloquant le rendu

Un avertissement sur le blocage du rendu par le JavaScript et CSS peut apparaître lorsque vous avez des fichiers empêchant le chargement de la page aussi rapidement que possible. Les JS et CSS spécifiques sont parfois conditionnels, ce qui signifie qu'ils ne sont pas tenus d'afficher un contenu au-dessus de la ligne de flottaison. Vous pouvez les empêcher de devenir des bloqueurs de rendu en utilisant les attributs `async` et `defer`.

Opportunity	Estimated Savings
1 Eliminate render-blocking resources	 2.24 s ^
Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. <a href="#">Learn more.</a>	

Pour éliminer le JavaScript et CSS bloquant le rendu, vous devez faire ce qui suit :

## Effacer le JS du chemin de rendu critique

Le déplacement de JavaScript hors du chemin de rendu critique se fait généralement en ajoutant l'attribut `defer` ou `async` aux éléments HTML script qui appellent les ressources JavaScript.

- **L'attribut `async`** indique au navigateur de commencer à télécharger la ressource immédiatement sans ralentir l'analyse HTML. Une fois que la ressource est disponible, l'analyse HTML est mise en pause pour que la ressource puisse être chargée.
- **L'attribut `defer`** indique au navigateur de ne pas télécharger la ressource jusqu'à ce que l'analyse HTML soit terminée. Une fois que le navigateur a terminé avec le HTML, il téléchargera et affichera tous les scripts différés

dans l'ordre dans lequel ils apparaissent dans le document.

## Optimiser la livraison des ressources CSS

L'optimisation de la livraison du CSS signifie essentiellement que vous devez trouver un moyen de ne pas bloquer le rendu à cause du CSS.

- Identifiez les styles qui sont nécessaires pour afficher le contenu au-dessus de la ligne de flottaison et mettez ces styles en inline avec le HTML.
- Utilisez le CSS conditionnel sur les appareils uniquement si nécessaire.
- Chargez le CSS restant de manière asynchrone.

Faire tout ce qui précède peut parfois être un processus délicat et nécessite certainement quelques ajustements en fonction des scripts que vous avez chargés sur votre site. Voici quelques plugins WordPress qui peuvent vous aider :

- Autoptimize (Gratuit, disponible sur WordPress.org)
- Async JavaScript (Gratuit, disponible sur WordPress.org)
- Hummingbird (Gratuit, disponible sur WordPress.org)

Pour une explication plus détaillée, nous vous recommandons de consulter notre article sur l'élimination du JavaScript et du CSS bloquant le rendu.

## Combiner les CSS et JavaScript externes dans WordPress

Le message d'avertissement "combine external CSS" est généralement affiché lorsque vous utilisez un CDN parce que vous hébergez vos fichiers CSS sur un domaine externe, tel que **cdn.domaine.com**. Dans le passé, un moyen rapide de résoudre ce problème était de concaténer vos fichiers CSS, ou de les combiner pour qu'ils se chargent en une seule requête.

Toutefois, si vous utilisez HTTPS avec un fournisseur qui prend en charge HTTP/2, cet

avertissement n'est plus aussi pertinent qu'avant. Avec HTTP/2, plusieurs fichiers CSS peuvent maintenant être chargés en parallèle sur une seule connexion. Et plus de 86% des navigateurs supportent HTTP/2.

Mais cela ne signifie pas nécessairement que cette optimisation est complètement morte. Dans certains cas, nous avons vu que cela accélérerait encore les sites WordPress. Cela dépend de la taille des fichiers et de leur nombre. Par conséquent, c'est une optimisation que nous vous recommandons de tester sur votre site.

L'une des façons les plus simples de combiner vos fichiers CSS et JavaScript externes est d'utiliser le plugin gratuit Autoptimize (disponible sur WordPress.org). Après les avoir combinés, vous verrez un fichier "autoptimize\_xxxxxxx.css" ou "autoptimize\_xxxxxxx.js". Il prend également en charge leur chargement à partir de votre CDN. Vous pouvez également le faire avec le plugin WP Rocket.

Name	Status	Domain
 editwp.com	200	editwp.com
 autoptimize_23d90d8a1eaacef4363d9e02c50ae363.css ←	200	mk0editwpn6...
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600...	200	fonts.googlea...
 jquery.js?ver=1.12.4	200	mk0editwpn6...
 header.jpg	200	mk0editwpn6...
 autoptimize_3fde0f2c0341cdc5afd7cc0c0e403f92.js ←	200	mk0editwpn6...
 jizDREVIthGc8qDlbSTKq4XkRiUf2zcZiVbJ.woff2	200	fonts.gstatic.c...
 jizAREVIthGc8qDlbSTKq4XkRi24_Sl0q1vjitOh.woff2	200	fonts.gstatic.c...
 jizAREVIthGc8qDlbSTKq4XkRi20-Sl0q1vjitOh.woff2	200	fonts.gstatic.c...

## Utilisation de la minification sur le HTML, CSS et JavaScript

Nous pouvons réduire la quantité de données que le navigateur doit télécharger en minifiant les ressources HTML, CSS et JavaScript. La minification est le processus qui consiste à supprimer les caractères inutiles comme les commentaires et les espaces du code source. Ces caractères sont extrêmement utiles dans le développement, mais ils sont inutiles pour l'affichage de la page dans le navigateur.

## HTML non minifié

Voici un exemple de code HTML non minifié.

```
1 <!DOCTYPE html>
2 <html lang="en-US" class="no-js no-svg">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link rel="profile" href="http://gmpg.org/xfn/11">
7
8 <script>(function(html){html.className = html.className.replace(/\bno-
9 js\b/, 'js')})(document.documentElement);</script>
10 <title>editwp &#8211; Just another WordPress site</title>
11 <link rel='dns-prefetch' href='//fonts.googleapis.com' />
```

## HTML minifié

Voici un exemple de code HTML minifié.

```
1 <!DOCTYPE html><html lang="en-US" class="no-js no-svg"><head><meta
2 charset="UTF-8"><meta name="viewport" content="width=device-width,
3 initial-scale=1"><link rel="profile" href="http://gmpg.org/xfn/11">
4 <script>(function(html){html.className = html.className.replace(/\bno-
5 js\b/, 'js')})(document.documentElement);</script> <link type="text/css"
6 media="all" href="https://mk0editwpn6qbfwru5o9.kinstacdn.com/wp-
7 content/cache/autoptimze/css/autoptimze_499850c34eb7b4e7cbd9d8b41b2a5
8 7d6.css" rel="stylesheet" /><style type="text/css"
9 media="print">#wpadminbar{display:none}</style><style type="text/css"
10 media="screen">html{margin-top:32px !important}* html body{margin-
```

Vous pouvez utiliser le plugin gratuit Autoptimize ou WP Rocket pour minifier facilement vos fichiers.

## Utiliser des domaines sans cookies

En général, lorsque vous diffusez du contenu tel que des images, du JavaScript, CSS, il n'y a aucune raison pour qu'un cookie HTTP l'accompagne, car il crée des frais supplémentaires. Une fois que le serveur définit un cookie pour un domaine particulier, toutes les requêtes HTTP ultérieures pour ce domaine doivent inclure le cookie. Cet avertissement est généralement affiché sur les sites ayant un grand nombre de requêtes.

Souvent, vous pouvez ignorer cet avertissement car les nouveaux protocoles tels que HTTP/2 rendent cela moins important. Le coût d'une nouvelle connexion est généralement plus élevé que celui de la diffusion en continu sur la même connexion.

Une façon simple de corriger cet avertissement est d'utiliser un fournisseur de CDN qui peut ignorer les cookies et supprimer les cookies, ce qui empêchera complètement le client de recevoir l'en-tête de réponse Set-Cookie. KeyCDN est l'un des fournisseurs CDN qui offre cette fonctionnalité. Par défaut, vous pouvez voir que les deux options suivantes sont activées. C'est une alternative facile sans avoir à déplacer et configurer votre site pour fournir des ressources statiques à partir d'un sous-domaine séparé.

#### Cache Cookies \*

By default, files with cookies are not cacheable. However, enabling this option will ignore the presence of cookies and therefore force the edge servers to cache these files.

#### Strip Cookies \*

This feature strips the cookies received from the origin server. The client will not receive the [Set-Cookie](#) response header. **It is highly recommended to enable this directive if you enable Cache Cookies.**

Si vous utilisez Cloudflare, vous ne pouvez pas désactiver les cookies sur les ressources diffusées par leur réseau. CloudFlare inclut son propre cookie de sécurité dans votre en-tête. Encore une fois, ces cookies sont très petits et les répercussions sur l'affichage sont extrêmement minimes. Mais si vous utilisez CloudFlare, il n'y a aucun moyen de contourner cet avertissement.

Une deuxième façon de contourner ce problème est de reconfigurer votre site WordPress pour fournir les ressources statiques d'un nouveau domaine ou sous-domaine.

## Désactiver les Embeds dans WordPress

Quand ils ont sorti WordPress 4.4, ils ont fusionné la fonction oEmbed dans le coeur. Cela permet aux utilisateurs d'intégrer des vidéos YouTube, des tweets

et de nombreuses autres ressources sur leurs sites simplement en collant une URL, que WordPress convertit automatiquement en embed et fournit un aperçu en direct dans l'éditeur visuel. Avec la mise à jour, WordPress lui-même est devenu un fournisseur oEmbed.

Cette fonction est utile pour beaucoup de gens, et vous voudrez peut-être la garder activée. Cependant, cela signifie qu'elle génère également une requête HTTP supplémentaire sur votre site WordPress pour charger le fichier `wp-embed.min.js`. Et cela se charge à l'échelle du site. Bien que ce fichier ne fasse que 1,7 Ko, les choses comme celles-ci s'accroissent avec le temps. La requête elle-même est parfois plus importante que

la taille du  
téléchargement  
de contenu.

Name	Status	Domain
 editwp.com	200	editwp.com
 global.js	200	mk0editwpn...
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...	200	fonts.google...
 jquery.scrollTo.js	200	mk0editwpn...
 wp-embed.min.js ←	200	mk0editwpn...
 style.css	200	mk0editwpn...
 jquery.js	200	mk0editwpn...
 header.jpg	200	mk0editwpn...
 skip-link-focus-fix.js	200	mk0editwpn...
 wp-emoji-release.min.js	200	editwp.com

Il y a plusieurs façons de désactiver les Embeds dans WordPress. Vous pouvez le faire avec les plugins suivants :

- Disable Embeds (Gratuit, disponible sur WordPress.org)
- Perfmatters (Disponible sur perfmatters.io)

Ou vous pouvez aussi désactiver les Embeds avec du code. En raison de sa longueur, vous pouvez le trouver sur notre site Web à l'adresse suivante :  
<https://kinsta.com/fr/base-de-connaissances/desactiver-embeds-wordpress/>

## Désactiver les Emojis dans WordPress

Semblable aux Embeds, dans WordPress 4.2, ils ont ajouté le support des emojis dans le noyau pour les navigateurs plus anciens. Le gros problème est que cela génère une

requête HTTP supplémentaire sur votre site WordPress pour charger le fichier `wp-emoji-release.min.js`. Et cela se charge à l'échelle du site. Bien que ce fichier

ne fasse que 10,5

Ko, il est inutile si vous n'utilisez pas d'emojis sur votre site.

Name	Status	Domain
 editwp.com	200	editwp.com
 global.js	200	mk0editwpr...
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...	200	fonts.google...
 jquery.scrollTo.js	200	mk0editwpr...
 wp-embed.min.js	200	mk0editwpr...
 style.css	200	mk0editwpr...
 jquery.js	200	mk0editwpr...
 header.jpg	200	mk0editwpr...
 skip-link-focus-fix.js	200	mk0editwpr...
 wp-emoji-release.min.js ←	200	editwp.com
 jizDREvItHgc8qDibSTKq4XkRIUf2zcZIVbJ.woff2	200	fonts.gstatic...

Il y a plusieurs façons de désactiver les Emojis dans WordPress. Vous pouvez le faire avec les plugins suivants :

- Disable Emojis (Gratuit, disponible sur WordPress.org)
- Perfmatters (Disponible sur perfmatters.io)

Ou vous pouvez aussi désactiver les Emojis avec du code. En raison de sa longueur, vous pouvez le trouver sur notre site Web à l'adresse suivante :

<https://kinsta.com/fr/base-de-connaissances/desactiver-emojis-wordpress/>

## Comment accélérer ou désactiver les commentaires WordPress

Une section de commentaires occupée sur un site peut causer beaucoup de problèmes de performances. Pensez simplement aux ressources qui sont nécessaires pour que les commentaires fonctionnent :

- Une base de données est interrogée pour extraire les commentaires existants.
- Des entrées de base de données sont créées pour chaque nouveau commentaire.

- Les commentaires et les métadonnées de commentaires sont reçus et traités par le navigateur du visiteur.
- Les ressources externes, telles que les Gravatars, sont demandées, téléchargées et chargées (nécessitant une recherche DNS séparée).
- Dans de nombreux cas, de grandes ressources JavaScript et jQuery doivent être téléchargées et traitées pour que le système de commentaires fonctionne comme il se doit.

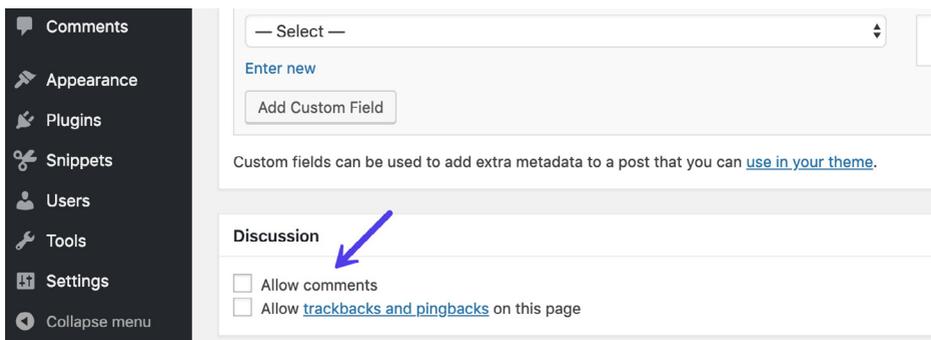
Voici quatre options différentes que vous pouvez choisir pour accélérer les commentaires WordPress :

## Option 1 – Désactiver les commentaires

Si votre site ne reçoit pas beaucoup de commentaires et que vous pensez qu'ils n'ont aucune valeur ajoutée, il serait peut-être préférable de désactiver complètement les commentaires. Rappelez-vous que les commentaires peuvent avoir un impact sur votre référencement, car Google va les parcourir en tant que contenu supplémentaire sur la page, vous ne devriez donc approuver que les commentaires de haute qualité. Jetez un coup d'œil à ces trois façons simples de désactiver les commentaires :

### 1. Désactiver les commentaires dans les options de WordPress

Vous pouvez désactiver individuellement les commentaires d'une page ou d'un article



The screenshot shows the WordPress Customizer interface. On the left is a dark sidebar with a menu containing 'Comments', 'Appearance', 'Plugins', 'Snippets', 'Users', 'Tools', 'Settings', and 'Collapse menu'. The main area is light gray and contains a dropdown menu with '— Select —' and a 'Custom Fields' section with an 'Add Custom Field' button. Below that is a 'Discussion' section with two checkboxes: 'Allow comments' (checked) and 'Allow trackbacks and pingbacks on this page' (unchecked). A blue arrow points to the 'Allow comments' checkbox.

en éditant et en décochant l'option "Autoriser les commentaires" dans la section "Discussion". Si vous ne voyez pas cette section, vous devrez peut-être la faire défiler vers le haut et l'activer dans "Screen Options".

Vous pouvez également désactiver globalement les discussions. Vous pouvez le faire en cliquant sur "Discussion" sous Paramètres dans votre tableau de bord WordPress, puis en décochant l'option "Autoriser les lecteurs à publier des commentaires sur les nouveaux articles". L'inconvénient de cette méthode est qu'elle n'affecte pas les anciens articles, mais seulement les nouveaux.

Si vous avez besoin de plus de contrôle et que vous voulez simplement désactiver globalement tous les commentaires en même temps, nous vous recommandons d'utiliser l'une des méthodes suivantes ci-dessous.

The screenshot shows the WordPress 'Discussion Settings' page. On the left is a dark sidebar with navigation links: Dashboard, Kinsta Cache, Posts, Media, Pages, Comments, Appearance, Plugins, Snippets, Users, Tools, Settings (highlighted), General, Writing, Reading, Discussion, and Media. The main content area is titled 'Discussion Settings' and has a 'Help' dropdown in the top right. Under 'Default article settings', there are three checkboxes: 'Attempt to notify any blogs linked to from the article' (checked), 'Allow link notifications from other blogs (pingbacks and trackbacks) on new articles' (checked), and 'Allow people to post comments on new articles' (unchecked, with a blue arrow pointing to it). A note below states '(These settings may be overridden for individual articles.)'. Under 'Other comment settings', there are several options: 'Comment author must fill out name and email' (checked), 'Users must be registered and logged in to comment' (unchecked), 'Automatically close comments on articles older than 14 days' (unchecked), 'Show comments cookies opt-in checkbox.' (unchecked), 'Enable threaded (nested) comments 5 levels deep' (checked), 'Break comments into pages with 50 top level comments per page and the last page displayed by default' (unchecked), and 'Comments should be displayed with the older comments at the top of each page' (selected).

## 2. Désactiver les commentaires avec un Plugin

Une des façons les plus simples de désactiver les commentaires est d'utiliser un plugin gratuit comme Disable Comments (disponible sur WordPress.org). Vous pouvez

désactiver les commentaires partout d'un seul clic, ou les désactiver sur les articles, les pages ou les types de posts des médias.

### 3. Désactiver les commentaires avec du Code

Vous pouvez également désactiver les commentaires avec du code en supprimant simplement la section commentaire de votre thème WordPress avec de simples commentaires HTML.

Dans votre fichier `page.php` , remplacez simplement le code suivant :

```
<?php comments_template( '', true ); ?>
```

Par ce code :

```
<!-- Begin Comment  
<?php comments_template( '', true ); ?>  
End Comment -->
```

Cela le supprime de vos pages. Et pour le supprimer de votre blog, naviguez jusqu'à votre fichier `single.php` et remplacez le code suivant :

```
<?php comments_template(); ?>
```

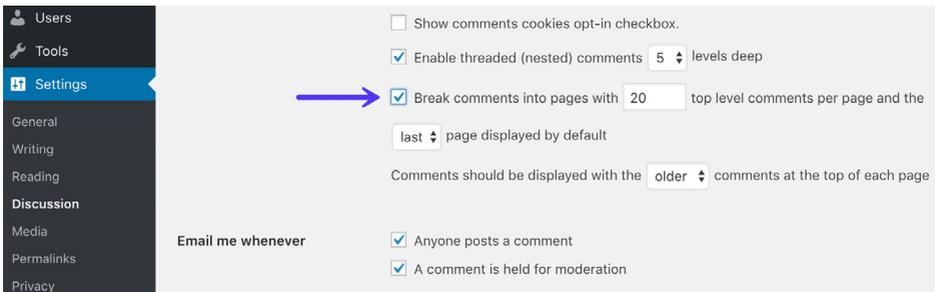
Par ce code :

```
<!-- Begin Comment  
<?php comments_template(); ?>  
End Comment -->
```

## Option 2 – Optimiser le système de commentaires natif de WordPress

Votre deuxième option serait d'optimiser le système de commentaires WordPress natif. Une façon serait de réduire le nombre de commentaires chargés avec le chargement initial de la page.

1. Allez dans Paramètres / Discussion dans la zone d'administration de WordPress.
2. Recherchez la section Autres réglages des commentaires.
3. Cochez la case à côté de Diviser les commentaires en pages avec et ajoutez une valeur pour le nombre de commentaires que vous voulez afficher au chargement initial de la page.



Une autre option que vous avez est d'utiliser des Gravatars hébergés sur votre CDN. C'est l'approche que nous adoptons chez Kinsta.

Par défaut, lorsque les commentaires WordPress sont chargés, chaque Gravatar unique nécessite une requête HTTP. Ainsi, si une page est chargée avec les commentaires de 50 commentateurs différents, 50 requêtes HTTP seront nécessaires pour télécharger tous ces Gravatars. Comme vous pouvez l'imaginer, cela peut avoir un impact sur la vitesse de votre page. Sans parler du fait que nous avons vu que la recherche DNS externe sur gravatar.com peut être parfois lente et dans certains cas même indisponible.



Assurez-vous de toujours tester la vitesse du système de commentaires tiers que vous essayez. Jetez un coup d'oeil à toutes les requêtes séparées que Disqus génère (comme montré ci-dessus). Bien que la plupart de ces requêtes se chargent de manière asynchrone, vous remarquerez toujours un temps de chargement supplémentaire si vous utilisez Disqus.

## Option 4 – Commentaires en Lazy Load

Votre quatrième option est de charger les commentaires en Lazy Load pour qu'ils ne ralentissent pas l'affichage initial de la page. Voici quelques plugins que vous voudrez peut-être consulter :

- **Lazy Load for Comments** (Gratuit, disponible sur WordPress.org): Ce plugin vous permet de charger les commentaires WordPress natifs avec le Lazy Load.
- **Disqus Conditional Load** (Gratuit, disponible sur WordPress.org): Si vous voulez utiliser le système de commentaires Disqus, il s'agit d'un plugin indispensable pour charger les commentaires avec le Lazy Load.

## Désactiver les flux RSS WordPress

Si vous n'utilisez pas la partie blogging de WordPress sur votre site, vous pouvez désactiver les flux RSS WordPress. Bien que cela n'aura pas un impact énorme sur la performance, tout aide. C'est aussi une chose de moins dont vous devrez vous inquiéter.

Découvrez ces deux façons différentes de désactiver les flux RSS dans WordPress :

### 1. Désactiver le flux RSS avec un Plugin

La première façon de désactiver un flux RSS WordPress est d'utiliser un plugin gratuit comme Disable Feeds (disponible sur WordPress.org). Ce plugin désactive tous les flux RSS/Atom sur votre site WordPress en redirigeant toutes les requêtes.

Vous pouvez également utiliser un plugin premium comme Perfmatters (développé par un membre de l'équipe de Kinsta, disponible sur [perfmatters.io](https://perfmatters.io)), qui vous permet de désactiver les flux RSS et les liens de flux RSS, ainsi que d'autres optimisations pour votre site WordPress.

## 2. Désactiver le flux RSS avec du Code

Si vous ne voulez pas utiliser un plugin, vous pouvez aussi désactiver les flux RSS avec du code. Déposez le code suivant dans le fichier `functions.php` de votre thème.

```
function itsme_disable_feed() {
    wp_die( __( 'No feed available, please visit the <a href="'. esc_url(
    home_url( '/' ) ) .'">homepage</a>!' ) );
}

add_action( 'do_feed', 'itsme_disable_feed', 1 );
add_action( 'do_feed_rdf', 'itsme_disable_feed', 1 );
add_action( 'do_feed_rss', 'itsme_disable_feed', 1 );
add_action( 'do_feed_rss2', 'itsme_disable_feed', 1 );
add_action( 'do_feed_atom', 'itsme_disable_feed', 1 );
add_action( 'do_feed_rss2_comments', 'itsme_disable_feed', 1 );
add_action( 'do_feed_atom_comments', 'itsme_disable_feed', 1 );
```

WordPress génère également des liens vers les flux RSS dans l'en-tête de votre page Web. Vous pouvez aller plus loin et supprimer ces liens à l'intérieur du code HTML de vos pages. Déposez le code suivant dans le fichier `functions.php` de votre thème.

```
remove_action( 'wp_head', 'feed_links_extra', 3 );
remove_action( 'wp_head', 'feed_links', 2 );
```

## Utiliser Prefetch et Preconnect

Les astuces et directives de ressources telles que prefetch et preconnect peuvent être un excellent moyen d'accélérer WordPress en arrière plan.

# Prefetch

Le Prefetch DNS vous permet de résoudre les noms de domaine (effectuer une recherche DNS en arrière-plan) avant qu'un utilisateur ne clique sur un lien, ce qui à son tour peut aider à améliorer les performances. Cela se fait en ajoutant une balise `rel="dns-prefetch"` dans l'en-tête de votre site WordPress.

```
<link rel="dns-prefetch" href="//domaine.com">
```

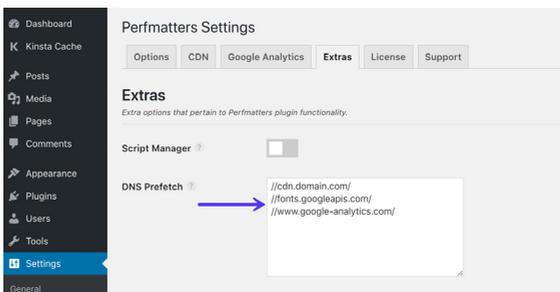
Certaines choses courantes pour lesquelles utiliser le Prefetch DNS est votre URL CDN, les polices Google, Google Analytics, etc.

```
<link rel="dns-prefetch" href="//cdn.domaine.com/">
<link rel="dns-prefetch" href="//fonts.googleapis.com/">
<link rel="dns-prefetch" href="//www.google-analytics.com">
```

Prefetch est également supporté par la plupart des navigateurs modernes. Il y a plusieurs façons d'implémenter cela.

1. Vous pouvez ajouter le code à votre en-tête en utilisant un plugin gratuit comme *Head, Footer and Post Injections* (disponible sur WordPress.org).
2. Vous pouvez ajouter le code manuellement dans le fichier `header.php` de votre thème.
3. Implémenter le Prefetch DNS en utilisant un plugin premium comme *Perfmatters* (Disponible sur [perfmatters.io](https://perfmatters.io)). Cliquez simplement sur l'onglet "Extras" dans le plugin *Perfmatters* et ajoutez des domaines.

Format : `//domaine.tld`  
(un par ligne)



## Preconnect

Preconnect permet au navigateur d'établir des connexions précoces avant une requête HTTP, ce qui élimine la latence aller-retour et fait gagner du temps aux utilisateurs.

Preconnect est un outil important dans votre boîte à outils d'optimisation... il peut éliminer de nombreux allers et retours coûteux de votre chemin de requête – dans certains cas, réduisant la latence de requête par centaines, voire par milliers de millisecondes.



Iya Grigorik

Cela se fait en ajoutant une balise `rel="preconnect"` dans l'en-tête de votre site WordPress.

```
<link rel="preconnect" href="//domaine.com">
```

Voici quelques exemples de choses pour lesquelles vous voudrez peut-être l'utiliser, notamment votre URL CDN ou les polices Google.

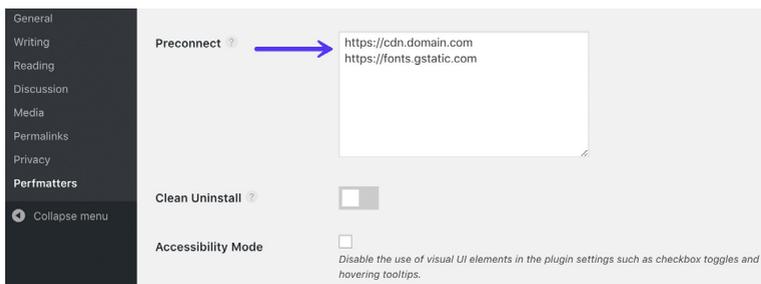
```
<link rel="preconnect" href="https://cdn.domaine.com">
```

```
<link rel="preconnect" href="https://fonts.gstatic.com">
```

Preconnect est supporté par la plupart des navigateurs modernes, à l'exception d'Internet Explorer, Safari, IOS Safari et Opera Mini. Il y a plusieurs façons d'implémenter cela.

- 1.** Vous pouvez ajouter le code à votre en-tête en utilisant un plugin gratuit comme *Head, Footer and Post Injections* (disponible sur [WordPress.org](https://wordpress.org/plugins/head-footer-and-post-injections/)).
- 2.** Vous pouvez ajouter le code manuellement dans le fichier `header.php` de votre thème.

- 3. Implémenter preconnect en utilisant un plugin comme Perfmatters** (disponible sur [perfmatters.io](https://perfmatters.io)). Cliquez simplement sur l'onglet "Extras" dans le plugin Perfmatters et ajoutez des domaines. Format : `scheme://domaine.tld` (un par ligne).



## Désactiver les scripts par page et par article

Une autre façon très puissante d'accélérer WordPress est d'analyser chaque requête qui se charge sur vos pages et articles. Vous finirez très probablement par trouver des scripts qui se chargent à l'échelle du site et qui ne devraient pas l'être.

Vous pouvez utiliser un plugin premium comme Perfmatters qui a une fonction "Script Manager" intégrée. Cela vous permet de désactiver les scripts (CSS et JavaScript) par page et par article, ou même à l'échelle du site en un seul clic. Encore une fois, ce plugin est développé par un membre de l'équipe de Kinsta.

Quelques exemples de ce à quoi cela peut servir :

- Le populaire plugin Contact Form 7 se charge à chaque page et à chaque article. Vous pouvez facilement le désactiver partout d'un seul clic et ne l'activer que sur votre page de contact.
- Les plugins de partage de médias sociaux ne devraient être chargés que sur vos articles. Vous pouvez facilement les désactiver partout et les charger uniquement sur les types d'articles, ou même les types d'articles personnalisés.

- Le plugin Table des matières (TOC) se charge sur chaque page et article. Avec le gestionnaire de scripts, vous pouvez facilement contrôler où vous voulez le charger.
- Si vous avez mis à niveau vers WordPress 5.0 et que vous n'utilisez pas l'éditeur de blocs Gutenberg, peut-être utilisez-vous encore l'éditeur classique ou un autre éditeur tiers, il y a deux scripts supplémentaires qui sont ajoutés en front-End à l'ensemble du site, que vous pouvez désactiver :  
`/wp-includes/css/dist/block-library/style.min.css` and `/wp-includes/css/dist/block-library/theme.min.css`.

## Pourquoi certains plugins sont-ils codés de cette façon ?

Vous vous demandez peut-être pourquoi tous les développeurs de plugins ne chargent pas leurs scripts uniquement lorsque le plugin est détecté sur la page ? C'est un peu plus compliqué que ça. Par exemple, si vous avez un plugin comme Contact Form 7, il a aussi des shortcodes qui vous permettent de le placer n'importe où. Cela inclut de le déposer dans un widget. Avec WordPress, il est beaucoup plus difficile d'interroger des données à partir de WordPress lorsque vous retirez des scripts plutôt que d'interroger des données à partir des métadonnées de l'article ou de la page.

Par conséquent, cela est souvent dû à des problèmes d'ergonomie. Moins ils ont de chance qu'un plugin se casse, moins ils auront de tickets et de support. Cependant, avec beaucoup de plugins sur le marché, il existe des moyens de contourner ce problème et de coder pour la performance s'ils le souhaitent. Malheureusement, le nombre de téléchargements et le nombre d'utilisateurs font parfois du codage pour la facilité d'utilisation une priorité.

## Visite du gestionnaire de scripts

Nous allons vous faire visiter le Script Manager. Après l'avoir activé dans votre barre d'outils, vous verrez tous les scripts qui se chargent sur l'URL courante, à la fois les fichiers JavaScript et CSS. Vous disposez alors des options suivantes :

1. **Statut On** : (Réglage par défaut)
2. **Statut Off** : Désactivé Partout (vous pouvez alors choisir les types d'articles sur lesquels vous voulez l'activer, ainsi que l'URL courante)
3. **Statut Off** : Désactivé uniquement sur l'URL courante (très utile pour l'utiliser sur votre page d'accueil)
4. **Statut Off** : Exceptions (URL actuelle, type d'article ou archive)

The screenshot displays the 'Script Manager' interface for WordPress. It features a sidebar with 'Script Manager', 'Global View', and 'Settings'. The main content area is titled 'Script Manager' and 'Manage scripts loading on the current page.' It is divided into two sections: 'Extended Widget Options' and 'Social Warfare'. Each section has a table of scripts with columns for 'Status', 'Script', 'Type', and 'Size'. The 'Extended Widget Options' section shows two scripts: 'jquery-widgetopts' (JS, 6.1 KB) and 'widgetopts-styles' (CSS, 13.6 KB), both with 'ON' status. The 'Social Warfare' section shows two scripts: 'social\_warfare\_script' (JS, 10 KB) and 'social\_warfare' (CSS, 45.2 KB), both with 'OFF' status. The 'social\_warfare\_script' entry includes options for 'Disabled' (Everywhere, Current URL), 'Exceptions' (Current URL), and 'Post Types' (Posts, Pages, Lists, Coupons).

**Tout est regroupé par nom de plugin ou de thème.** Il est donc très facile de désactiver un plugin entier à la fois. Généralement, un plugin WordPress aura à la fois un fichier JavaScript et un fichier CSS. Un thème WordPress peut avoir plus de 10 fichiers.

Après avoir sélectionné et/ou modifié les paramètres, assurez-vous d'appuyer sur « Enregistrer » en bas de l'écran. Vous pouvez ensuite faire un test dans un outil de test de vitesse de site Web pour vous assurer que les scripts ne se chargent plus sur la page ou l'article. N'oubliez pas de vider votre cache d'abord ! Et si quelque chose ne s'affiche pas correctement sur votre site, vous pouvez toujours le réactiver dans les paramètres

pour revenir à la normale.

Lors d'un test de vitesse par woorkup, ils ont pu **réduire les temps de chargement totaux de 20,2%**. Rien que sur leur page d'accueil, ils ont pu réduire le nombre de requêtes HTTP de 46 à 30. Leur taille de page est également passée de 506,3 Ko à 451,6 Ko.

## Analyser les performances de services tiers

Fondamentalement, tout ce que vous appelez de l'extérieur à partir de votre site a des conséquences sur le temps de chargement. Ce qui aggrave encore ce problème, c'est que certains d'entre eux n'ont des lenteurs que de façon intermittente, ce qui rend l'identification de la question encore plus difficile.

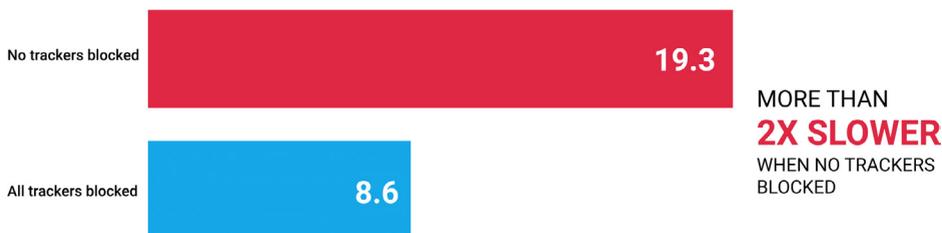
Un service externe tiers pourrait être considéré comme tout ce qui communique avec votre site WordPress depuis l'extérieur de votre propre serveur. Voici quelques exemples courants que nous rencontrons régulièrement :

- Plateformes de médias sociaux comme Twitter, Facebook et Instagram (widgets ou pixels de conversion)
- Réseaux publicitaires tiers comme Google AdSense, Media.net, BuySellAds, Amazon Associates, etc.
- Analyse de sites Web et scripts de suivi tels que Google Analytics, Crazy Egg, Hotjar, AdRoll, etc.
- Outils de A/B testing tels qu'Optimizely, VWO, Unbounce
- Systèmes de commentaires WordPress tels que Disqus, Jetpack, commentaires Facebook
- Outils de sauvegarde et de sécurité tels que VaultPress, Sucuri, CodeGuard
- Outils de partage tels que SumoMe, HelloBar
- Réseaux CDN comme KeyCDN, Amazon CloudFront, CDN77, StackPath
- Javascript hébergé en externe

Quel est l'impact de certains de ces trackers tiers sur les performances ? Dans notre propre étude de cas, nous avons constaté que **les scripts tiers augmentent le temps de chargement des pages de 86,08%**.

Ghostery a également mesuré les 500 meilleurs domaines américains dans Alexa, et les résultats ont été étonnants, bien que pour nous, pas surprenant. Les sites Web étaient 2x plus lents quand aucun tracker n'était bloqué du tout. Ce qui signifie que ces scripts de suivi tiers sont l'un des principaux facteurs qui contribuent à ralentir la vitesse de chargement des pages sur le Web.

## AVERAGE PAGE LOAD TIME WITH vs. WITHOUT TRACKERS

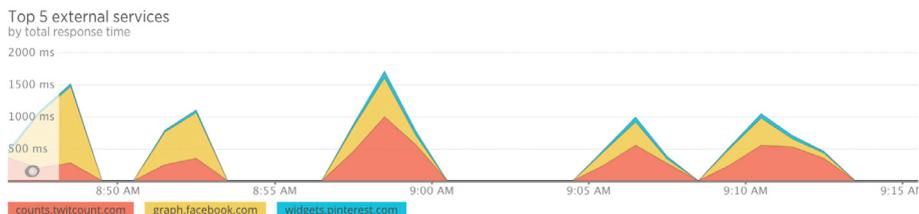


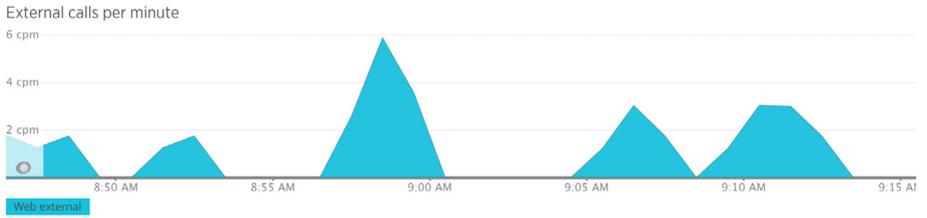
Average page load time (seconds) measured by Ghostery, of the top 500 US domains according to Alexa.

Source de l'image : Ghostery

Vous devez être très prudent sur votre site WordPress. Un seul mauvais appel vers l'API d'un service tiers pourrait faire perdre de la vitesse à tout votre site ! Oui, ça ne devrait pas marcher comme cela, mais dans bien des cas, oui. On l'a vu plus de fois qu'on ne peut compter.

New Relic est un excellent moyen simple et efficace de surveiller vos services externes au fil du temps. Dans l'exemple ci-dessous, nous pouvons voir des appels externes vers [twitcount.com](https://counts.twitcount.com), [graph.facebook.com](https://graph.facebook.com) et [widgets.pinterest.com](https://widgets.pinterest.com).

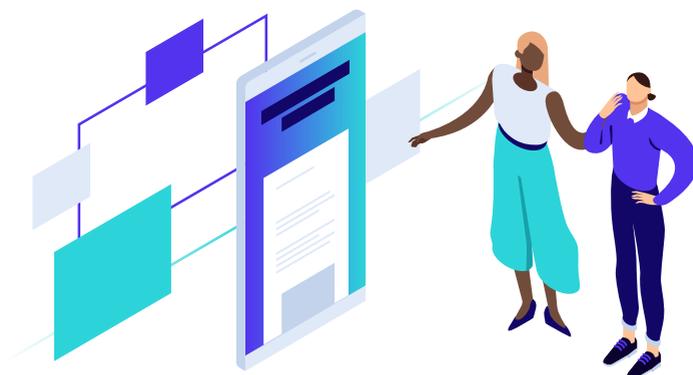




Il est important que chaque fois que vous ajoutez une nouvelle fonctionnalité ou un nouveau plugin à votre site, vous enquêtiez sur les ressources externes qui le chargent. Moins il y en a, mieux c'est !

CHAPITRE 16:

# Toujours optimiser avec le Mobile- First en tête



**Google a commencé à déployer son index mobile-first le 26 mars 2018. Auparavant, les systèmes de recherche, d'indexation et de classement de Google utilisaient la version de bureau des sites Web. L'indexation Mobile-first signifie que Googlebot utilisera désormais la version mobile de votre site WordPress pour l'indexation et le classement. Cela permet d'améliorer l'expérience de recherche pour les utilisateurs mobiles.**

Lorsqu'il s'agit d'optimiser votre site pour le mobile-first, **la vitesse est l'un des facteurs les plus importants à prendre en compte.** La vitesse joue un rôle majeur dans tout, de la facilité d'utilisation aux taux de rebond et à déterminer si les acheteurs potentiels reviendront ou non sur votre site. En fait, la vitesse est maintenant un facteur de base pour Google Search et les annonces pour les recherches mobiles.

De mauvaises expériences mobiles conduiront la majorité des utilisateurs à ne jamais revenir. Selon le dernier rapport sur la vitesse des pages Google, le temps moyen de chargement d'un site mobile en 2018 était de 15 secondes. Pouvez-vous imaginer attendre aussi longtemps pour charger une seule page ? Incroyable.

Les utilisateurs exigent (et méritent) mieux. Selon le même rapport sur la vitesse des pages, **53% des visiteurs de sites mobiles quittent des pages qui mettent plus de trois secondes à se charger.**

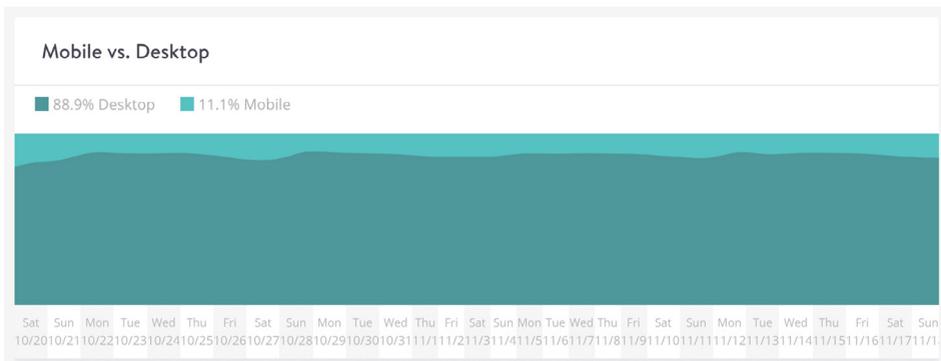
Les expériences mobiles lentes ne tuent pas seulement les conversions. Elles vous empêchent d’avoir une chance de convertir des prospects. Lorsque le temps de chargement des pages augmente de quelques secondes seulement, la probabilité que quelqu’un rebondisse augmente de façon exponentielle. Voici quelques points à prendre en compte lors de l’optimisation pour le mobile.

## Vérifiez votre trafic mobile

Il est toujours important de jeter un coup d’œil à la quantité de trafic mobile que vous recevez, car cela pourrait modifier un peu vos priorités. Vous pouvez voir combien d’appareils mobiles visitent votre site dans Google Analytics sous “Audience / Mobile / Overview”. Comme vous pouvez le voir sur ce site, plus de 67% de tout le trafic provient du mobile. C’est beaucoup !

Primary Dimension: Device Category		Acquisition	
Device Category		Users	New Users
		26,614 % of Total: 100.00% (26,614)	24,262 % of Total: 100.00% (24,261)
<input type="checkbox"/>	1. mobile	18,048 (67.93%)	16,174 (66.66%)
<input type="checkbox"/>	2. desktop	6,783 (25.53%)	6,459 (26.62%)
<input type="checkbox"/>	3. tablet	1,739 (6.54%)	1,629 (6.71%)

Si vous êtes un client Kinsta, vous pouvez également vérifier votre trafic mobile par rapport au trafic de bureau dans MyKinsta Analytics. Comme vous pouvez le voir sur ce site, plus de 88% du trafic provient du bureau. Il est toujours important de vérifier et non de supposer. Ce n’est pas parce que tout le monde dit que les choses deviennent mobiles, que c’est le cas pour votre site. Regardez les données.



## Assurez-vous que votre site est responsive

Tested on: Nov 18, 2018 at 11:13 PM

### Page is mobile friendly

This page is easy to use on a mobile device

Additional resources

- 📅 [Open site-wide mobile usability report](#)
- 🎓 [Learn more about mobile-friendly pages](#)
- 🎓 [Post comments or questions to our discussion group](#)

SCREENSHOT    HTML

En 2019, votre site web a intérêt à être responsive ! Cela signifie qu'il utilise des Media Queries pour réduire automatiquement l'échelle des choses sur les appareils mobiles. Si vous ne l'avez pas encore fait, vous êtes probablement déjà derrière vos concurrents. Tous les thèmes WordPress que nous avons mentionnés plus tôt dans cet article sont

pleinement responsives et sont impressionnants sur tous les appareils.

Utilisez l'outil Mobile-Friendly de Google pour tester et vérifier que votre site Web répond à toutes les exigences.

## Vérifiez deux fois pour vous assurer que le Srcset fonctionne

Dans le passé, il était très important que vous téléchiez des images à l'échelle et que vous ne laissiez pas le CSS les redimensionner. Cependant, ce n'est plus aussi important puisque WordPress 4.4 supporte maintenant les images responsives (non réduites par CSS). WordPress crée automatiquement plusieurs tailles de chaque image téléchargée dans la médiathèque. En incluant les tailles disponibles d'une image dans un attribut srcset, les navigateurs peuvent maintenant choisir de télécharger la taille la plus appropriée et ignorer les autres. Voyez un exemple de ce à quoi ressemble votre code ci-dessous.

```

```

En raison de tous les plugins et personnalisations d'images tiers, il y a eu beaucoup de fois où nous avons vu que cela ne fonctionnait pas correctement. Par conséquent, il est important de vérifier que vos images reçoivent correctement le srcset ajouté avec différentes versions pour différentes tailles d'écran. L'optimisation des images est maintenant importante pour toujours.

## Google AMP pourrait être une solution pour vous

Google AMP (Accelerated Mobile Pages Project) a été lancé en octobre 2015. Le projet s'appuie sur AMP HTML, un nouveau framework ouvert et entièrement construit à partir de technologies web existantes, qui permet aux sites web de construire des

pages web légères. Pour le dire simplement, il offre un moyen de proposer une version épurée de votre page Web actuelle.

Nous avons une sorte de relation d'amour et de haine avec Google AMP, et une grande partie de la communauté aussi. Nous l'avons testé nous-mêmes et nous n'avons pas vu de bons résultats. Cependant, cela ne veut pas dire que vous n'en verrez pas. Chaque site web est différent, et Google AMP est constamment amélioré.

Vous pouvez rapidement démarrer avec Google AMP sur votre site WordPress avec l'un des plug-ins suivants :

- AMP for WordPress (Gratuit, disponible sur WordPress.org)
- AMP for WP (Gratuit, disponible sur WordPress.org)

## Résumé

Comme vous pouvez probablement le voir, nous sommes obsédés par toutes les différentes façons dont vous pouvez accélérer WordPress. Avoir un site rapide permet d'augmenter votre classement, améliore l'exploration des moteurs de recherche, améliore les taux de conversion, augmente le temps passé sur le site, et diminue votre taux de rebond. Sans parler du fait que tout le monde aime visiter un site Web rapide !

Nous espérons que ce guide d'accélération vous a été utile et que vous avez pu en retirer quelques éléments et les appliquer à votre site WordPress.

N'oubliez pas de consulter également notre programme d'affiliation d'hébergement WordPress sur [kinsta.com/fr/affiliation/](https://kinsta.com/fr/affiliation/). Gagnez jusqu'à 500 \$ pour chaque recommandation + 10 % de **commissions mensuelles récurrentes à vie**. C'est la meilleure offre d'affiliation que vous trouverez dans l'industrie. Accélérons le web ensemble !

Pour en savoir plus, visitez <https://kinsta.com/fr/>.

Merci d'avoir lu !







La vitesse de votre site WordPress est plus importante que vous ne le pensez. Peu importe le type de site que vous utilisez, qu'il s'agisse d'une boutique WooCommerce ou d'un blog, les temps de chargement des pages sont en relation directe :

- Meilleur classement des mots-clés (la vitesse est un facteur pour Google).
- Augmenter le taux de conversion.
- Amélioration de l'exploration par les moteurs de recherche tels que Google et Bing.
- Temps moyen plus long passé sur votre site.
- Diminution du taux de rebond.

Sans parler du fait qu'un site Web qui se charge rapidement offre une meilleure expérience globale à vos visiteurs.

WordPress propulse maintenant plus d'un tiers de tous les sites Web sur Internet ! Bien que cela soit génial, cela signifie aussi qu'il y a des milliers de thèmes, de plugins et de technologies différents qui doivent tous coexister. Pour l'utilisateur WordPress de tous les jours, cela peut rapidement se transformer en cauchemar lorsqu'un site commence à ralentir. Beaucoup ne savent pas pourquoi ni même par où commencer le dépannage.

L'équipe de Kinsta s'occupe de ce type de problèmes au quotidien, et a tout vu ! Dans ce guide, nous examinons en profondeur les étapes applicables que vous pouvez suivre dès maintenant pour accélérer vos sites WordPress. Qu'il s'agisse de conseils pour débutants ou de stratégies avancées, vous trouverez forcément quelque chose d'utile. Il contient plus de 15 ans d'expérience et de dures leçons que nous avons apprises.