

ZO  
VERSNEL  
JE JE  
WORDPRESS  
SITE



Brian Jackson

**KINSTA**

Managed WordPress Hosting

# Zo versnel je je WordPress site

(ULTIEME GIDS)



Scan de QR code om meest recente versie van deze gids te bekijken of ga naar:  
<https://kinsta.com/nl/leren/wordpress-sneller-maken/>

Gepubliceerd door Kinsta

Boekontwerp door Maja Szakadat



## Over de auteur - Brian Jackson

Brian is de Chief Marketing Officer van Kinsta. Hij richt zich op alles van het ontwikkelen van nieuwe online groeistrategieën, het creëren van content, technische SEO en outreach binnen de gemeenschap. Hij schrijft met veel plezier long-form artikelen en heeft meer dan 2.500 blog posts gepubliceerd.

Een van Brians grootste passies is WordPress, dat hij al meer dan tien jaar gebruikt. Hij heeft gewerkt in zowel de content delivery network industrie als nu in de high-performance hosting sector en is gespecialiseerd in web performance optimalisatie en strategische manieren om websites te versnellen.

Zo versnel je je WordPress site

# Hoofdstukken

**08**

WordPress website types:  
Statisch of dynamisch

**11**

Kies hoogwaardige  
WordPress hosting

**23**

Kies een server die het  
dichtst bij je bezoekers staat

**32**

Premium DNS is beter  
dan gratis DNS

**36**

Je WordPress  
thema's

**45**

De ware feiten over  
WordPress plugins

**50**

Optimale WordPress  
instellingen

**56**

Waarom cache zo  
belangrijk is

**66**

Afbeeldingsoptimalisatie  
is een must

**75**

Tuning van je  
database

**93**

Gebruik van een Content  
Delivery Network (CDN)

**103**

Media en e-mail  
verwijderen wanneer dat  
nodig is

**108**

Knelpunten en trage  
plugins vinden

**127**

Aanbevelingen over  
back-end optimalisatie

**141**

Tips voor front-end  
optimalisatie

**168**

Optimaliseer altijd met  
Mobile-First in gedachten



## Zo versnel je je WordPress site

We hebben in de loop der jaren veel tutorials gepubliceerd met manieren om WordPress te optimaliseren en te versnellen. Maar soms kan het verwarrend zijn om alles wat je nodig hebt op één plek te vinden. Dus vandaag gaan we alles wat we weten over het turbochargen van WordPress met je delen, meer dan 15 jaar ervaring en harde lessen, allemaal in één ultieme gids. Of je nu net begint met WordPress of een doorgewinterde ontwikkelaar bent, we beloven je dat je iets nuttigs zult vinden in deze gids!

**Meer dan een derde van het web wordt nu aangedreven door WordPress.** Hoewel dit geweldig is, betekent het ook dat er duizenden verschillende thema's, plugins en technologieën naast elkaar moeten bestaan. Voor de alledaagse WordPress gebruiker kan dit snel veranderen in een nachtmerrie wanneer hun site langzamer wordt en ze niet weten waarom, of zelfs niet waar ze moeten beginnen met het oplossen van problemen.

Vandaag duiken we in toepasbare stappen die je meteen kan uitvoeren om verbeteringen te zien op je eigen WordPress sites. We zullen ook enkele bronnen delen die voor ons van onschatbare waarde zijn geweest.



HOOFDSTUK 01:

# WordPress website types: Statisch of dynamisch



Voordat we ons in de optimalisaties verdiepen, is het belangrijk om eerst te begrijpen dat **niet alle WordPress sites hetzelfde zijn**. Daarom hebben veel gebruikers problemen, omdat je niet elk probleem op dezelfde manier kunt aanpakken. We geven WordPress sites altijd een classificatie: statisch of dynamisch. Laten we dus eerst de verschillen tussen deze twee soorten sites onderzoeken.

## Statische sites

Statische sites zijn meestal sites zoals blogs, kleine bedrijfssites, nieuwssites met weinig verkeer, persoonlijke sites, fotografie sites, enz. Met statisch bedoelen we dat de gegevens op deze WordPress sites **niet vaak** (misschien een paar keer per dag) **veranderen**. Zelfs het grootste deel van onze Kinsta site zou als een statische website worden beschouwd.

Dit is belangrijk omdat veel van de aanvragen direct vanuit de cache op de server kunnen worden bediend met bliksemsnelle snelheden! Maak je geen zorgen, we duiken verderop dieper in het onderwerp caching. Dit betekent dat ze minder database-oproepen zullen hebben en dat er niet zoveel systeem resources nodig zullen zijn om Google-prestaties te bereiken.

## Dynamische sites

Aan de andere kant hebben we de dynamische sites. Dit zijn onder andere e-commerce- (WooCommerce of Easy Digital Downloads), community- en lidmaatschap-sites, forums (bbPress of BuddyPress) en learning management systemen (LMS). Met dynamisch bedoelen we dat de gegevens op deze WordPress sites **regelmatig veranderen** (er vinden om de paar minuten of zelfs om de seconde servertransacties plaats). Dit betekent dat niet alle verzoeken aan de server direct vanuit de cache kunnen worden bediend en dat er extra server resources en database queries nodig zijn.

Deze sites hebben doorgaans ook een **groot aantal gelijktijdige bezoekers en sessies**. Op een informatieve of zakelijke WordPress site die meestal statisch is, kan een bezoeker vijf of tien minuten blijven totdat hij of zij vindt wat hij of zij nodig heeft (en dit is een hoog aantal, meestal zijn de bounce rates veel hoger). Op dynamische sites is het tegenovergestelde het geval. Bezoekers komen meestal naar de site om zich met iets of iemand bezig te houden. Als ze een online cursus volgen, is het niet ongebruikelijk dat ze urenlang blijven.

Je kunt zien waar dit heen gaat. De gelijktijdige bezoekers die verbonden zijn met je WordPress host tellen snel op. Tot overmaat van ramp heb je dan een groot aantal gelijktijdige bezoekers bovenop een “uncacheable content” probleem.

*Je kunt niet alle WordPress sites hetzelfde behandelen als het gaat om prestaties. Statische en dynamische sites zijn twee zeer verschillende dingen.*

HOOFDSTUK 02:

# Kies hoogwaardige WordPress hosting



Een WordPress host is een bedrijf dat alle gegevens van je website opslaat. Je meldt je aan voor een pakket en al je foto's, teksten, video's, enz. bevinden zich op een server in het datacenter van de host. De WordPress host biedt je een eenvoudige manier om toegang te krijgen tot de gegevens, deze te beheren en naar je bezoekers te sturen. Vrij eenvoudig toch? Nou, niet helemaal.

Er zijn drie zeer verschillende soorten WordPress hosts die je op het web tegenkomt. Laten we kijken naar de voors en tegens van elk type. Het is belangrijk dat je in het begin de juiste keuze maakt, anders ga je jezelf hoofdpijn bezorgen en tijd verspillen.

## 1. Shared WordPress Hosting

De eerste en meest populaire vorm van WordPress hosting is wat we 'shared hosting' noemen. Dit zijn onder andere de grootste hosts in de industrie zoals EIG-bedrijven als Bluehost en HostGator, maar ook providers als Siteground, GoDaddy en InMotion Hosting. Ze maken meestal gebruik van cPanel, en de gemiddelde klant betaalt meestal tussen de \$3 en \$25 per maand.

Iedereen die dit soort hosting gebruikt zal op een bepaald moment traagheid ervaren, het is slechts een kwestie van tijd. Waarom? Omdat shared hosts de neiging hebben om hun servers te overbelasten, wat op zijn beurt de prestaties van jouw site kan beïnvloeden. Het schorsen van je account of het zien van frequente 500-fouten zijn

veel voorkomende dingen die je zult ervaren, dit omdat de hosts grenzen moeten stellen en systeem resources moeten consolideren om te overleven. Of erger nog, downtime van de website. Ook al weet je het niet, je WordPress site staat waarschijnlijk op dezelfde server als die van 200+ andere mensen. Alle problemen die voorkomen op andere sites kunnen ook op jouw site voor problemen zorgen.



Shared WordPress hosting

Het maakt niet uit hoe je het uitrekent, na de kosten, levert 3 dollar per maand geen inkomsten op voor het hostingbedrijf. Vooral als je daar ook support bij meetelt. Eén support ticket en ze staan al in het rood. De manier waarop ze veel van hun geld verdienen is met upselling en verborgen kosten. Deze upsells omvatten zaken als migraties, domeinregistraties, SSL-certificaten, etc. Een andere veel voorkomende tactiek is het geven van enorme korting bij inschrijving. Maar als het contract verloopt, krijg je de echte rekening.

De meeste van deze hosts bieden wat zij noemen hun “onbeperkte resources” plan aan. Je hebt dit waarschijnlijk allemaal gezien. Nou, er bestaat niet zo iets als onbeperkte resources. Wat hosts achter de schermen doen, is de klanten een groot deel van de resources laten gebruiken. Dit, op zijn beurt, eindigt met boze klanten die weggaan, waardoor er ruimte ontstaat voor meer klanten die niet veel resources gebruiken. Aan het einde heb je een vicieuze cirkel waar het hosting bedrijf goedkope pakketten promoot en hoopt dat nieuwe klanten niet veel resources gaan gebruiken en upsells zullen aanschaffen.

Klantenservice en ondersteuning bij shared hosting zijn bijna altijd ondergeschikt aan het volume van de sites ten opzichte van de supportmedewerkers. Shared hosts moeten veel moeite doen om zelfs maar een beetje winst te maken en dit leidt meestal tot een onaangename ervaring voor de klant.

*Als het gaat om shared hosting,  
krijg je meestal wat je betaalt.*

## 2. DHZ VPS WordPress Hosting

Het tweede type WordPress hosting is DHZ VPS, of “Doe het zelf op een virtuele privé-server”. De gebruikers van deze setup zijn meestal bootstrap startups en gebruikers met iets meer ervaring met ontwikkeling, serverbeheer en WordPress. Zij zijn de doe-het-zelvers. Deze mensen proberen meestal geld te besparen, maar ze zijn ook bezig met de prestaties en beseffen het belang ervan voor het succes van hun bedrijf. Veelgebruikte setups kunnen bestaan uit het gebruik van een VPS-provider zoals Digital Ocean, Linode of Vultr; samen met een tool zoals ServerPilot om het gemakkelijker te beheren.

Een kleine VPS van DigitalOcean begint bij \$5 per maand en het populaire pakket bij ServerPilot begint bij \$10 per maand. Dus afhankelijk van je installatie, kun je een kostprijs verwachten van tussen de \$5 tot \$15 of meer per maand. De DHZ aanpak kan kosten besparen, maar het betekent ook dat je verantwoordelijk bent als er iets kapot gaat, en voor het optimaliseren van je server voor de prestaties.

De doe-het-zelf aanpak kan geweldig zijn, maar het kan ook een averechtse uitwerking hebben op je als je niet voorzichtig bent. Ga niet op deze manier te werk als je niet technisch onderlegd bent of gewoon omdat je wilt sleutelen! Je tijd is geld waard en je zou het moeten besteden aan het laten groeien van je bedrijf.

### 3. Managed WordPress Hosting

Het derde type hosting is wat we bij Kinsta aanbieden en dat is managed WordPress hosting. Dit type hosts verzorgen alle back-end server gerelateerde taken voor je, samen met het verlenen van ondersteuning wanneer je dat nodig hebt. Ze zijn veelal afgestemd op het werken met WordPress en ondersteunen functies zoals één-klik-testomgevingen en automatische back-ups. Hun supportteams zullen meer kennis van zaken hebben als het gaat om het kennen van hun weg in het CMS, omdat ze zich dagelijks op één platform richten.

*Als je tijd wilt besparen, is managed  
WordPress hosting de manier!*

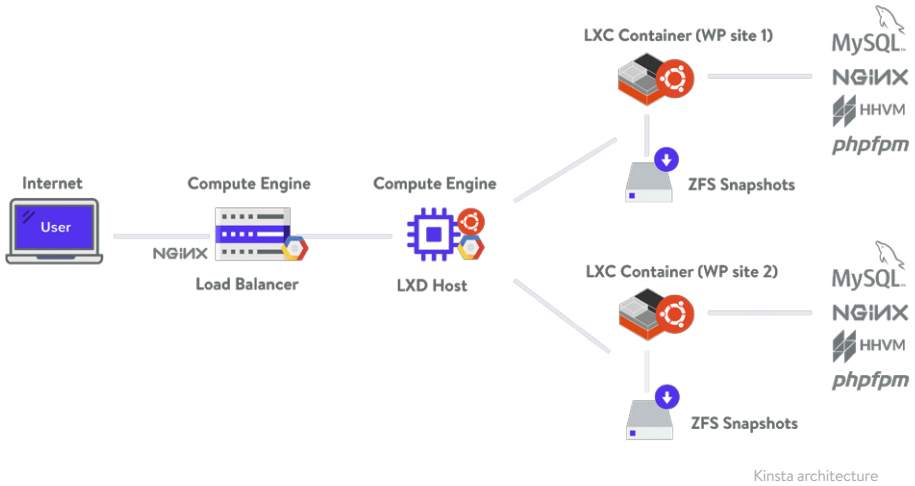
Pakketten voor managed WordPress hosting variëren meestal van \$25 tot \$150 per maand of meer, afhankelijk van de grootte van je site en de behoeften. Grote bedrijven zoals jQuery, Intuit, Plesk, Dyn, NGINX, en zelfs het Witte Huis gebruiken allemaal WordPress om hun website te hosten. Sommige populaire managed WordPress hosts die je waarschijnlijk kent, of misschien ook gebruikt, zijn onder andere WP Engine, Flywheel, Pressable, Media Temple, Pressidium, en Pagely.

### Kinsta kiest voor een andere aanpak

Kinsta brengt managed WordPress hosting echter naar een hoger niveau. Ons hosting platform valt niet in één van de traditionele hosting categorieën. Onze gehele infrastructuur is gebouwd op Google Cloud Platform en is anders dan de traditionele shared, VPS of dedicated infrastructuur.

Elke WordPress site op ons platform draait in een geïsoleerde softwarecontainer die alle softwareresources bevat die nodig zijn om de site te draaien (Linux, NGINX, PHP, MySQL). Dit betekent dat de software die elke site draait volledig privé is en niet gedeeld wordt, zelfs niet tussen je eigen sites.





Elke sitecontainer draait op virtuele machines in één van de vele GC-datacenters. Elke machine heeft tot 96 CPU's en honderden GB RAM. Hardwareresources (RAM/CPU) worden door onze virtuele machines automatisch aan elke sitecontainer toegewezen op basis van de behoefte (een mooie functie die we auto-scaling noemen).

Elk jaar brengt Review Signal hun WordPress hostingbenchmarks uit en we zijn er trots op dat Kinsta vijf jaar op rij heeft bewezen het beste bedrijf op alle niveaus te zijn! En niet alleen op één of twee van onze pakketten, maar op elk pakket, van Starter tot en met Enterprise.

“Kinsta had perfecte LoadStorm en Blitz-resultaten. Ze hadden ook geen problemen met andere tests. Ik heb geen woorden om hun prestaties te beschrijven.”



Kevin Ohashi  
Oprichter en WP-consultant, ReviewSignal

We hebben ook geen eerste- of tweedelijns supportmedewerkers. Ons hele supportteam bestaat uit WordPress ontwikkelaars en Linux-hosting-engineers, waarvan velen hun eigen servers hebben beheerd, thema's en plugins hebben

gemaakt en hebben bijgedragen aan WordPress. Dit zorgt ervoor dat je deskundig advies krijgt van iemand die actief WordPress gebruikt en ontwikkelt.

Je kan chatten met dezelfde support teamleden die onze Fortune 500 en zakelijke klanten ondersteunen. We zijn zo kieskeurig over de kwaliteit van ons supportteam dat we slechts minder dan 1% van de sollicitanten in dienst nemen. Een betere ondersteuning zul je nergens anders vinden!

Om meer te weten te komen over waarom je moet kiezen voor Kinsta voor managed WordPress hosting, ga je naar onze website [kinsta.com/nl](https://kinsta.com/nl). Ongeacht wie je kiest als je hostingprovider, je moet altijd op zoek gaan naar deze volgende serverfuncties om ervoor te zorgen dat je website zo snel mogelijk draait.

“Met WP Engine worden basisproblemen meestal snel opgelost. Echter, voor sommige problemen die complex zijn, zal een oplossing enige tijd in beslag nemen, en er zal veel heen en weer gecommuniceerd moeten worden. Dit is een probleem wanneer je een high-end WordPress site draait, en er is een dringend probleem dat snel moet worden opgelost. Als je vraagt om een aanbeveling tussen deze twee, is Kinsta naar mijn mening beter. Ze bieden veel meer dan ze beloven. Je hoeft je nooit zorgen te maken over de traagheid van de site, downtime, het krijgen van kwaliteitsondersteuning of andere hostinggerelateerde problemen.”



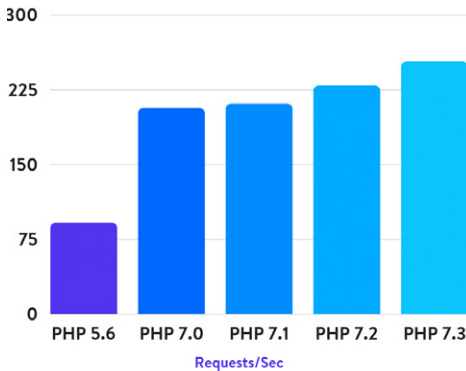
Harsh Agrawal  
Bekroonde Blogger, ShoutMeLoud

## PHP 7 of hoger voor de beste prestaties

PHP is een open-source, server-side scripting en programmeertaal die voornamelijk wordt gebruikt voor webontwikkeling. Het grootste deel van de WordPress software is geschreven in PHP, samen met plugins en thema's, wat PHP een zeer belangrijke taal maakt voor de WordPress gemeenschap. Je moet ervoor zorgen dat je WordPress host minstens PHP 7 of hoger aanbiedt.

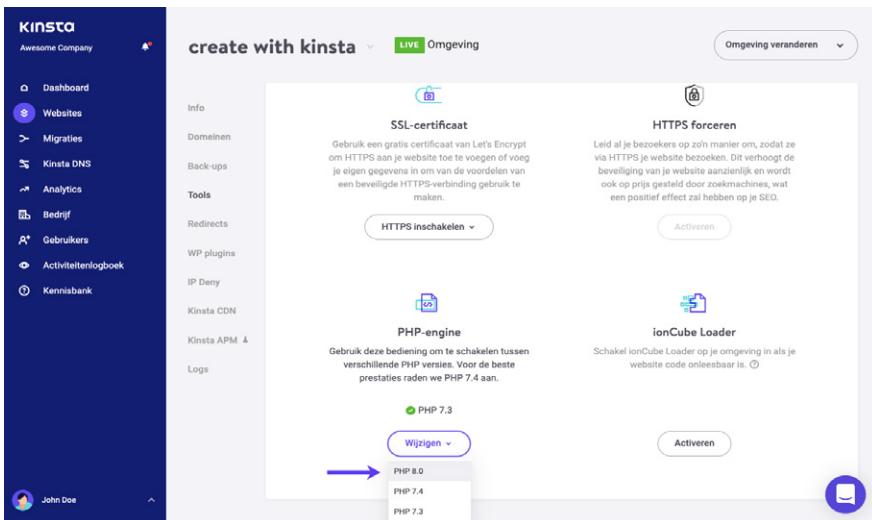
Als er verschillende versies van PHP zijn die je host op je server aanbiedt, kies dan voor PHP 7.3, die enorme prestatieverbeteringen biedt.

### WordPress 5.0



Uit onze recente PHP benchmark blijkt dat, als je PHP 7.3 vergelijkt met PHP 5.6, het 3x zoveel verzoeken (transacties) per seconde verwerken! PHP 7.3 is ook gemiddeld 9% sneller dan PHP 7.2. Dit kan ook invloed hebben op de snelheid van je WordPress admin dashboard.

Hogere snelheden plus verbeterde beveiliging, is de reden waarom Kinsta altijd de meest recente versies van PHP aanbiedt. Ook kun je met één muisklik van PHP-versie veranderen.



En let op voor WordPress hosters die HHVM aanbieden als alternatief voor PHP. HHVM is niet langer een geschikte oplossing voor WordPress hosting omdat het officieel geen PHP meer ondersteunt.

## Kies een host die **NGINX** gebruikt

Achter de schermen gebruikt elke WordPress host een webserver om je WordPress sites van stroom te voorzien. De meest voorkomende keuzes zijn NGINX en Apache.

Wij raden je sterk aan om een host te kiezen die NGINX gebruikt, vanwege de optimalisatie van de prestaties. NGINX presteert vaak beter dan andere populaire webserver in benchmark tests, vooral in situaties met statische content of een hoog aantal gelijktijdige verzoeken, daarom gebruikt Kinsta NGINX.

Enkele bekende bedrijven die NGINX gebruiken zijn Autodesk, Atlassian, Intuit, T-Mobile, GitLab, DuckDuckGo, Microsoft, IBM, Google, Adobe, Salesforce, VMWare, Xerox, LinkedIn, Cisco, Facebook, Target, Citrix Systems, Twitter, Apple, Intel en nog veel meer. (bron: Siftery.com)

Volgens W3Techs is Apache goed voor 44,0% van alle websites, waardoor het de meest gebruikte optie is. Maar als je kijkt naar de meest populaire webserver onder de high-traffic websites (top 10.000), dan heeft NGINX 41,9% van de markt, Apache slechts 18,1%. Het wordt gebruikt door enkele van de meest resource-intensieve sites die er bestaan, waaronder Netflix, NASA en zelfs WordPress.com.

## Het netwerk van je host

Bij het kiezen van een WordPress host denk je er misschien niet eens aan om te vragen of te onderzoeken welk netwerk ze gebruiken, maar je zou het wel moeten doen. Het netwerk kan een enorme impact hebben op de prestaties van je site en zelfs op de snelheid van je WordPress dashboard. Veel hosts zullen dit buiten hun marketing laten, omdat ze zullen kiezen voor het goedkoopste netwerk om kosten te besparen.

Hier zijn een paar vragen die je moet stellen:

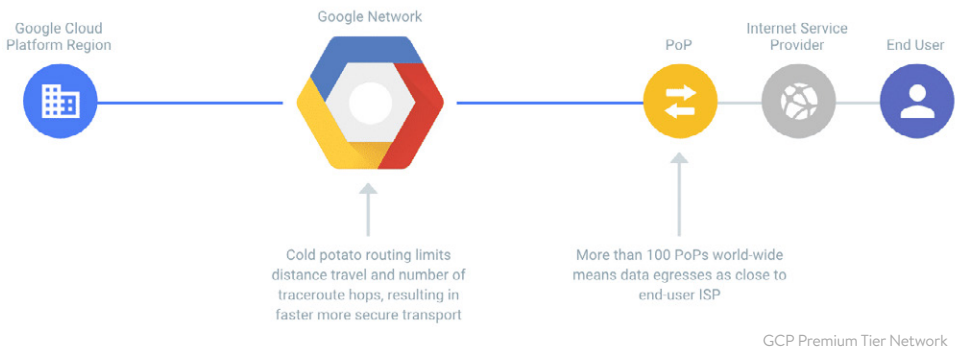
- **Over welke netwerken verstuurt je gegevens?** Gaat het merendeel ervan over openbare ISP-netwerken of particuliere infrastructures zoals Google of Microsoft? Deze grote providers hebben netwerken die zijn gebouwd en geoptimaliseerd voor lage latency en snelheid. Ze hebben zelfs hun eigen internetkabels op de oceaanvloer!

- **Zijn de netwerken die je gebruikt redundant?** Wat gebeurt er als een kabel per ongeluk wordt doorgesneden? Dit gebeurt vaker dan je denkt.

Al in 2017 kondigde Google hun standaard tier-netwerk aan, dat een trager netwerk is, maar tegen een lagere prijs. Bij Kinsta maken we gebruik van hun **premium tier netwerk** voor al onze hostingpakketten. Hoewel dit voor ons een extra kostenpost is, zorgt het ervoor dat je razendsnelle snelheden krijgt.

*Kinsta maakt gebruik van het premium tier-netwerk van het Google Cloud Platform. Want je websites verdienen het beste.*

Volgens Google bereikt het premium tier-netwerk betere netwerkprestaties door de duur van de reis op het openbare internet te verkorten; packets komen (en verlaten) het netwerk van Google zo dicht mogelijk bij de gebruiker en reizen vervolgens op de backbone van Google voordat ze de VM bereiken. Het standaard netwerk levert uitgaand verkeer van GCP naar het internet via een openbaar netwerk(ISP) in plaats van het netwerk van Google.



Om het anders te zeggen, dat is misschien makkelijker te begrijpen:

- **Premium tier-packets brengen meer tijd door op het netwerk van Google,** met minder bounces, en presteren dus beter (maar kosten meer).
- Standaard tier-packets besteden minder tijd op het netwerk van Google, en meer tijd op publieke netwerken, en presteren dus slechter (maar kosten minder).

Hoeveel impact heeft dit? Nou, voor gegevens die over de continenten reizen, is hun premium tier netwerk gemiddeld ongeveer **41% sneller** dan het standaard tier netwerk. Voor data die naar een nabijgelegen regio (hetzelfde continent) reist, is het premium tier netwerk ongeveer **8% sneller**. Terwijl netwerken slechts een fractie van de totale laadtijd van de pagina's uitmaken, telt elke milliseconde mee!

*Voor data die over de continenten reist, is het premium tier netwerk van Google Cloud gemiddeld 41% sneller!*

Redundantie is ook belangrijk en daarom gebruikt Google ten minste drie onafhankelijke paden (N+2-redundantie) tussen twee locaties op het Google-netwerk, zodat het verkeer tussen de locaties blijft doorstromen, zelfs in het geval van een storing.

Zoals je inmiddels wel kunt zien, gebeurt er achter de schermen veel als het gaat om netwerken. Zorg ervoor dat je WordPress host een gerenommeerd netwerk gebruikt en niet kiest voor de lagere niveaus om kosten te besparen.

## HTTP/2 is een must-have

HTTP/2 is een webprotocol dat in 2015 is uitgebracht en dat is ontworpen om de levering van websites te versnellen. Vanwege de browserondersteuning is HTTPS (SSL) vereist. Als je WordPress host geen HTTP/2 ondersteunt, moet je op zoek gaan naar een nieuwe provider. Met de verhuizing van het hele web naar HTTPS is dit niet langer alleen een leuke functie, maar ook een noodzaak.

De verbetering van de prestaties met HTTP/2 is te danken aan verschillende redenen, zoals ondersteuning van betere multiplexing, parallel laden, HPACK-compressie met Huffman-codering, de ALPN-extensie en serverpush. Vroeger was er nogal wat TLS

overhead met HTTPS, maar dit is nu een stuk minder dankzij HTTP/2 en TLS 1.3. **Kinsta ondersteunt HTTP/2 en TLS 1.3 op al onze servers en CDN.**

Een andere groot pluspunt van HTTP/2 is dat je je bij de meeste WordPress sites geen zorgen meer hoeft te maken over het combineren van bestanden of het gebruiken van meerdere (sub)domeinen. Dit zijn nu verouderde optimalisaties. En HTTP/3 is de volgende!

HOOFDSTUK 03:

# Kies een server die het dichtst bij je bezoekers staat

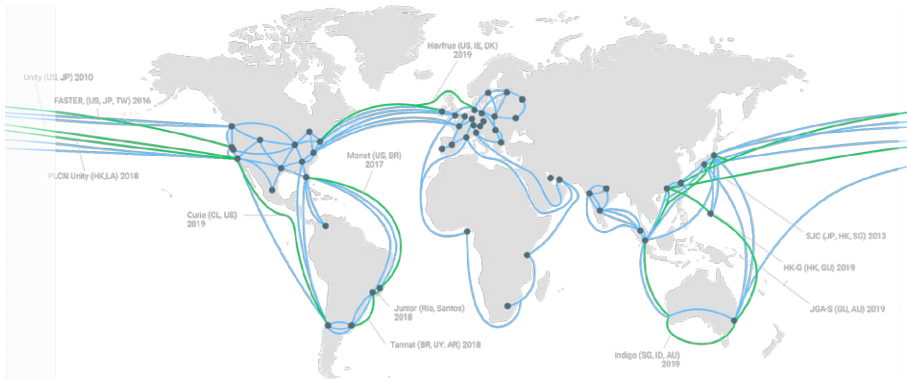




**Eén van de allereerste dingen die je moet doen bij het hosten van je WordPress site is bepalen waar de meerderheid van je bezoekers of klanten vandaan komen. Waarom is dit belangrijk? Omdat de locatie waar je je website host een belangrijke factor is bij het bepalen van de totale netwerk latency en TTFB. Het heeft ook invloed op je SFTP-snelheden en de reactiesnelheid van het WordPress dashboard.**

**Network Latency:** Dit heeft betrekking op de tijd en/of vertraging die gemoeid is met de overdracht van gegevens over een netwerk. Met andere woorden, hoe lang het duurt voordat een pakketje gegevens van het ene naar het andere punt gaat. Tegenwoordig wordt dit meestal in milliseconden gemeten; het kan echter seconden duren, afhankelijk van het netwerk. Hoe dichterbij nul, hoe beter.

Hoe groter de afstand, hoe hoger de latency, wat neerkomt op een grotere vertraging. Je kunt de snelheid van het licht niet verslaan als het gaat om gegevens die over glasvezelkabels reizen. Hieronder staat een kaart van het Google Cloud Platform netwerk. Er liggen duizenden kilometers kabel in de oceaan en over de hele wereld. Dit is één van de redenen waarom het één van de snelste netwerken ter wereld is. En zelfs met dit uitgebreide state of the art netwerk is er nog steeds sprake van vertragingen.



Er zijn een paar verschillende factoren die bijdragen aan vertragingen in het netwerk, waaronder:

- Transmissievertraging:** Verschillende soorten media, zoals draadloze of glasvezelverbindingen, introduceren allemaal enige vertraging omdat ze slechts een bepaald aantal bits kunnen verwerken. Zo zal een glasvezelverbinding een lagere transmissievertraging hebben dan een T1-lijn. Er moet ook rekening gehouden worden met de grootte van het pakket
- Verplaatsingsvertraging:** Dit is de tijd die het pakketje data nodig heeft om van punt A naar punt B verplaatst te worden. Dit kan onder meer zijn: afstand, netwerkvertragingen, enz.
- Wachrijvertraging:** Als de bandbreedte wordt overschreden, kunnen er wachrijvertragingen optreden waarbij de gegevens op de host of router moeten wachten. Dit kan worden beïnvloed door drukte op het netwerk.


Sommigen zouden kunnen beweren dat netwerklatency niet erg belangrijk is; het kan het echter wel zijn met betrekking tot de plaats waar je ervoor kiest om je WordPress site te hosten. Voor media en andere bestanden (zoals afbeeldingen, JavaScript, CSS, video) kan een CDN veel van de extra latency verhelpen, door een kopie van een server dichterbij aan te leveren. In de meeste configuraties moet je echter nog steeds je hostingsserver benaderen voor de initiële request. En dit is de reden waarom latency belangrijk is!

Latency is vooral belangrijk als het gaat om bedrijven die de bezoekers op een specifieke geografische locatie bedienen. Laten we bijvoorbeeld zeggen dat je een e-commerce winkel hebt in Sydney, en dat 90% van je klanten uit Australië komt.

Je bedrijf zou profiteren van het plaatsen van je site op een server in Australië, in tegenstelling tot het hosten van je site in Europa of de Verenigde Staten.

**TTFB:** Dit staat voor tijd tot de eerste byte (Time To First Byte). Simpel gezegd is dit een meting van hoe lang de browser moet wachten voordat hij zijn eerste byte aan gegevens van de server ontvangt. Hoe langer het duurt om die gegevens te krijgen, hoe langer het duurt om je pagina weer te geven. Nogmaals, hoe dichterbij nul, hoe beter.

Een veel voorkomende misvatting is dat dit wordt berekend na DNS lookup tijden, echter, de oorspronkelijke berekening van TTFB in het netwerk is altijd inclusief netwerk latency. Dit is 3-staps proces en vertragingen en latency kunnen overal tussenin voorkomen, waardoor je totale TTFB wordt opgeteld.

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. Verzoek aan de server:** Wanneer iemand je website bezoekt, wordt er eerst een HTTP-verzoek van de client (browser) naar de server gestuurd. In deze stap zijn er verschillende factoren die vertragingen kunnen veroorzaken. Trage DNS-zoektijden kunnen bijdragen aan een langere tijd voor het verzoek. Als de server geografisch ver weg staat, kan dit leiden tot vertraging in de afstand die de gegevens moeten afleggen. Ook, als je complexe firewall regels hebt, kan dit de routing tijd verhogen. En vergeet de internetsnelheid van de client niet.

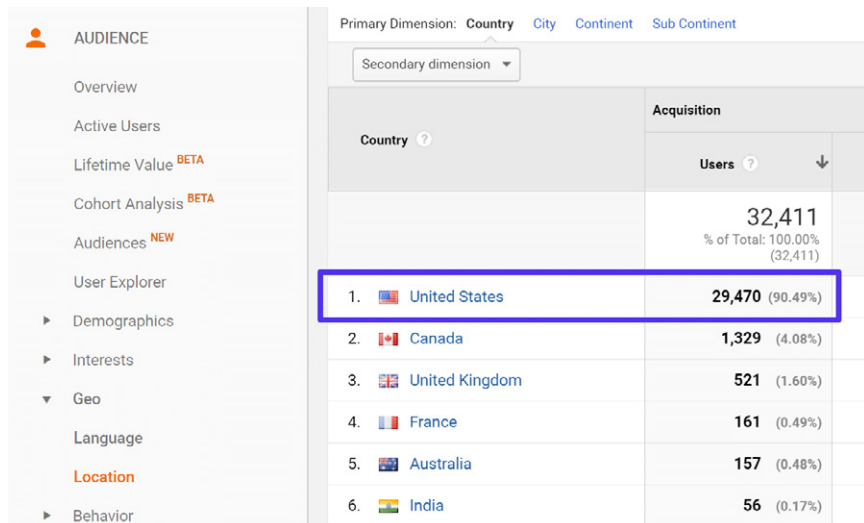
**2. Serververwerking:** Nadat de aanvraag is verzonden, moet de server deze verwerken en een antwoord genereren. Dit kan een aantal verschillende vertragingen met zich meebrengen, zoals langzame database-queries, te veel externe scripts, het niet cachen van de eerste request, slecht geoptimaliseerde code of WordPress thema, en inefficiënte server resources zoals schijf-I/O of geheugen.

**3. Response naar de browser:** Nadat de server de aanvraag verwerkt heeft, moet hij deze terugsturen naar de client (of beter gezegd, de eerste byte terugsturen). Dit wordt sterk beïnvloed door zowel de netwerksnelheid van de server als van de client. Als de client traag internet heeft vanaf een Wi-Fi hotspot, zal dit in de TTFB tot uiting komen.

Over het algemeen **is alles onder 100 ms geweldig en een goede TTFB**. Google PageSpeed Insights adviseert onder 200 ms voor de server response tijd. Als je site zich in het bereik van 300-500 ms bevindt, is dit vrij standaard. En als de TTFB meer dan 600 ms is, kan het zijn dat je iets verkeerd geconfigureerd hebt op je server of dat het tijd is om te upgraden naar een betere webstack. Of volg onze aanbevelingen hieronder over hoe je je TTFB kunt verminderen. En vergeet niet dat SSL/TLS onderhandelingen ook een factor kunnen zijn.

We zullen je niet lastigvallen met alle technische details, alles wat **je moet weten is dat je wilt dat je netwerklancy en TTFB zo laag mogelijk zijn**. Eén van de eenvoudigste manieren om dit te bereiken is door een server te kiezen die het dichtst bij je bezoekers staat. Je kunt de beste locatie bepalen door onderstaande tips te volgen.

## Tip 1 - Controleer de geolocatie van je bezoekers in Google Analytics.



The screenshot shows the Google Analytics interface for the 'AUDIENCE' section. The primary dimension is set to 'Country'. The table displays the top 6 countries by user acquisition. The 'United States' row is highlighted with a blue box.

Primary Dimension: Country		Acquisition	
Country		Users	
		32,411	% of Total: 100.00% (32,411)
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%)

Eén van de allereerste dingen die je kunt doen is kijken naar de geolocatie van je bezoekers in Google Analytics. Deze kan je vinden onder “Publiek / Geo / Locatie”.

In dit voorbeeld zie je dat meer dan 90% van het verkeer uit de Verenigde Staten komt. Dus in de meeste gevallen zou je je WordPress site op een server in de Verenigde Staten moeten plaatsen. Je kan de gegevens ook nog verder naar de steden filteren. Dit is vooral belangrijk als je een lokaal bedrijf hebt. In dit geval zouden we een centrale locatie als Iowa, USA aanraden.

## Tip 2 - Controleer de e-commerce gegevens

Als je een e-commerce winkel runt, zorg er dan voor dat je ook controleert waar je klanten vandaan komen. Dit is natuurlijk hoe je inkomsten genereert, dus dit zijn je belangrijkste bezoekers. Dit zou moeten samenvallen met je verkeer hierboven; alleen is dat niet altijd het geval. Als je e-commerce data of doelstellingen in Google Analytics hebt, kan je die informatie eenvoudig over de geolocatiegegevens heen leggen om een beter geïnformeerde beslissing te nemen. Of controleer de locatie-informatie die is opgeslagen in de database van je e-commerce-platform.

## Tip 3 - Doe een Quick Latency Test

Er zijn een heleboel handige gratis tools om de latency van je huidige locatie te meten voor verschillende cloudaanbieders. Dit kan je helpen om snel te evalueren welke regio de beste keuze is voor je site.

- **gcping.com:** (meet de latency naar Google Cloud Platform regio's, inclusief Kinsta-servers)
- **cloudping.info:** (meet de latency naar Amazon regio's)
- **azurespeed.com:** (meet de latency naar Azure regio's)

In dit voorbeeld hieronder zien we dat Los Angeles, USA (us-west2) het snelst is van waar we gevestigd zijn. Als je echter klanten in de hele Verenigde Staten bedient, is

het misschien beter om te kiezen voor Iowa, USA (us-central1) om een lage latency te garanderen voor bezoekers van zowel de west- als de oostkust.

## Meet de latency van GCP regio's

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

Bij Kinsta bieden we 24 verschillende datacenters aan over de hele wereld. Je kunt eenvoudig een site kiezen die zowel een lage latency als een lage TTFB heeft! Dit helpt ook bij het verminderen van de netwerkhop.



Kies een server die het dichtst bij je bezoekers staat

## Extra manieren om de latency en TTFB te verminderen

Naast het kiezen van een dichtbij gelegen serverlocatie zijn er nog een paar andere manieren om de latency te verminderen.

- Implementeer caching op je WordPress site. In onze tests heeft caching onze TTFB met maar liefst 90% gereduceerd!
- Gebruik maken van een content delivery network (CDN) voor het serveren van gecachte bestanden van POP's over de hele wereld. Dit helpt de netwerklatency te verlagen voor bezoekers die misschien niet in de buurt van je server zijn.
- Maak gebruik van het HTTP/2-protocol om het aantal verzoeken tot een minimum te beperken, dankzij de parallelisatie. HTTP/2 is ingeschakeld op alle Kinsta-servers.
- Verminder het aantal externe HTTP-verzoeken. Deze kunnen elk hun eigen toegevoegde latency hebben op basis van de locatie van hun server.
- DNS speelt een rol in TTFB, dus je moet een eersteklas DNS-provider gebruiken met snelle responstijden.
- Gebruik prefetch en prerender om taken achter de schermen uit te voeren terwijl de pagina wordt geladen.

Maak je geen zorgen; we zullen alle bovenstaande aanbevelingen verderop in dit boek behandelen.

## SFTP-snelheden en het WordPress Admin Dashboard

Je bezoekers en klanten moeten altijd je prioriteit zijn. Maar een ander aspect van de prestaties waar velen het niet over hebben, is hoe **sommige van deze beslissingen je dagelijks werk beïnvloeden**. De locatie van het datacenter die je kiest, heeft invloed op de snelheid van je SFTP-download en -upload (het overzetten van bestanden met een FTP-client) en op de snelheid van je WordPress Admin dashboard.

Dus terwijl je er zeker van wilt zijn dat je een locatie kiest die het beste is voor je bezoekers, moet je er ook rekening mee houden dat dit invloed kan hebben

op het beheer van de site. Taken zoals het uploaden van bestanden naar de WordPress mediabibliotheek zullen sneller gaan wanneer je site wordt gehost op een datacenter in de buurt.

We horen consequent van onze klanten dat ze verbaasd zijn over hoeveel sneller hun admin dashboard bij ons is. Er zijn een aantal factoren die hierop van invloed zijn, maar het hebben van 24 verschillende datacenters is een grote uitdaging! Kies een locatie die zowel voor je bezoekers als voor jou werkt! Jij bent immers degene die waarschijnlijk duizenden uren gaat werken aan je website.

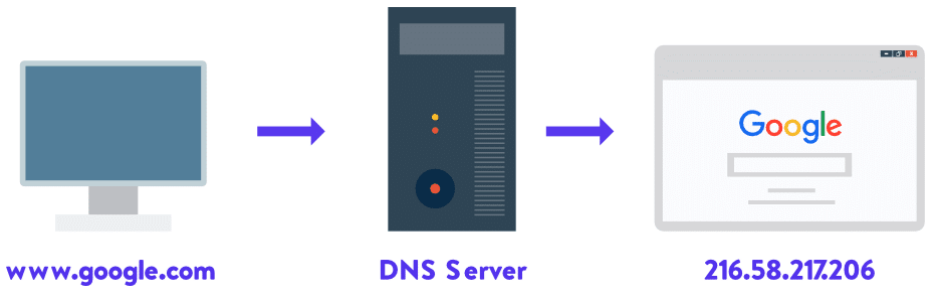


HOOFDSTUK 04:

# Premium DNS is beter dan gratis DNS



DNS, kortweg Domain Name System, is één van de meest voorkomende maar vaak verkeerd begrepen componenten van het weblandschap. Eenvoudig gezegd helpt DNS het verkeer op het internet te sturen door domeinnamen te verbinden met echte webserver. In wezen is er een mensvriendelijk verzoek nodig - een domeinnaam zoals `kinsta.com` - welke vertaald wordt naar een computer-vriendelijk IP-adres - zoals `216.58.217.206`.



How DNS works

Je kunt zowel gratis DNS als premium DNS gebruiken. Alle Kinsta klanten krijgen toegang tot premium DNS via Amazon Route 53. En in het algemeen geloven we dat premium DNS een noodzaak is in de wereld van vandaag.

Een grote reden om te kiezen voor premium DNS is **snelheid en betrouwbaarheid**. Het opzoeken van DNS-records en het doorsturen van verkeer kost tijd, al is het maar een kwestie van milliseconden.

Meestal is de gratis DNS die je krijgt van je domeinnaam registrar relatief langzaam, terwijl premium DNS vaak betere prestaties biedt. Zo vonden we in onze tests de gratis **NameCheap DNS 33% langzamer** dan Amazon Route 53 premium DNS. Bovendien kan premium DNS een betere beveiliging en beschikbaarheid bieden, vooral wanneer je site slachtoffer is van een DDoS-aanval.

Je kunt een tool als SolveDNS gebruiken om je DNS-zoektijden te controleren. DNSPerf biedt ook uitstekende prestatiegegevens over alle top DNS-providers.

Voor een goede middenweg tussen het gratis DNS van je domeinregistrar en het premium DNS, is Cloudflare DNS een gratis dienst die nog steeds veel van de voordelen van premium DNS biedt. En ze zijn razendsnel met een gemiddelde reactietijd van minder dan 20 ms over de hele wereld (zie hieronder).

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]

Echter, een voorbehoud bij Cloudflare is dat het ook meer downtime heeft dan veel andere aanbieders. Als je voornamelijk bezoekers in de Verenigde Staten bedient, is DNS Made Easy een andere grote premium DNS-provider die je misschien wilt bekijken. Ze hebben een reputatie voor het leveren van enkele van de beste DNS uptimes in de afgelopen tien jaar.

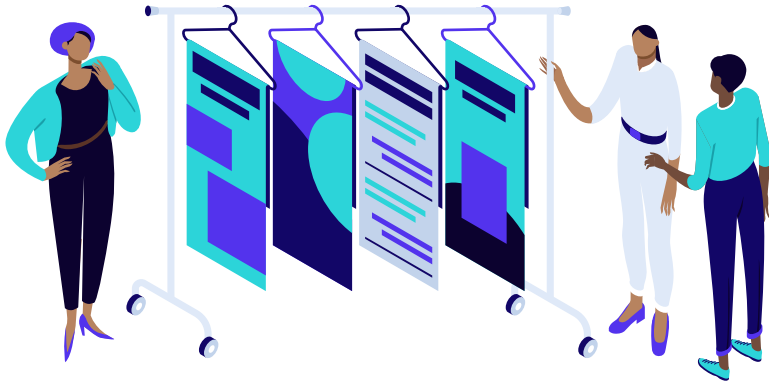
In een periode van 30 dagen heeft DNSPerf de volgende uptime van deze providers laten zien:

- DNS Made Easy: 99,99% wat gelijk staat aan 4m 23,0s maandelijkse downtime.
- Amazon route 53: 99,88% wat gelijk staat aan 52m 35,7s maandelijkse downtime..
- Cloudflare: 99,85% wat gelijk staat aan 1 uur 5m 44,6s maandelijkse downtime..

Is downtime zo belangrijk bij DNS-providers? Het antwoord hierop is echt ja en nee. DNS wordt bij ISP's meestal gecached met behulp van de time to live-waarde (TTL) op het DNS-record. Als een DNS-provider dus 10 minuten lang downtime heeft, zul je waarschijnlijk niets merken. Downtime doet er echter wel toe als de provider consequent langere en frequente onderbrekingen heeft, of als je ISP en DNS-records allebei echt lage TTL-waarden gebruiken.

HOOFDSTUK 05:

# Je WordPress thema's



Iedereen houdt van een gloednieuw WordPress thema, maar wees voorzichtig voordat je een nieuwe gaat installeren met allerlei nieuwe mooie functies. Er zijn veel verschillen als het gaat om gratis versus betaalde thema's. Wat betreft de prestaties heeft elk element dat je ziet in een thema enige invloed op de algehele snelheid van je website. En helaas, met duizenden beschikbare thema's, zijn er zowel goede als slechte thema's.

*Je WordPress thema is belangrijk voor de prestaties van je site. Kies vanaf het begin de beste.*

Hoe moet je weten welke je moet kiezen? We raden aan om te kiezen voor één van de volgende twee opties:

- Een snel lichtgewicht WordPress thema dat is gebouwd met alleen de functies die je nodig hebt, niets meer.
- Een meer functie-rijk WordPress thema, maar je kunt functies die niet in gebruik zijn uitschakelen.

Denk aan dingen als Google Fonts, Font Awesome iconen, sliders, galerijen, video en parallax scripts, etc. Dit zijn slechts een paar van de vele dingen die je zou moeten kunnen

uitschakelen als je ze niet gebruikt. Je wilt niet dat je deze dingen zelf moet uitschakelen nadat je alles opgezet hebt. En we gaan je geen 50 verschillende manieren laten zien om dingen uit te schakelen. In plaats daarvan moet je beginnen met of overschakelen naar een WordPress thema dat vanaf het begin licht is of je deze opties geeft.

Hieronder staan een aantal WordPress thema's die we aanraden en waar je niet mee de mist in kan gaan! Vertrouw ons, je kan ons later bedanken.

Elk onderstaand thema is volledig compatibel met WooCommerce en Easy Digital Downloads, WPML, BuddyPress en bbPress. We voeren een paar snelheidstesten uit met elk thema met behulp van de volgende configuratie:

- Gehost op Kinsta, waar WordPress 5.0 draait.
- PHP 7.3 en SSL (HTTPS)
- Kinsta CDN
- Imagify is gebruikt voor optimalisatie van afbeeldingen.

## GeneratePress

GeneratePress is een snel, lichtgewicht (minder dan 1MB gezip), mobiel responsive WordPress thema gebouwd met snelheid, SEO en gebruiksvriendelijkheid in het achterhoofd. Gebouwd door Tom Osborne, een ontwikkelaar uit Canada. Het wordt actief bijgewerkt en goed ondersteund. Zelfs een paar Kinsta teamleden gebruiken GeneratePress voor hun projecten.

Er is zowel een gratis als een premium versie beschikbaar. Als je kijkt naar WordPress.org, dan heeft de gratis versie op dit moment meer dan 200.000 actieve installaties, 2+ miljoen downloads en een indrukwekkende 5 van de 5-sterrenclassificatie (meer dan 850 mensen hebben 5 sterren gegeven).

Eén van de mooie dingen van GeneratePress is dat alle opties gebruik maken van de native WordPress Customizer, wat betekent dat je elke verandering die je maakt direct kunt zien voordat je op de publicatie knop drukt. Dit betekent ook dat je geen nieuw thema bedieningspaneel hoeft te leren.

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

Install

Premium

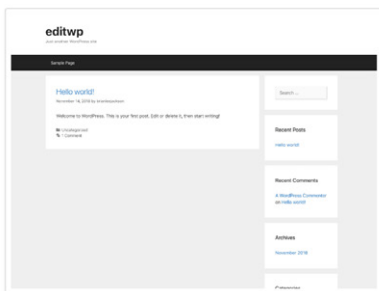
**1,643,956+**  
Downloads

**700+**  
★★★★★

**100,000+**  
Active websites

**40,000+**  
Happy customers

Hoe snel is het? We hebben een nieuwe installatie van GeneratePress geïnstalleerd, vijf snelheidstesten in Pingdom gedaan en het gemiddelde genomen. De totale laadtijd was **305 ms** met een totale **paginagrootte van slechts 16,8 KB**. Het is altijd goed om een basistest te hebben om te zien waartoe het thema in staat is in termen van ruwe prestaties.



Performance grade

**A 97**

Page size

**16.8 KB**

Load time

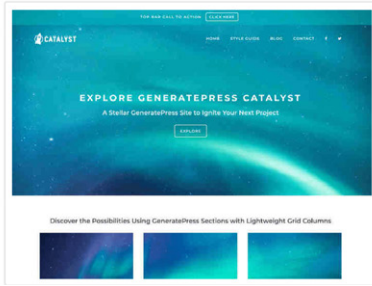
**305 ms**

Requests

**7**

Vervolgens hebben we nog een aantal tests uitgevoerd met één van de kant-en-klare thema's uit de GeneratePress site library. Deze bevat afbeeldingen, achtergronden, nieuwe secties, etc. Een voordeel van GeneratePress is dat het veel kant-en-klare thema's heeft waar geen plugin voor nodig is. Je kunt zien dat het nog steeds onder **400 ms** presteert.





Performance grade

**B 88**

Page size  
**837.5 KB**

Load time  
**396 ms**

Requests  
**36**

In een real-world omgeving heb je misschien andere dingen lopen, zoals Google Analytics, Facebook remarketing pixel, Hotjar, enz. Maar je zou gemakkelijk moeten kunnen mikken onder de 1-seconde.

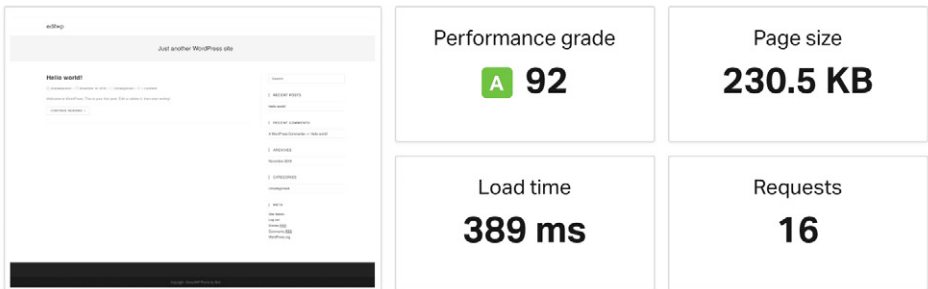
We laten je later in dit boek meer manieren zien waarop je WordPress kunt optimaliseren en versnellen.

## OceanWP

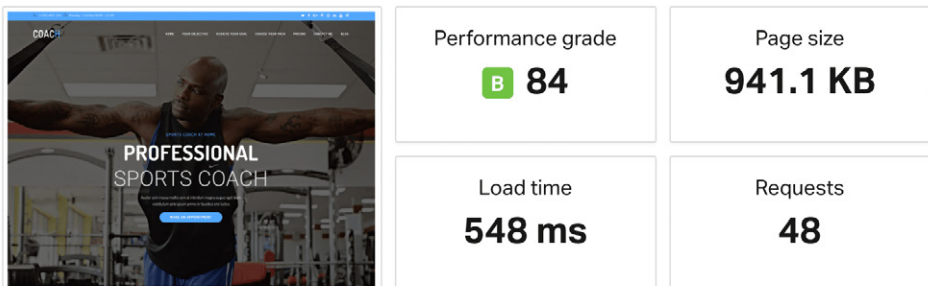
Het OceanWP thema is licht van gewicht en zeer uitbreidbaar. Het stelt je in staat om bijna elk type website te maken, zoals een blog, portfolio, zakelijke website of WooCommerce site met een mooi & professioneel ontwerp. Gebouwd door Nicolas Lecocq, wordt het ook actief bijgewerkt en goed ondersteund.

Net als bij GeneratePress is er zowel een gratis als een premium versie beschikbaar. Als je kijkt naar WordPress.org, dan heeft de gratis versie op dit moment meer dan 400.000 actieve installaties, en een indrukwekkende 5-sterrenclassificatie (meer dan 2.600 mensen hebben 5 sterren gegeven).

Hoe snel is het? We hebben een nieuwe installatie van OceanWP neergezet, vijf snelheidstesten in Pingdom gedaan en het gemiddelde genomen. De totale laadtijd was **389 ms** met een totale **paginagrootte van slechts 230,8KB**. De scripts in OceanWP zijn iets groter, maar zorgen niet voor veel vertraging.



Vervolgens hebben we nog een aantal tests uitgevoerd met een van de demo thema's uit de OceanWP site library. Deze bevat afbeeldingen, achtergronden, nieuwe secties en vereiste de Elementor page builder plugin. Je kunt zien dat het nog steeds onder **600 ms** presteert.

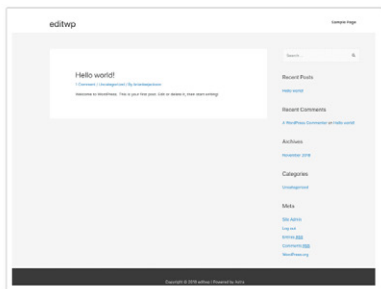


# Astra

Astra is een snel, volledig aanpasbaar & mooi thema geschikt voor blogs, persoonlijke portefeuilles, zakelijke websites en WooCommerce winkels. Het is zeer licht (minder dan 50 KB aan de voorkant) en biedt een ongeëvenaarde snelheid. Gebouwd door het team van Brainstorm Force, wordt het actief geupdate en goed ondersteund. Je zou ze kunnen herkennen als de makers van de populaire All In One Schema Rich Snippets plugin die al vele jaren bestaat.

Net als bij GeneratePress en OceanWP is er zowel een gratis als een premium versie beschikbaar. Als je kijkt naar WordPress.org, dan heeft de gratis versie op dit moment meer dan 400.000 actieve installaties, 1,6+ miljoen downloads, en nog eens een indrukwekkende 5 van de 5-sterren waardering (meer dan 2.500 mensen hebben 5 sterren gegeven).

Hoe snel is het? We hebben een nieuwe installatie van Astra neergezet, vijf snelheidstesten in Pingdom gedaan en het gemiddelde genomen. De totale laadtijd was **243 ms** met een totale **paginagrootte van slechts 26,6 KB**.



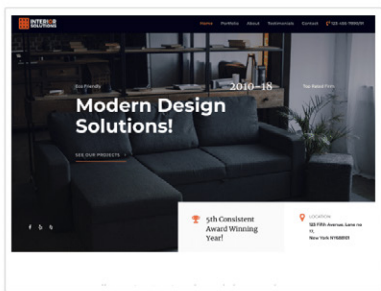
Performance grade  
**A 98**

Page size  
**26.6 KB**

Load time  
**243 ms**

Requests  
**5**

Vervolgens hebben we nog een aantal tests uitgevoerd met één van de demothema's uit de Astra Starter kit-bibliotheek. Deze bevat afbeeldingen, achtergronden, nieuwe secties en vereiste de Elementor page builder plugin. Je kunt zien dat de site nog steeds onder 700 ms geklokt is. Let op: de afbeeldingen in deze demo waren volledig gecomprimeerd, de makers hadden gekozen voor afbeeldingen van hoge resolutie.



Performance grade  
**B 86**

Page size  
**1.6 MB**

Load time  
**652 ms**

Requests  
**54**

Het is belangrijk om de verschillen tussen de snelheidstesten met deze drie thema's met een korreltje zout te nemen. Het probleem is dat het bijna onmogelijk is om een volledig nauwkeurige vergelijking te maken. Het belangrijkste dat we wilden laten zien is dat al deze WordPress thema's razendsnel zijn, zowel out of the box als met de meegeleverde demo's!

## Waarschuwing over page builders

Zoals je waarschijnlijk hebt gemerkt, hadden OceanWP en Astra beide page builders nodig om de thema's van de bibliotheek van hun site te gebruiken. Hier zijn een paar dingen om in gedachten te houden bij het gebruik van een plugin voor page builders:

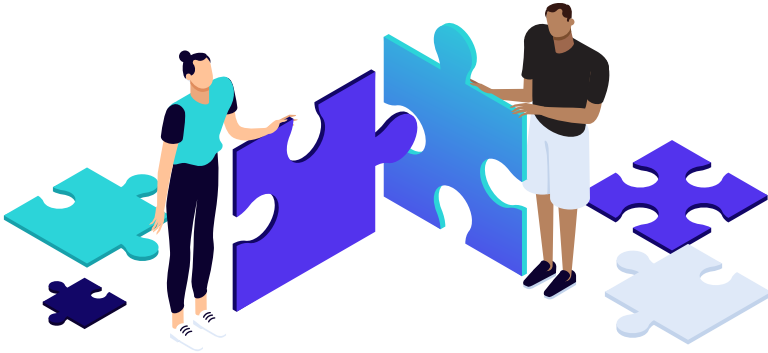
- Sommige page builders kunnen de laadtijd van je site verlengen. Dit komt omdat ze extra CSS en JS moeten laden om de dingen voor je te laten werken zonder code. We raden altijd aan om je WordPress site te testen voor en na de installatie van een page builder.
- Je maakt jezelf afhankelijk van de page builder voor het ontwerp. Zorg ervoor dat je er één kiest die regelmatig wordt bijgewerkt en alles heeft wat je nodig hebt voor de lange termijn.

Met dat gezegd hebbende, zijn we nog steeds grote fans van page builders zoals Elementor en Beaver Builder. Ze zijn voor het grootste deel ontwikkeld met het oog op prestaties en voegen slechts een klein beetje overhead toe. Voor de meesten is de functionaliteit en gebruiksvriendelijkheid de moeite waard, want met deze plugins kun je alles maken wat je maar kunt bedenken! In sommige gevallen zijn ze ook sneller, omdat ze een vervanging kunnen zijn voor 5+ andere plugins die je anders had moeten gebruiken.

Als je geen plugin voor page builders nodig hebt, ga deze dan niet zomaar installeren. Het zal ook interessant zijn om te zien hoe de nieuwe Gutenberg editor de komende jaren een rol zal spelen in het ontwerp van de site.

HOOFDSTUK 06:

# De ware feiten over WordPress plugins



En dan nu een primeur met betrekking tot WordPress plugins. Er is je misschien verteld dat je niet te veel plugins moet installeren, anders zou het je WordPress site vertragen. Hoewel dit soms waar is, is het niet de meest kritische factor. **Het aantal plugins is niet zo belangrijk als de kwaliteit van de plugins.** Zo, we hebben het gezegd.

Net als bij thema's is het van belang hoe de plugin wordt ontwikkeld en of deze is gebouwd met het oog op prestaties. We hebben bij Kinsta veel klanten die 30-40 plugins gebruiken en hun sites laden nog steeds ruim onder een seconde.

Hoewel het leuk is om code toe te voegen aan je site, is dit niet altijd praktisch om de volgende redenen:

- 1.** Je moet de code zelf onderhouden en aanpassen als de normen veranderen. Mensen hebben het druk, waarom zou je niet vertrouwen op de fantastische ontwikkelaars die de standaarden beter kennen dan de meesten?
- 2.** Meestal zal een goed gecodeerde plugin niet veel meer overhead introduceren dan de code zelf.
- 3.** Je moet niet vergeten dat een meerderheid van de WordPress gemeenschap niet zo tech-savvy is als de ontwikkelaars. Plugins zijn oplossingen die helpen bij het oplossen van problemen.

Met dat gezegd hebbende, er zijn natuurlijk ook plugins beschikbaar die je wilt vermijden. Echt, we hebben het ergste van het ergste gezien bij Kinsta. Veel, niet alle,

van de plugins die we bij Kinsta verboden hebben, veroorzaken problemen met de prestaties.

Het kan echter niet genegeerd worden dat één van de dingen die mensen leuk vinden aan WordPress de enorme bibliotheek met plugins van derden is. Maar met 56.000+ gratis plugins op WordPress.org alleen al en nog eens duizenden elders, kan het moeilijk zijn om die ene plugin te vinden die je nodig hebt. Over een speld in een hooiberg gesproken...

We proberen alleen informatie te delen over dingen die we dagelijks gebruiken. En ja, we gebruiken WordPress plugins op onze site, net als de rest van jullie. Veel van de teamleden van Kinsta ontwikkelen en verkopen zelfs plugins.

## Een groot probleem met WordPress plugins

Een groot probleem met WordPress plugins is het de-installatieproces. Wanneer je een WordPress plugin of thema installeert, slaat het gegevens op in de database. Het probleem is dat wanneer je een plugin verwijdert met behulp van één van de standaardmethoden, het meestal tabellen en rijen in je database achterlaat. Na verloop van tijd kan dit tot een heleboel gegevens leiden en zelfs je site beginnen te vertragen. In ons voorbeeld hebben we de Wordfence security plugin verwijderd, en het liet 24 tabellen achter in onze database (zoals hieronder te zien is). Het is nog erger als de gegevens in je wp\_options-tabel staan.

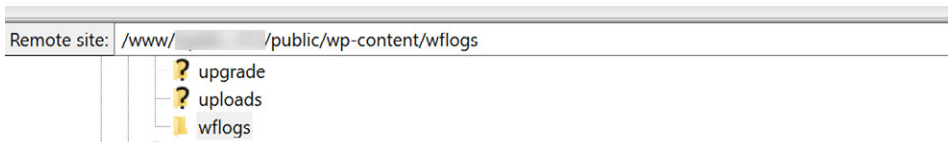
<input type="checkbox"/>	wp_wfBadLeechers	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	latin1_swedish_ci	16 KIB
<input type="checkbox"/>	wp_wfBlockedIPLog	★	Browse	Structure	Search	Insert	Empty	Drop	35	InnoDB	utf8_general_ci	16 KIB
<input type="checkbox"/>	wp_wfBlocks	★	Browse	Structure	Search	Insert	Empty	Drop	32	InnoDB	utf8_general_ci	32 KIB
<input type="checkbox"/>	wp_wfBlocksAdv	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	16 KIB
<input type="checkbox"/>	wp_wfConfig	★	Browse	Structure	Search	Insert	Empty	Drop	156	InnoDB	utf8_general_ci	16 KIB
<input type="checkbox"/>	wp_wfCrawlers	★	Browse	Structure	Search	Insert	Empty	Drop	10	InnoDB	latin1_swedish_ci	16 KIB
<input type="checkbox"/>	wp_wfFileMods	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	16 KIB
<input type="checkbox"/>	wp_wfHits	★	Browse	Structure	Search	Insert	Empty	Drop	1,907	InnoDB	latin1_swedish_ci	1.3 MiB
<input type="checkbox"/>	wp_wfHoover	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	32 KIB
<input type="checkbox"/>	wp_wfIssues	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	16 KIB
<input type="checkbox"/>	wp_wfKnownFileList	★	Browse	Structure	Search	Insert	Empty	Drop	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 K1B
<input type="checkbox"/>	wp_wfNet404s	★	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8_general_ci	32 K1B
<input type="checkbox"/>	wp_wfNotifications	★	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8_general_ci	16 K1B
<input type="checkbox"/>	wp_wfPendingIssues	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	16 K1B
<input type="checkbox"/>	wp_wfReverseCache	★	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	latin1_swedish_ci	16 K1B
<input type="checkbox"/>	wp_wfScanners	★	Browse	Structure	Search	Insert	Empty	Drop	618	InnoDB	latin1_swedish_ci	64 K1B
<input type="checkbox"/>	wp_wfSNIPCache	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	64 K1B
<input type="checkbox"/>	wp_wfStatus	★	Browse	Structure	Search	Insert	Empty	Drop	40	InnoDB	utf8_general_ci	48 K1B
<input type="checkbox"/>	wp_wfThrottleLog	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	32 K1B
<input type="checkbox"/>	wp_wfVulnScanners	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	latin1_swedish_ci	16 K1B

WordFence tables

En naast de database laten veel plugins ook nog eens extra mappen en bestanden achter. Onze ervaring is dat dit vaak gebeurt met beveiligings- en caching-plugins die extra mappen aanmaken om te kunnen loggen. Bijvoorbeeld, nadat de Wordfence plugin was verwijderd, bleef er een “wflogs” map achter in onze wp-content directory. Wordfence dient hier slechts als een voorbeeld, de meeste plugins en thema’s op de markt werken op deze manier.



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

WordFence logs

## Waarom doen ontwikkelaars dit?

Dus je vraagt je waarschijnlijk af, waarom hebben ontwikkelaars geen opruim-opties als je een plugin deactiveert en verwijdert? Nou, dat hebben ze wel. Maar, hier zijn een paar redenen waarom ze waarschijnlijk niet zo voor de hand liggend zijn.

- 1. Ze willen de instellingen voor de gebruiker behouden** Als je een WordPress plugin verwijdert en besluit deze later opnieuw te proberen, zullen al je instellingen en gegevens er nog steeds zijn. Hoewel dit super handig is, is het niet de meest efficiënte manier.
- 2. Ze geven niet om prestaties.** Sommige ontwikkelaars zouden kunnen beargumenteren dat het achterlaten van tabellen geen invloed heeft op de prestaties. Maar stel je voor dat een site in de loop van tien jaar honderden plugins heeft gebruikt, die mogelijk duizenden rijen of tabellen hebben gegenereerd. Database-queries hebben een significante impact op de prestaties van je WordPress site, en plugins kunnen veel van deze queries doen als de ontwikkelaar niet opgelet heeft. Over het algemeen zou een goed geschreven plugin alleen de tabellen of rijen waaraan hij gekoppeld is moeten opruimen, maar dit is niet altijd het geval. We hebben dit uit de eerste hand gezien bij Kinsta, lange database queries die een site langzamer maken vanwege onnodige data in de wp\_options tabel die is achtergelaten.
- 3. Ze hebben een fout gemaakt.** In het WordPress plugin-handboek staat zelfs dat “minder ervaren ontwikkelaars soms de fout maken om de deactiveringshook voor dit doel te gebruiken”.

Het goede nieuws? Er zijn manieren om op te ruimen en je op een goede manier te ontdoen van een plugin. Links naar de volgende uitgebreide tutorials zijn beschikbaar op onze website <https://kinsta.com/nl/leren/wordpress-sneller-maken/>.

- Hoe verwijder je een WordPress plugin (op de juiste manier)
- Hoe kan je handmatig tabellen opruimen die achtergelaten zijn?

HOOFDSTUK 07:

# Optimale WordPress instellingen



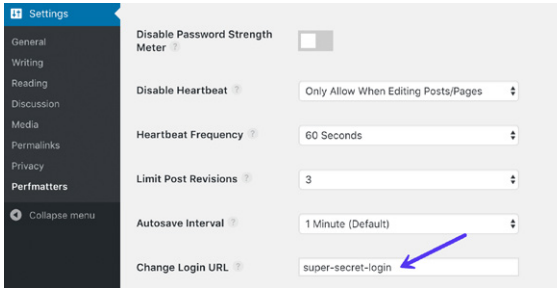
**Nu naar de optimale WordPress instellingen. Hier zijn een paar wijzigingen die je kunt maken om je WordPress site te versnellen. Veel van deze wijzigingen zijn zeer subtiel, maar alles helpt!**

## **Wijzig de WordPress login URL**

Standaard is de login URL van je WordPress site `domein.nl/wp-admin/`. Eén van de problemen hiermee is dat alle bots, hackers en scripts die er zijn dit ook weten. Door de URL te wijzigen, kan je je site minder doelwit maken, beter beschermen tegen brute force attacks, en de bandbreedte die de bots die deze URL herhaaldelijk opvragen gebruiken, verminderen.

Het wijzigen van je WordPress login URL kan ook helpen bij het voorkomen van veel voorkomende fouten zoals “429 Too Many Requests”. Dit is geen oplossing voor alle problemen, het is slechts een klein trucje dat je kan helpen om je te beschermen en de belasting van die pagina te verminderen.

Om je WordPress login URL te wijzigen raden wij je aan één van de volgende plugins te gebruiken:



Change WordPress login URL in Perfmatters

- **WPS Hide Login** (Gratis, beschikbaar op [WordPress.org](https://WordPress.org))
- **Perfmatters** (Premium, maar inclusief andere prestatie-optimalisatie-instellingen. Ontwikkeld door een teamlid van Kinsta en beschikbaar op [perfmatters.io](https://perfmatters.io))

## Uitschakelen of aanpassen van plugin en thema updates

Trage WordPress admin dashboards kunnen worden beïnvloed door het netwerk, de locatie van het datacenter en zelfs door PHP versies. Maar een andere factor waar niet veel mensen over praten is de WordPress update checker die op de achtergrond draait. Dit is een geval waar het hebben van veel WordPress plugins en thema's je problemen zou kunnen opleveren. Het team van WeFoster heeft hier een geweldige zin voor bedacht, die het "Third Party Plugin Update Check Syndrome" of TPPUCS noemt.

In wezen is het probleem dat de ingebouwde WordPress update checker achter de schermen een extern GET-verzoek doet (<https://third-party-plugin/update-check.php>). Soms kan dit periodiek of zeer frequent zijn. Als dit vaak gebeurt, kan dit je admin dashboard sterk vertragen.

Dit is meer een probleem met de manier waarop de update checker in WordPress is opgebouwd. Als je last hebt van lange WordPress admin dashboard laadtijden, kan je dit misschien eens proberen; de remedie is om automatische updates uit te schakelen. Waarschuwing: Doe dit alleen als je van plan bent om handmatig te controleren op updates. Veel updates bevatten beveiligings- en bugfixes.

Om updates uit te schakelen, raden wij aan één van de volgende plugins te gebruiken (beide zijn beschikbaar op [WordPress.org](https://WordPress.org)):

- **Disable All WordPress Updates:** Volledig gratis zonder instellingen. Doet wat het moet doen, en doet dat goed.

- **Easy Updates Manager:** Biedt meer controle over selectieve updates. De core-versie is gratis

Je kunt eenvoudig een herinnering instellen, de plugin één keer per week uitschakelen, controleren op updates en de plugin vervolgens opnieuw inschakelen.

## Pingbacks uitschakelen

Een pingback is een geautomatiseerd commentaar dat wordt aangemaakt wanneer een andere blog naar jouw website linkt. Er kunnen ook self-pingbacks worden gemaakt wanneer je een link naar een artikel in je eigen blog plaatst.

Wij raden je aan deze gewoon uit te schakelen, omdat ze waardeloze requests en extra spam op je site genereren. Vergeet niet dat hoe minder verzoeken je WordPress site hoeft te maken, hoe beter, vooral op sites met veel verkeer. Om nog maar te zwijgen van het feit dat een pingback op je eigen website gewoonweg vervelend is. Volg de onderstaande stappen om pingbacks uit te schakelen.

## Stap 1 - Pingbacks van andere blogs uitschakelen

Klik in je WordPress dashboard op “Instellingen → Discussie”. Onder de sectie Discussie-instellingen vink je de optie “Link meldingen van andere blogs (pingbacks en trackbacks) op nieuwe artikelen toestaan” uit.

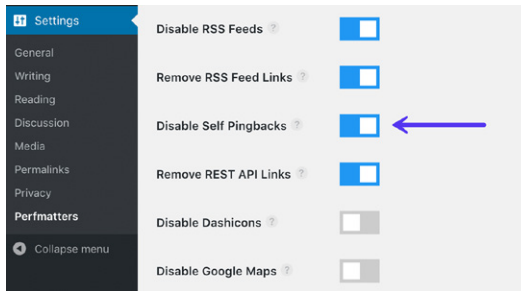
### Discussion Settings

#### 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.)*

## Stap 2 - Self-pingbacks uitschakelen



Disable self-pingbacks with Perfmatters

Voor het uitschakelen van self-pingbacks heb je een paar opties. Je kunt gebruik maken van de gratis No Self Pings plugin (beschikbaar op WordPress.org). Of je kunt een premium plugin gebruiken zoals Perfmatters (beschikbaar op perfmatters.io).

Als alternatief kan je ook self-pingbacks uitschakelen door de volgende code toe te voegen aan het bestand functions.php van je WordPress thema. Waarschuwing, het bewerken van de broncode van een WordPress thema kan je site breken als het niet correct wordt uitgevoerd. Tip, je kunt eenvoudig PHP-fragmenten als deze toevoegen met de gratis Code Snippets plugin (beschikbaar op WordPress.org). Dit betekent dat je je thema nooit hoeft aan te raken.

```
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' );
```

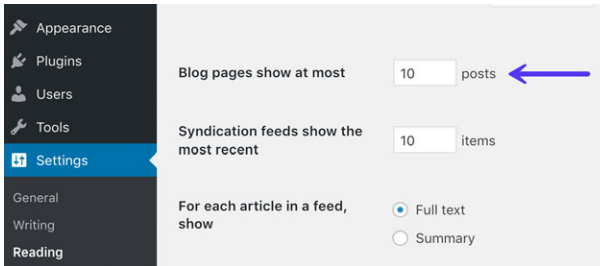
## Beperk de berichten op je Blog-feed

Of je blog feed nu is ingesteld als je homepage of als een andere pagina van je site, je hoeft niet 50 thumbnails tegelijk te laden. Voor degenen die blogs met veel verkeer hebben, is de homepage de belangrijkste pagina van je site, en je wilt dat deze snel laadt. Hoe minder verzoeken en media, hoe beter de prestaties.

Dit is ook precies de reden waarom de paginering is uitgevonden (zie hieronder). Paginering is wat je ziet aan het einde van blog feeds die je in staat stellen om naar de volgende pagina te bladeren. Meestal zijn dit getallen, of ze kunnen “volgende/vorige” gebruiken. Je WordPress thema heeft waarschijnlijk al een aangepaste paginering ingebouwd.



WordPress stelt standaard de limiet voor nieuwe WordPress-installaties op 10, maar we hebben dit zo vaak zien veranderen dat we de tel kwijt zijn. Zorg er dus voor dat je dubbel controleert welke waarde je gebruikt. We raden aan om ergens tussen 8 en 12 te gebruiken. We gebruiken zelf 12 op onze Kinsta blog homepage.



Je kunt deze optie vinden in je WordPress admin dashboard onder “Instellingen / Lezen”. Je kunt dan de waarde voor “Blog pagina’s tonen maximaal” wijzigen.



HOOFDSTUK 08:

# Waarom cache zo belangrijk is



Caching is veruit één van de **belangrijkste en gemakkelijkste manieren om WordPress te versnellen!** Maar voordat we je laten zien hoe je caching kunt gebruiken, is het essentieel om eerst te begrijpen hoe het werkt en te weten welke verschillende soorten caching die beschikbaar zijn.

## Wat is caching?

Elke bezochte webpagina op je WordPress site vereist een verzoek aan de server, verwerking door die server (inclusief databasequeries), en vervolgens een eindresultaat dat door de server naar de browser van de gebruiker wordt gestuurd. Het resultaat is je website, compleet met alle bestanden en elementen die nodig zijn.

Zo heb je bijvoorbeeld een header, afbeeldingen, een menu en een blog. Aangezien de server al deze verzoeken moet verwerken, duurt het enige tijd voordat de volledige webpagina aan de gebruiker wordt geleverd - vooral bij complexe of grotere websites.

Dat is waar een WordPress caching plugin kan helpen! Caching instrueert de server om een aantal bestanden op te slaan op schijf of in RAM, afhankelijk van de configuratie. Zo kan het zich dezelfde inhoud herinneren en dupliceren die het in het verleden al eens

heeft gemaakt. In principe vermindert het de hoeveelheid werk die nodig is om een paginaweergave te genereren. Als gevolg daarvan **laden je webpagina's veel sneller, rechtstreeks vanuit de cache.**

Enkele andere voordelen van caching zijn:

- **Je server gebruikt minder bronnen** - Dit resulteert in snelheid, omdat minder gebruik van bronnen zorgt voor een snellere site. Het legt ook minder druk op je server. Dit is zeer belangrijk als het gaat om zeer dynamische sites, zoals lidmaatschapssites, en het bepalen van wat je wel en niet met cache kunt gebruiken.
- **Je ziet lagere TTFB** - Caching is één van de makkelijkste manieren om je TTFB te verlagen. In feite is het zo dat in onze tests caching de TTFB tot 90% verlaagt!

## Soorten caching

Als het gaat om soorten caching, zijn er twee verschillende soorten die vaak worden gebruikt:

1. Caching op serverniveau
2. Caching met een plugin

### 1. Caching op serverniveau

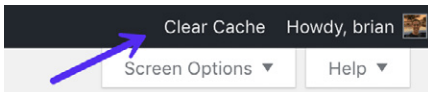
Caching op serverniveau is veruit één van de gemakkelijkste manieren om de eindgebruiker te bereiken. Dit betekent dat de WordPress hostingprovider het voor je afhandelt. Bij Kinsta maken we gebruik van de volgende **vier soorten cache**, die allemaal automatisch op software- of serverniveau worden uitgevoerd:

- Bytecode cache
- Object cache
- Page cache
- CDN cache

Dit betekent dat je je geen zorgen hoeft te maken over het knoeien met ingewikkelde en verwarrende caching plugins. Je kunt stoppen met googelen naar de “beste caching plugins” en je richten op meer productieve taken.

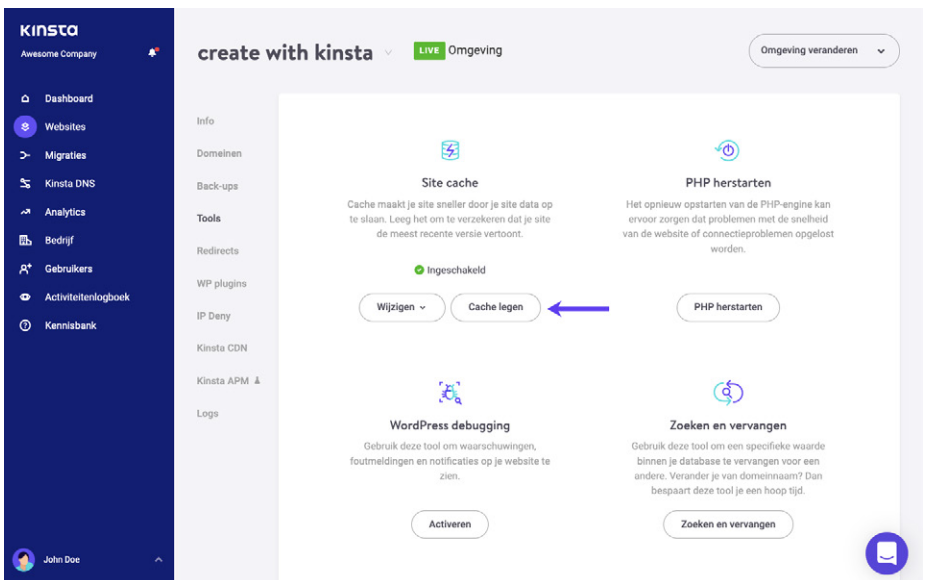
De paginacache is zo geconfigureerd dat deze zonder extra configuratie werkt met WordPress. Je hoeft niets te doen! Start gewoon je WordPress site en de paginacaching zal werken.

We hebben ook caching regels voor e-commerce sites zoals WooCommerce en Easy Digital Downloads. Standaard zijn er bepaalde pagina's, zoals de winkelwagen pagina, my-account en checkout, uitgesloten van caching. Gebruikers omzeilen automatisch de cache wanneer de `woocommerce_items_in_cart` cookie of `edd_items_in_cart` worden gedetecteerd om te zorgen voor een soepel en in-sync afrekenproces.



Je kunt de cache van je WordPress site op elk gewenst moment eenvoudig legen via de link in de WordPress admin-werkbalk.

Het is ook geïntegreerd in ons MyKinsta-dashboard. Klik op Tools en klik op “Clear Cache”.



## 2. Caching met een plugin

Als je hostingprovider geen cache aanbiedt, kun je gebruik maken van een WordPress caching plugin. Op basis van onze ervaring bevelen wij één van de volgende aan:

- **WP Rocket** (premium, verkrijgbaar bij wp-rocket.me)
- **Cache Enabler** (gratis, beschikbaar op WordPress.org)
- **W3 Total Cache** (gratis, beschikbaar op WordPress.org)

Bij Kinsta ondersteunen we WP Rocket volledig! We staan meestal geen caching plugins toe in onze omgeving omdat ze in strijd zijn met onze ingebouwde caching oplossing. Echter, vanaf WP Rocket 3.0 zal hun pagina caching functionaliteit automatisch worden uitgeschakeld wanneer ze op Kinsta servers draaien.

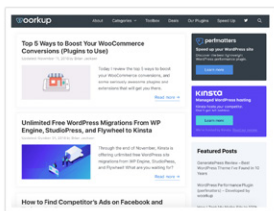
Hierdoor kunnen Kinsta-cliënten gebruik maken van onze snelle caching op serverniveau, en ook profiteren van de fantastische optimalisatiemogelijkheden die WP Rocket te bieden heeft.

## Geen caching vs. caching

Hoeveel helpt caching? Het bewijs is duidelijk.

We hebben een aantal snelheidstesten uitgevoerd met Kinsta's server caching, zodat je het verschil kunt zien, zowel wat betreft de totale snelheid als de TTFB.

## Geen caching

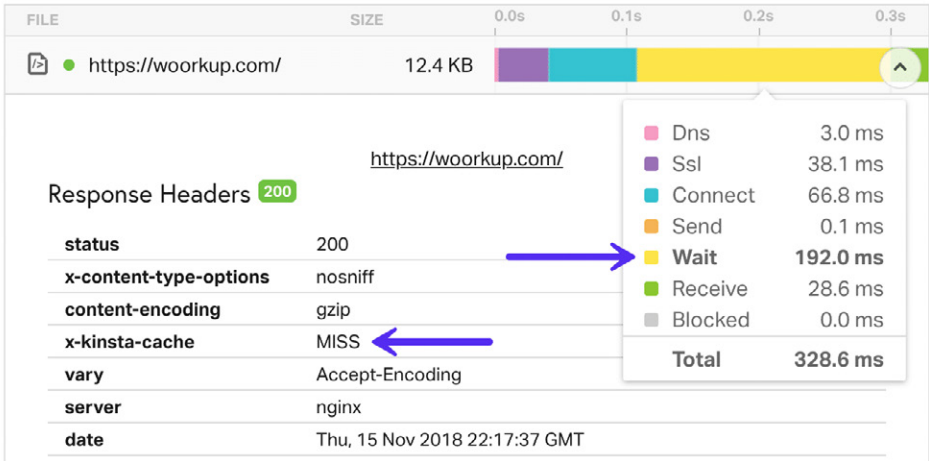


Performance grade <b>B 89</b>	Page size <b>230.6 KB</b>
Load time <b>507 ms</b>	Requests <b>40</b>

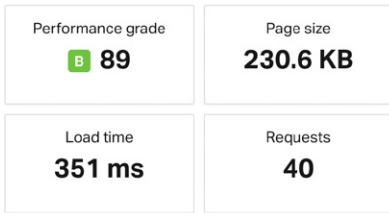
We hebben eerst vijf tests op Pingdom uitgevoerd zonder ingeschakelde cache en namen het gemiddelde.

# TTF zonder caching

Het is ook belangrijk om het verschil in TTFB zonder en met caching op te merken. TTFB in Pingdom wordt vertegenwoordigd door de gele “wachtende” balk. Zoals je kunt zien is de TTFB zonder caching 192 ms. Je kunt zien dat het niet vanuit de cache komt, omdat de x-kinsta-cache header een MISS toont.



# Met caching geactiveerd

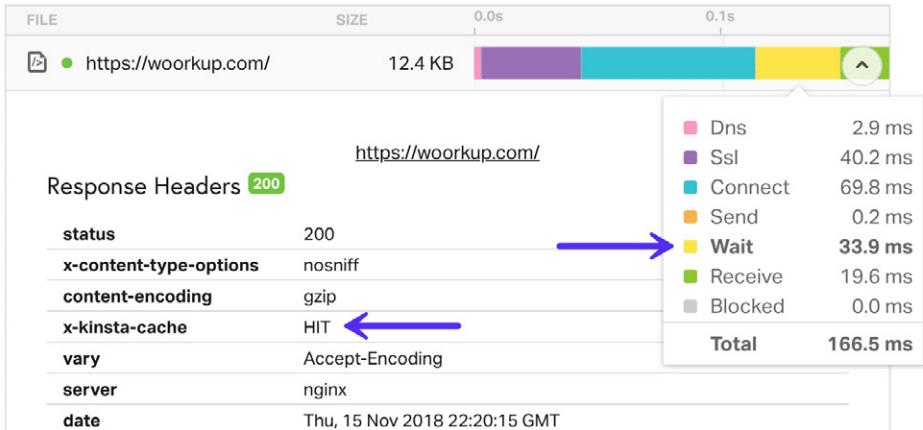


We hebben vervolgens server caching ingeschakeld en vijf tests op Pingdom uitgevoerd en het gemiddelde genomen.

Zoals je kunt zien is door de **server caching de laadtijd van onze pagina's met 33,77% verminderd!** En dat zonder extra werk. Deze door ons geteste site is ook redelijk geoptimaliseerd, dus grotere niet-geoptimaliseerde sites zullen ongetwijfeld nog grotere verschillen zien.

# TTFB met caching ingeschakeld

Als we nu eens kijken naar de TTFB met caching ingeschakeld, kunnen we zien dat het onder de 35 ms is. Je kunt zien dat het uit de cache komt omdat de de x-kinsta-cache header een HIT toont.



CDN-cache is net zo belangrijk als de cache van je WordPress host. We kijken verderop uitgebreider naar CDN's.

*WordPress caching kan de laadtijd van je pagina's gemakkelijk met meer dan 33% verminderen!*

## Problemen met caching en lidmaatschapssites

Lidmaatschapssites bevatten veel **ongecachte inhoud** en pagina's die voortdurend veranderen. Dingen zoals de loginpagina voor leden van de community (die constant opgevraagd kan worden, afhankelijk van de grootte van de site), afrekenpagina's voor digitale goederen of cursussen, en discussiefora zijn veel voorkomende uitdagingen, omdat deze meestal niet in de cache kunnen worden geplaatst.

Maar daar houdt het niet op. Op standaard WordPress sites wordt het WordPress dashboard ook niet in de cache geplaatst voor “**ingelogde**” gebruikers. Dit is prima als je maar een paar auteurs en admins hebt, maar als je ineens duizenden leden hebt die het dashboard gebruiken, veroorzaakt dit direct problemen met de prestaties, omdat niets van het dashboard vanuit de cache op de server kan komen. Dit betekent dat je de kracht en de architectuur achter de schermen nodig hebt om het te ondersteunen. Shared hostingproviders zullen in deze omstandigheden meestal het onderspit delven.

## Object caching voor dynamische sites

Als het gaat om WordPress lidmaatschapssites, zijn de gebruikelijke cache opstellingen meestal niet genoeg, omdat ze er niet altijd optimaal gebruik van maken. Dit is waar **object caching een belangrijke rol speelt**.

Object cache slaat de resultaten van database-queries op, zodat de volgende keer dat een bepaald deel van de gegevens nodig is, deze vanuit de cache kunnen worden geleverd zonder dat de database hoeft te worden geraadpleegd. Dit versnelt de uitvoering van PHP en vermindert de belasting van je database. Dit is extreem belangrijk met lidmaatschapssites! Met WordPress kan je object caching op een aantal verschillende manieren implementeren:

1. Een caching-oplossing van derden zoals W3 Total Cache
2. **Redis** (aanbevolen)
3. Memcached

Wij bieden Redis aan als add-on bij Kinsta, zodat je ten volle kunt profiteren van de snelle object caching voor jouw lidmaatschapssites.

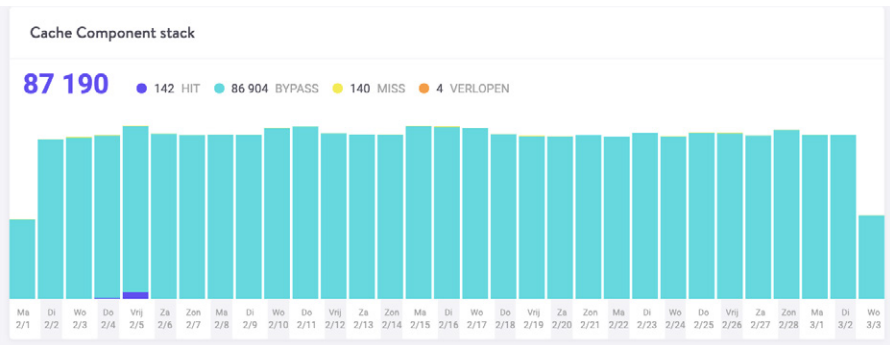
## Het analyseren van cache

Weet je nog dat we het over de `x-kinsta-cache` header hadden hierboven? Afhankelijk van je hostingprovider of cache-oplossing kan de header iets anders heten. Elke keer als er een verzoek wordt gedaan vanaf je WordPress site heeft die header een waarde, zoals HIT, BYPASS, MISS, en EXPIRED. Zo kan je zien hoe je cache presteert.

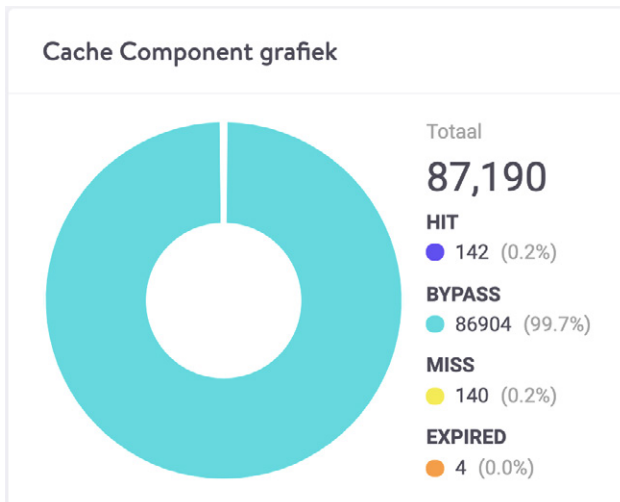


Het verhogen van de cache hit ratio van je WordPress site is belangrijk omdat je zoveel mogelijk van je site uit de cache wilt halen. Bij Kinsta kan je de gegevens in onze MyKinsta analytics tool en de Kinsta cache logs analyseren om te bepalen of er cache BYPASSING GET verzoeken zijn die in de cache kunnen worden geplaatst of POST verzoeken die kunnen worden geëlimineerd.

De cache component stack (zoals hieronder weergegeven) laat je de status van elk verzoek zien, of het nu een HIT, BYPASS, MISS of EXPIRED was. Je kunt de gegevens filteren op de afgelopen 24 uur, 7 dagen of 30 dagen.



De cache component grafiek geeft je een blik op je caching ratio's. Hoe meer aanvragen er vanuit de cache worden verwerkt, hoe beter. Zoals je kunt zien in het voorbeeld hieronder, heeft deze WordPress site een HIT cache ratio van 96,2%. Wat erg goed is!

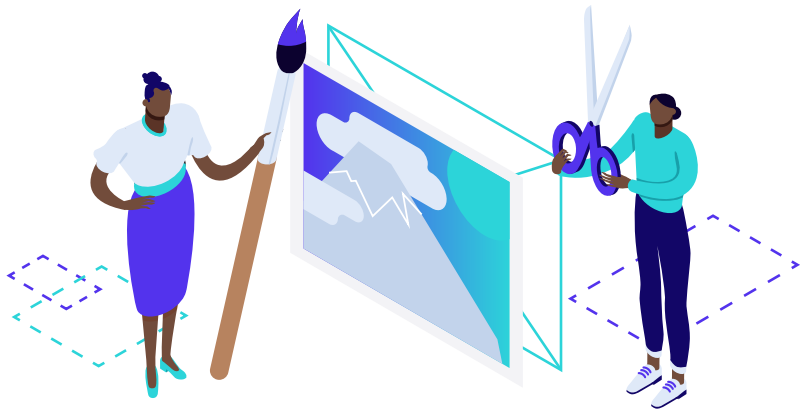


De top cache bypasses sectie laat je zien welke verzoeken niet worden bediend vanuit de cache. Typisch kunnen dit CRON-jobs zijn, admin-ajax verzoeken, e-commerce afrekenpagina's, query strings, UTM-parameters, enz.

PAID	AANVRAGEN
/wp-cron.php?server_triggered_cronjob	40 039
/?password-protected-login&redirect_to=http%3A%2F%2Fcreatingwithkinsta.temp513.kinsta.cloud%2F%3Fkinsta-monitor	21 521
/?password-protected-login&redirect_to=https%3A%2F%2Fcreatingwithkinsta.temp513.kinsta.cloud%2F%3Fkinsta-monitor	21 520
/	1 012
/robots.txt	413
?ezole_domain_verify=1	178
/2020/03/31/sample-post/	101
/sample-post/	100
/wp-login.php	85
/wordpress-seo/	81

HOOFDSTUK 09:

# Afbeeldings- optimalisatie is een must

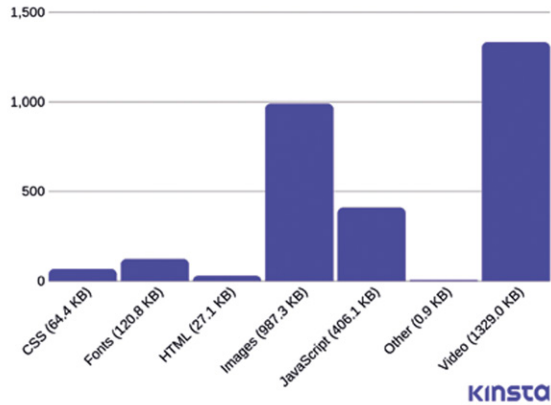


**Afbeeldingsoptimalisatie is een ander eenvoudig principe dat je kunt uitvoeren en dat een aanzienlijke impact heeft op de totale laadtijd van je pagina. Dit zou geen keuze moeten zijn, maar verplicht voor elke site!**

Grote afbeeldingen vertragen je webpagina's, wat een minder dan optimale gebruikerservaring oplevert. Het optimaliseren van afbeeldingen is het proces van het verkleinen van hun bestandsgrootte, met behulp van een plugin of script, die op hun beurt de laadtijd van de pagina versnellen. Lossy en lossless compressie zijn twee veelgebruikte methoden.

Volgens HTTP Archive maken afbeeldingen, sinds augustus 2019, gemiddeld **34% uit van het totale gewicht van een webpagina**. Dus na video's, die veel moeilijker te optimaliseren zijn, zijn beelden veruit de eerste plaats waar je moet beginnen! Het is belangrijker dan JavaScript, CSS en lettertypes. En ironisch genoeg is een goede workflow voor afbeeldingsoptimalisatie één van de eenvoudigste dingen om te implementeren, maar veel website eigenaren zien dit over het hoofd.

## Gemiddelde bytes per pagina (KB)



Gemiddeld 54% van het totale gewicht van een pagina in december 2017 bestond uit afbeeldingen. Het lijkt er dus op dat het web als geheel beter wordt in afbeeldingsoptimalisatie! Maar 34% is nog steeds een getal dat niet kan worden genegeerd. Als je geen videocontent op je website hebt, zijn afbeeldingen waarschijnlijk nog steeds je nummer 1 pijnpunt voor het gewicht van de pagina's.

*Afbeeldingen maken gemiddeld 34% van het totale gewicht van een webpagina uit. Optimaliseer ze!*

## Het vinden van het evenwicht (bestandsgrootte en kwaliteit)

Het primaire doel bij het opslaan van je afbeeldingen is het vinden van de **balans tussen de laagste bestandsgrootte en acceptabele kwaliteit**. Er is meer dan één manier om bijna al deze optimalisaties uit te voeren. Eén van de meest elementaire manieren is om ze te comprimeren voordat ze naar WordPress worden geüpload. Meestal kan dit gedaan worden in een tool zoals Adobe Photoshop of Affinity Photo. Of door gebruik te maken van de nieuwe online Squoosh app van Google. Deze taken kunnen echter ook automatisch worden uitgevoerd met behulp van plugins, waar we hieronder meer over zullen vertellen.

De twee belangrijkste dingen om te overwegen zijn het **bestandsformaat** en het **type compressie dat je gebruikt**. Door de juiste combinatie van bestandsformaat en compressietype te kiezen kunt je een afbeelding tot wel 5 keer verkleinen. Je zult met elke afbeelding of elk bestandsformaat moeten experimenteren om te zien wat het beste werkt.

Voordat je begint met het wijzigen van je afbeeldingen, moet je ervoor zorgen dat je het beste bestandstype hebt gekozen. Er zijn verschillende soorten bestanden die je kunt gebruiken:

- **PNG** – produceert beelden van hogere kwaliteit, maar heeft ook een grotere bestandsgrootte. Wordt gemaakt als een lossless beeldformaat, hoewel het ook lossy kan zijn.
- **JPEG** – maakt gebruik van lossy en lossless optimalisatie. Je kunt het kwaliteitsniveau aanpassen voor een goede balans tussen kwaliteit en bestandsgrootte.

Idealiter gebruik je JPEG (of JPG) voor afbeeldingen met veel kleur en PNG voor eenvoudige afbeeldingen.

Hoe zit het met GIF's? Geanimeerde GIF's zijn altijd leuk, maar ze verslechteren de webprestaties. Veel GIF's zijn meer dan 1 MB groot. We raden aan deze te bewaren voor sociale media en Slack. Als er één is waar je niet zonder kunt leven in je blogpost, moet je ze altijd zwaar comprimeren.

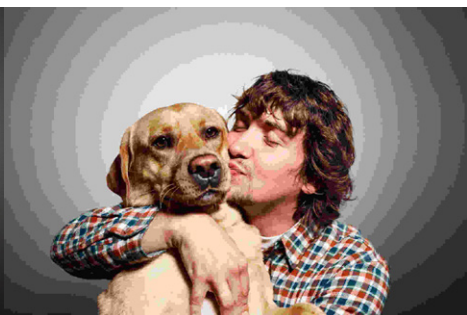
## Compressie kwaliteit vs. grootte

Hier is een voorbeeld van wat er kan gebeuren als je een beeld te veel comprimeert. De eerste gebruikt een zeer lage compressiegraad, wat resulteert in de hoogste kwaliteit (maar grotere bestandsgrootte). De tweede gebruikt een zeer hoge compressiegraad, wat resulteert in een afbeelding van zeer lage kwaliteit (maar kleinere bestandsgrootte). Opmerking: De originele afbeelding is 2,06 MB.

Zoals je kunt zien is de eerste afbeelding hierboven 590 KB. Dat is behoorlijk groot voor één foto! Het is over het algemeen het beste als je het totale gewicht van een webpagina onder de 1 of 2 MB kunt houden. 590 KB zou al een vierde van dat gewicht zijn. De tweede foto ziet er verschrikkelijk uit, maar dan is het bestand maar 68 KB. Wat je wilt doen is een middenweg vinden tussen de compressiegraad (kwaliteit) en de bestandsgrootte.



Lage compressie (hoge kwaliteit) JPG - 590 KB



Hoge compressie (lage kwaliteit) JPG - 68 KB



Medium compressie (grote kwaliteit) JPG - 151 KB

Dus we hebben de foto opnieuw geoptimaliseerd met een gemiddelde compressiegraad, en zoals je hiernaast kunt zien, ziet de kwaliteit er nu goed uit, en de bestandsgrootte is 151 KB, wat acceptabel is voor een foto met een hoge resolutie. Dit is bijna 4x kleiner dan de originele foto met lage compressie. We proberen de meeste van onze foto's onder de 100 KB te houden voor de beste prestaties.

## Lossy vs. Lossless optimalisatie

Het is ook belangrijk om te begrijpen dat er twee soorten compressie zijn die je kunt gebruiken, lossy en lossless.

Lossy compressie houdt in dat **een deel van de gegevens** in je afbeelding **wordt verwijderd**. Dit betekent dat je misschien een degradatie ziet (vermindering van de kwaliteit of wat sommigen pixelvorming noemen). Je moet dus voorzichtig zijn met hoeveel je je afbeelding reduceert. Niet alleen vanwege de kwaliteit, maar ook omdat je het proces niet kunt omkeren. Natuurlijk is één van de grote voordelen van lossy compressie en waarom het één van de meest populaire compressiemethoden is, dat je **de bestandsgrootte aanzienlijk kunt verkleinen**.



Lossless compressie, in tegenstelling tot lossy, **vermindert de kwaliteit van het beeld niet**. Hoe is dit mogelijk? Meestal wordt dit gedaan door onnodige metadata te verwijderen (automatisch gegenereerde gegevens die worden geproduceerd door het apparaat dat het beeld vastlegt). Het grootste nadeel van deze methode is echter dat je **geen significante vermindering van de bestandsgrootte zult zien**. Met andere woorden, het zal in de loop van de tijd veel schijfruimte in beslag nemen.

Je zal moeten experimenteren met wat het beste voor je werkt. Voor de meerderheid van de gebruikers **raden we aan om lossy compressie te gebruiken** omdat je een beeld gemakkelijk kan comprimeren tot meer dan 70% (soms zelfs tot meer dan 90%!) zonder veel kwaliteitsverlies. Vermenigvuldig dit met 15 afbeeldingen op een pagina en het zal een belangrijke rol spelen bij het verminderen van de laadtijd van je site.

## Plugins om afbeeldingen te comprimeren

Het goede nieuws is dat er een aantal geweldige WordPress plugins zijn die je kunt gebruiken om het hele proces te automatiseren. Hier zijn enkele plugins die we aanbevelen (die allemaal beschikbaar zijn op WordPress.org):

- **Imagify** (lossy en lossless - optimaliseert beelden op een externe server)
- **WP Smush** (lossy en lossless - optimaliseert beelden op een externe server)
- **Optimole** (lossy en lossless - optimaliseert beelden op een externe server)
- **EWWW Cloud** (lossy en lossless - optimaliseert beelden op een externe server)
- **ShortPixel** (lossy en lossless - optimaliseert beelden op een externe server)

Het belangrijkste bij het kiezen van een plugin voor afbeeldingsoptimalisatie is



om **er een te gebruiken die de beelden extern comprimeert en optimaliseert**. Dit vermindert de belasting van je site. Alle bovenstaande plugins doen dit.

We gebruiken de Imagify plugin zelf op onze Kinsta website. Deze comprimeert de afbeeldingen automatisch wanneer we ze uploaden naar de WordPress mediabibliotheek. Zo hoeven we ons nergens zorgen over te maken. Na verloop van tijd kan je een goed gevoel krijgen voor het compressieniveau van de afbeeldingen die je wilt gebruiken. Imagify heeft de volgende niveaus: Normaal, Agressief en Ultra.

**We gebruiken zelf Aggressief** en zien meestal een **besparing van 60-70%**, afhankelijk van de afbeelding. Let op: we gebruiken veel meer PNG's dan JPEG's omdat de meeste van onze beelden iconen en illustraties zijn, geen foto's.

The screenshot shows two entries in the WordPress media library. The first entry is for 'bedrock-trellis.png' (6.16 KB, 74.68% saving) and the second is for 'kinsta-architecture-2.png' (82.47 KB, 62.41% saving). Both entries show a 'VIEW DETAILS' button and a 'Restore Original' button. Blue arrows point to the 'VIEW DETAILS' buttons.

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

Hoeveel sneller zal je WordPress site zijn als je gebruik maakt van afbeeldingscompressie? Het hangt allemaal af van de grootte van je originele afbeeldingen en hoe groot ze zijn na het comprimeren. We hebben enkele snelheidstesten uitgevoerd en vastgesteld dat een goede oplossing voor het comprimeren van afbeeldingen **de laadtijd van de pagina's met meer dan 80% kan verminderen!**

## Lazy Loading

Als je veel foto's hebt, kun je overwegen om lazy loading toe te passen. Dit is een optimalisatietechniek die zichtbare inhoud laadt, maar het downloaden en renderen van inhoud die onder de vouw verschijnt, vertraagt.

Lazy loading werkt zo:

- De browser bouwt de webpagina DOM zonder afbeeldingen te downloaden en video's voor te laden.
- JavaScript wordt gebruikt om te bepalen welke afbeeldingen moeten worden gedownload en welke video's moeten worden voorgeladen op basis van de inhoud die in eerste instantie zichtbaar is wanneer de pagina wordt geladen. Deze afbeeldingen en video's worden gedownload en voorgeladen.
- Het downloaden en renderen van extra video's wordt vertraagd totdat een bezoeker van de site naar beneden scrollt en er extra inhoud in beeld komt.

Het eindresultaat is dat afbeeldingen niet worden gedownload en video's niet worden geladen totdat ze daadwerkelijk nodig zijn. Dit kan een aanzienlijke prestatieverhoging betekenen voor sites die veel hoge-resolutie afbeeldingen en embedded video's bevatten.

Gelukkig zijn er een heleboel geweldige gratis WordPress plugins die het mogelijk maken om lazy loading vrij eenvoudig te maken (ze zijn allemaal beschikbaar op WordPress.org):

- **a3 Lazy Load**
- **Lazy Load by WP Rocket**
- **Lazy Load for Videos**

## Aanvullende tips voor afbeeldingsoptimalisatie

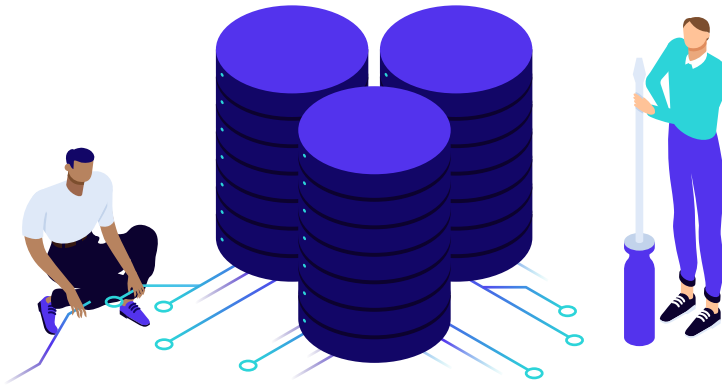
Hier zijn een paar laatste tips voor afbeeldingsoptimalisatie:

- De dagen van het uploaden van beelden die alleen op de breedte van de kolom of DIV staan, zijn voorbij. Reactieve beelden werken out of the box in WordPress (sinds versie 4.4) en zullen automatisch kleinere beeldformaten weergeven aan mobiele gebruikers.
- SVG's kunnen een ander geweldige alternatief zijn voor het gebruik van beelden. Alle handgetekende illustraties die je ziet op de Kinsta website zijn SVG's (vectoren). SVG's zijn meestal een stuk kleiner in bestandsgrootte, maar niet altijd.

- Gebruik pictogram lettertypen in plaats van tekst in afbeeldingen te plaatsen - ze zien er beter uit als ze geschaald zijn en nemen minder ruimte in beslag. En als je een lettertype-generator gebruikt, kan je ze nog meer optimaliseren. We hebben de grootte van ons pictogram lettertype-bestand met **maar liefst 97,59%** verkleind met behulp van een lettertype-generator.
- Misschien wil je ook eens kijken naar WebP, een nieuw afbeeldingsformaat van Google dat snel aan populariteit wint.

HOOFDSTUK 10:

# Tuning van je database



Hieronder vind je enkele tips over hoe je je WordPress database kunt optimaliseren. Net als een auto heeft je database onderhoud nodig omdat deze na verloop van tijd vervuild kan raken.

Vooraf lidmaatschapssites maken het lastig, omdat ze meestal **complexere query's genereren**, die op hun beurt weer extra vertraging in het ophalen van de informatie uit de MySQL-database met zich meebrengen. Veel van dit is te wijten aan alle extra bewegende delen en grote hoeveelheden data die sites zoals deze hebben. Dit kan ook worden veroorzaakt door sites die sterk afhankelijk zijn van zoekopdrachten voor navigatie of het gebruik van `WP_Query`.

En niet te vergeten, je hebt ook grote hoeveelheden gelijktijdige gebruikers die continu de database opvragen.

## Gebruik de InnoDB MySQL Storage Engine

Veel oudere sites gebruiken nog steeds de MyISAM storage engine in hun database. InnoDB heeft de afgelopen jaren bewezen beter te presteren en betrouwbaarder te zijn.

*InnoDB is als synthetische olie,  
terwijl MyISAM genoeg neemt met  
de normale versie.*

Hier zijn een paar voordelen van InnoDB ten opzichte van MyISAM:

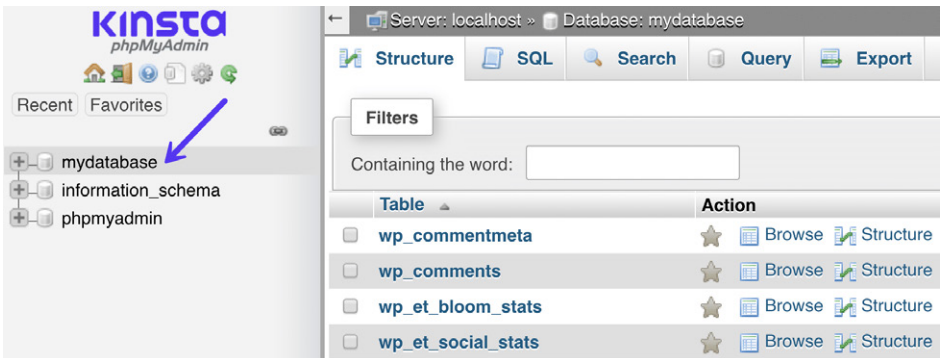
- InnoDB heeft een **vergrendeling op rijniveau**. MyISAM heeft alleen een volledige vergrendeling op tabelniveau. Hierdoor kunnen je queries sneller worden verwerkt.
- InnoDB heeft wat men noemt een referentiële integriteit die betrekking heeft op het ondersteunen van **vreemde sleutels** (RDBMS) en relatiebeperkingen, MyISAM niet (DMBS).
- InnoDB ondersteunt **transacties**, wat betekent dat je deze kunt vastleggen en terugdraaien. MyISAM kan dat niet.
- InnoDB is betrouwbaarder omdat het gebruik maakt van transactielogboeken voor automatisch herstel. MyISAM heeft dat niet.

Dus nu vraag je je misschien af, gebruik je InnoDB of MyISAM? Als je op een vrij nieuwe WordPress site draait, is de kans groot dat je al gebruik maakt van de InnoDB MySQL storage engine. Maar met oudere WordPress sites wil je misschien een snelle check doen. Sommige sites hebben misschien zelfs MyISAM- en InnoDB-tabellen door elkaar, waarin je verbeteringen kunt zien door ze overal te converteren naar MyISAM.

Volg deze eenvoudige stappen hieronder om het te controleren.

## Stap 1

Log in op phpMyAdmin en klik op je MySQL-database.



The screenshot shows the phpMyAdmin interface for a MySQL database named 'mydatabase'. The left sidebar shows a tree view with 'mydatabase' selected, indicated by a blue arrow. The main area displays a table list with the following columns: 'Table' and 'Action'. The table list contains the following entries:

Table	Action
wp_commentmeta	★ Browse Structure
wp_comments	★ Browse Structure
wp_et_bloom_stats	★ Browse Structure
wp_et_social_stats	★ Browse Structure

## Stap 2

Doe een quick scan of sorteer de kolom “Type” en je kunt zien welke storage engine je tabellen gebruiken. In dit voorbeeld hieronder zie je dat twee van de tabellen nog steeds gebruik maken van MyISAM.

Actions	Type	Storage Engine	Size
Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci 64 KiB
Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci 48 KiB
Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci 64 KiB
Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci 80 KiB
Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci 48 KiB
Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci 32 KiB
Browse Structure Search Insert Empty Drop	113	MyISAM	utf8mb4_unicode_ci 80 KiB
Browse Structure Search Insert Empty Drop	471	MyISAM	utf8mb4_unicode_ci 208 KiB
Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci 48 KiB
Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci 16 KiB

Operations Triggers

Table options

Rename table to: wp\_comments

Adjust privileges

Table comments:

Storage Engine: MyISAM (dropdown menu open showing: CSV, MRG\_MyISAM, SEQUENCE, MyISAM, MEMORY, InnoDB, PERCONA)

Collation:

PACK\_KEYS:

CHECKSUM:

DELAY\_KEY\_WRITE:

AUTO\_INCREMENT: 2890

ROW\_FORMAT: DYNAMIC

Go

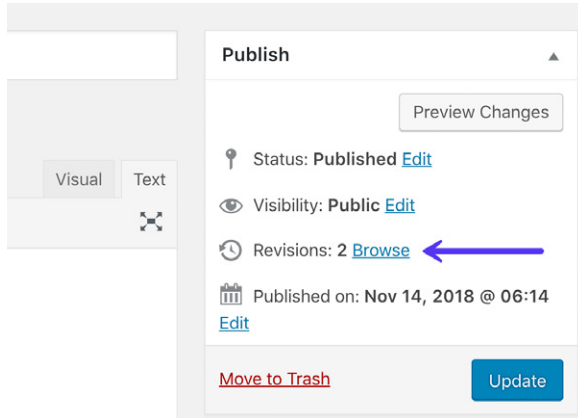
Als je wat gevonden hebt, dan is het waarschijnlijk tijd om ze naar InnoDB over te zetten. We raden altijd aan om je host te benaderen en te vragen of zij dit voor je kunnen doen. Bij Kinsta worden de databasetabellen van elke klant automatisch omgezet naar InnoDB door ons migratieteam.

Maar je kunt ze ook handmatig omzetten met phpMyAdmin. Klik gewoon op de myISAM-tabel, klik op het tabblad “Bewerkingen” en verander de storage engine.

Als je een ontwikkelaar bent, is er nog een ander alternatief om ze te converteren, namelijk met WP-CLI.

# Verwijder en beperk pagina- en bericht revisies

Wanneer je een pagina of bericht opslaat in WordPress, creëert het een zogenaamde revisie. Dit gebeurt in zowel concepten als reeds gepubliceerde berichten die worden geüpdatet. Revisies kunnen nuttig zijn in het geval dat je moet terugkeren naar een eerdere versie van je content van je content.



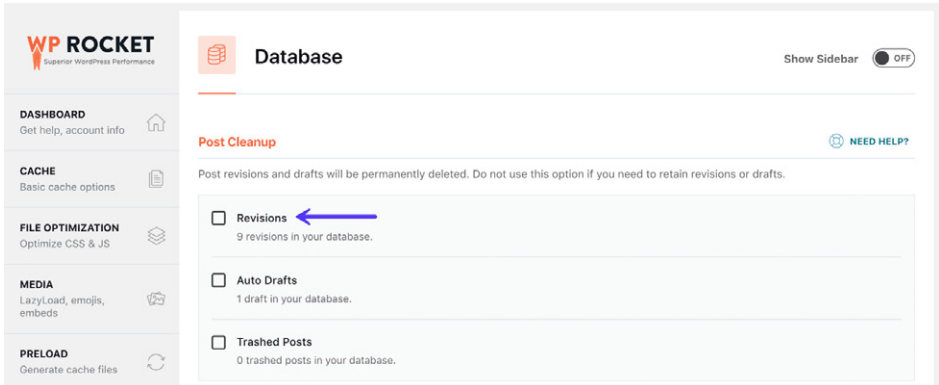
Revisies kunnen echter ook de prestaties van je WordPress site beïnvloeden. Op grote sites kan dit zeer snel oplopen tot duizenden rijen in je database die niet noodzakelijkerwijs nodig zijn. En hoe meer rijen je hebt, hoe groter je database wordt, wat opslagruimte in beslag neemt. Hoewel er voor dit doel indexen zijn gemaakt, hebben we nog steeds gezien dat dit probleem de WordPress sites vertraagt. Er zijn een paar dingen die je kunt doen.

## 1. Oude revisies verwijderen

Als je een oudere WordPress site hebt met veel pagina's en berichten, is het misschien tijd om die oude revisies snel op te schonen en te verwijderen. Je kunt dit doen met MySQL, maar met alle slechte stukjes code die over het web zweven, raden we je aan een back-up te maken van je site en een gratis plugin te gebruiken zoals WP-Sweep.

Een andere favoriete plugin, WP Rocket, heeft ook een database optimalisatie functie om revisies op te ruimen.

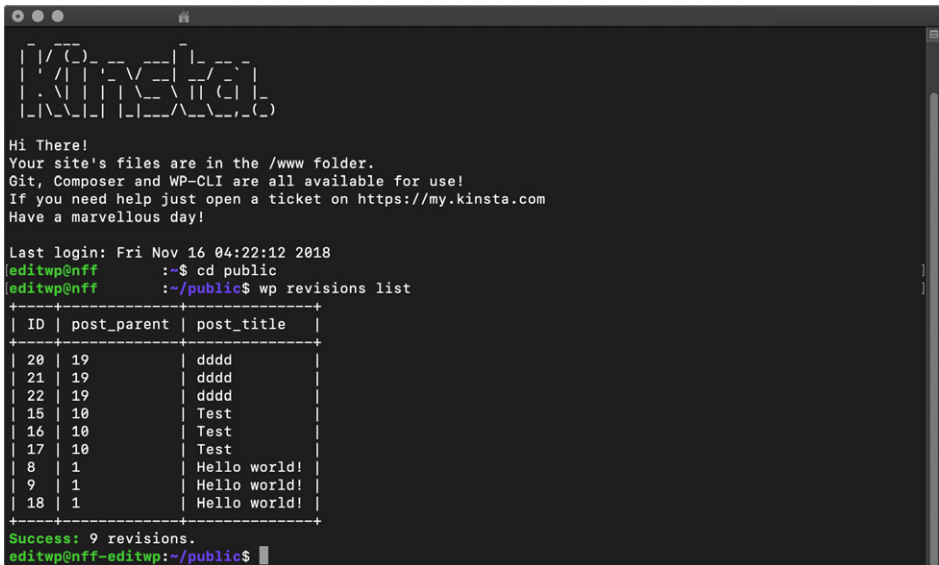




Als je handig bent met WP-CLI, zijn er een paar commando's die je hiervoor kunt gebruiken.

Log in op je server via SSH en voer het volgende commando uit om het aantal revisies dat momenteel in de database staat te krijgen en te zien.

```
wp revisions list
```



Als je een fout krijgt, moet je misschien eerst het `wp-revisions-cli` pakket installeren met het volgende commando:

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

Je kunt dan het volgende commando uitvoeren om de revisies op te schonen:

```
wp revisions clean
```

## 2. Revisies beperken

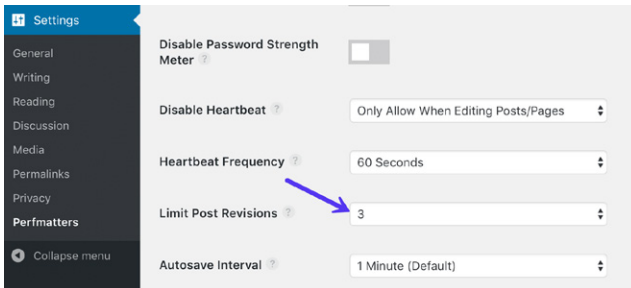
Een andere goede strategie die we bij Kinsta gebruiken is het beperken van het aantal revisies dat per bericht of pagina kan worden opgeslagen. Wanneer je het instelt op zo'n tien, lopen de revisies niet uit de hand, ook niet als je veel bijwerkt.

Om de revisies te beperken, kan je de volgende code toevoegen aan je `wp-config.php` bestand. De onderstaande code moet boven de `'ABSPATH'` worden ingevoegd, anders werkt het niet. Je kunt het nummer wijzigen in hoeveel revisies je wilt bewaren in je database.

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

```
59
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



Limit revisions with Perfmatters

Of je kunt gebruik maken van een premium plugin zoals Perfmatters (beschikbaar op [perfmatters.io](https://perfmatters.io)) om revisies te beperken.

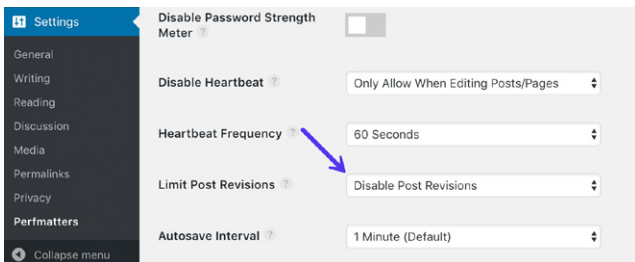
### 3. Revisies uitschakelen

Als laatste optie kan je ook revisies op je site volledig uitschakelen. Als je deze route volgt, raden wij je ten zeerste aan de eerste optie hierboven te volgen om de revisies te verwijderen en daarna uit te schakelen. Op deze manier is je database volledig vrij van alle oude revisies en worden er geen nieuwe revisies meer toegevoegd.

Om de revisies uit te schakelen, kan je de volgende code toevoegen aan je wp-config.php bestand. De onderstaande code moet boven de 'ABSPATH' worden ingevoegd, anders werkt het niet.

```
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
```



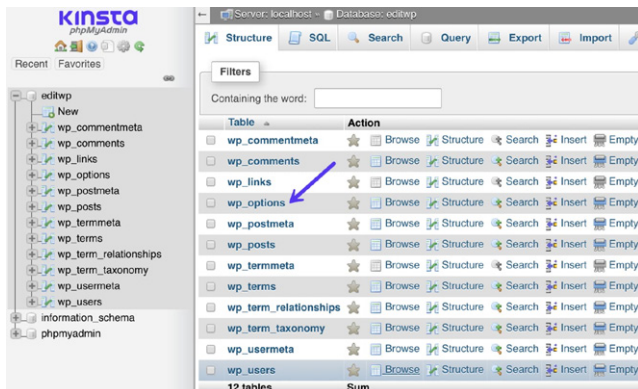
Disable revisions with Perfmatters

Of je kunt een plugin zoals Perfmatters gebruiken om de revisies uit te schakelen.

## Opschonen van je wp\_options tabel en autoloade data

De wp\_options tabel wordt vaak over het hoofd gezien als het gaat om de algemene prestaties van WordPress en de database. Vooral op oudere en grote sites kan dit gemakkelijk de boosdoener zijn voor trage queries op je site als gevolg van autoloade data die wordt achtergelaten door plugins en thema's van derden. Echt, we zien dit elke dag!

De wp\_options tabel bevat allerlei gegevens van je WordPress site zoals:



- Site URL, home URL, admin e-mail, standaard categorie, berichten per pagina, tijdsindeling, enz.
- Instellingen voor plugins, thema's, widgets
- Tijdelijke gegevens in de cache

Deze tabel bevat de volgende velden (kolommen):

- option\_id
- option\_name
- option\_value
- **autoload** (dit is het veld waar we op moeten letten als het op prestaties aankomt)

+ Options

	option_id	option_name	option_value	autoload
<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

Een van de belangrijke dingen om te begrijpen over de `wp_options` tabel is het **autoload** veld. Dit bevat een ja of nee waarde (vlag). Dit controleert in wezen of het al dan niet wordt geladen door de functie `wp_load_alloptions()`. Autoload data is data die op elke pagina van je WordPress site wordt **geladen**. Net zoals we je hebben laten zien hoe je bepaalde scripts kan uitschakelen voor het laden op je gehele site, geldt hier hetzelfde. Het autoload attribuut is standaard ingesteld op “ja” voor ontwikkelaars, maar in theorie zou niet elke plugin hun data op elke pagina moeten laden.

Het probleem waar WordPress sites tegenaan kunnen lopen is wanneer er een grote hoeveelheid autoloadeed data in de `wp_options` tabel staat. Dit is typisch een gevolg van het volgende:

- De gegevens worden door een plugin autoloadeed, terwijl deze op “nee” gezet zou moeten zijn. Een goed voorbeeld hiervan is een contactformulierplugin. Moet hij op elke pagina of alleen op de contactpagina gegevens laden?
- Plugins of thema's zijn verwijderd van de WordPress site, maar hun opties zijn nog steeds achtergelaten in de `wp_options` tabel. Dit kan betekenen dat er bij elke aanvraag onnodige autoloadeed gegevens worden opgevraagd.
- Plugin- en thema-ontwikkelaars slaan gegevens in de `wp_options`-tabel op in plaats van hun eigen tabellen te gebruiken. Er zijn argumenten voor beide opties, omdat sommige ontwikkelaars de voorkeur geven aan plugins die geen extra tabellen maken. Echter, de `wp_options` tabel is ook niet ontworpen om duizenden rijen te bevatten.

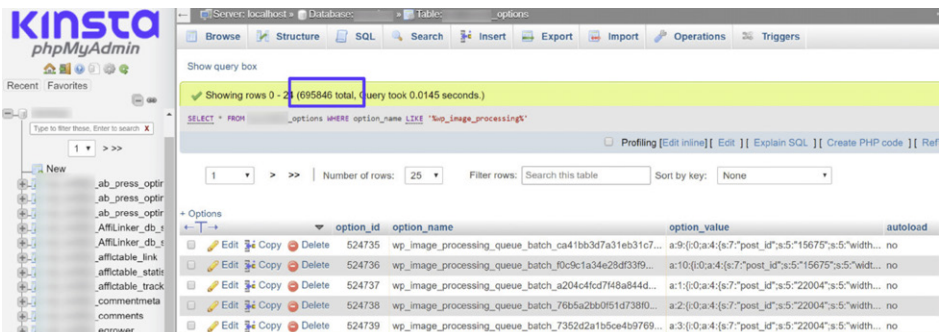
Hoeveel is teveel autoloade data? Dit kan natuurlijk variëren, maar idealiter wil je dat dit tussen de 300 KB en 1MB ligt. Zodra je het bereik van 3-5 MB of meer begint te naderen, zijn er waarschijnlijk dingen die geoptimaliseerd of verwijderd kunnen worden uit de autoloade. En alles boven 10 MB moet meteen worden aangepakt. Dit betekent niet altijd dat het een probleem zal veroorzaken, maar het is een goede plek om te beginnen.

Het opschonen van autoloade data is iets waar we klanten altijd mee helpen bij Kinsta.

*Wanneer heb je voor het laatst je wp\_options tabel opgeruimd? Ja... dat dachten we al. Kom op!*

## Opschonen van transients

Tenzij je object cache gebruikt, slaat WordPress tijdelijke (transients) records op in de wp\_options tabel. Meestal krijgen deze een verlooptijd en moeten ze in de loop van de tijd verdwijnen. Dat is echter niet altijd het geval. We hebben een aantal databases gezien waar duizenden oude transient records staan. Sterker nog, we hebben op één site enkele corrupte transient records gezien waardoor meer dan **695.000 rijen stonden opgeslagen** in de wp\_options tabel. Yikes!



The screenshot shows the phpMyAdmin interface. The main window displays a query result for the wp\_options table. The status bar indicates "Showing rows 0 - 25 (695046 total, Query took 0.0145 seconds.)". The query is: `SELECT * FROM wp_options WHERE option_name LIKE 'wp_image_processing%'`. Below the query, there are controls for filtering and sorting. The table view shows columns: option\_id, option\_name, option\_value, and autoload. The first few rows are visible, showing image processing queue batch records.

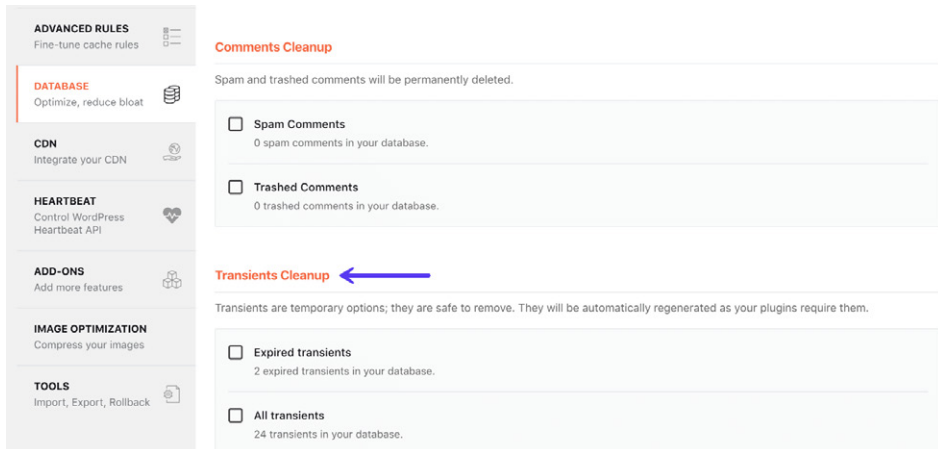
option_id	option_name	option_value	autoload
524735	wp_image_processing_queue_batch_ca41bb3d7a31eb31c7...	a:9:{0:a:4:{s:7:"post_id";s:5:"15675";s:5:"width...	no
524736	wp_image_processing_queue_batch_f0c9c1a34e28df33f9...	a:10:{0:a:4:{s:7:"post_id";s:5:"15675";s:5:"width...	no
524737	wp_image_processing_queue_batch_a204c4fd748a844d...	a:1:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...	no
524738	wp_image_processing_queue_batch_76b5a2bb0f51d73810...	a:2:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...	no
524739	wp_image_processing_queue_batch_7352d2a1b5ce4b9769...	a:3:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...	no

Het is ook belangrijk om op te merken dat transients niet standaard worden ge-autoload. Je kunt gebruik maken van een query zoals hieronder om te zien of er sprake is van autoloade transients.

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

Een betere en veiligere optie zou zijn om een gratis plugin te gebruiken zoals Transient Cleaner of Delete Expired Transients (beide zijn beschikbaar op WordPress.org) die alleen de verlopen transients uit je wp\_options tabel kunnen opschonen. Het lijkt er echter op dat er nu een functie in WordPress is, toegevoegd in 4.9, die verlopen transients opschooft. Hopelijk gebeurt dat nu dus automatisch op je site.

WP Rocket heeft ook de mogelijkheid om transients op te ruimen in hun database optimalisatie opties.



## WordPress sessies opruimen

Een ander veelvoorkomend probleem dat we hebben gezien is dat de cron jobs soms uit de pas lopen of niet starten, en daardoor worden sessies niet opgeruimd. Je kunt het

krijgen van grote aantallen `_wp_session` rijen in je database beperken. In dit voorbeeld hieronder heeft de betreffende site meer dan 3 miljoen rijen in hun `wp_options` tabel. En de tabel was gegroeid tot meer dan 600 MB groot.

wp_links	0	InnoDB	utf8mb4_unicode_ci	32	KiB
wp_options	~3,419,339	InnoDB	utf8mb4_unicode_ci	620.9	MiB
wp_postmeta	12,986	InnoDB	utf8mb4_unicode_ci	9.1	MiB
wp_posts	2,367	InnoDB	utf8mb4_unicode_ci	4	MiB

Je zou een query als de onderstaande kunnen gebruiken om te zien of je dit probleem hebt:

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

<< < 6491 > >> | Number of rows: 500 | Filter rows: Search this table | Sort by key: None

option_id	option_name	option_value	autoload
3301835	_wp_session_254a6d16b4400d5aabf3c48b41b72d2b	a:1:{s:10:"edd_errors";s:0:""} no	no
3301836	_wp_session_expires_254a6d16b4400d5aabf3c48b41b72d...	1520801306	no
3301837	_wp_session_747c86aefae52149d1068a60f240249f	a:1:{s:10:"edd_errors";s:0:""} no	no
3301838	_wp_session_expires_747c86aefae52149d1068a60f24024...	1520801306	no
3301839	_wp_session_17aa53659273873116d83993bae7cfd8	a:1:{s:10:"edd_errors";s:0:""} no	no
3301840	_wp_session_expires_17aa53659273873116d83993bae7cf...	1520801306	no
3301841	_wp_session_f3f29f822eec97951504e4d66afcd081	a:1:{s:10:"edd_errors";s:0:""} no	no
3301842	_wp_session_expires_f3f29f822eec97951504e4d66afcd0...	1520801306	no
3301843	_wp_session_2d4be94abb1e766ee229506e1c26e60e	a:1:{s:10:"edd_errors";s:0:""} no	no

In de meeste gevallen kan je deze dan veilig verwijderen (zoals een cron job zou moeten hebben) met het volgende commando:

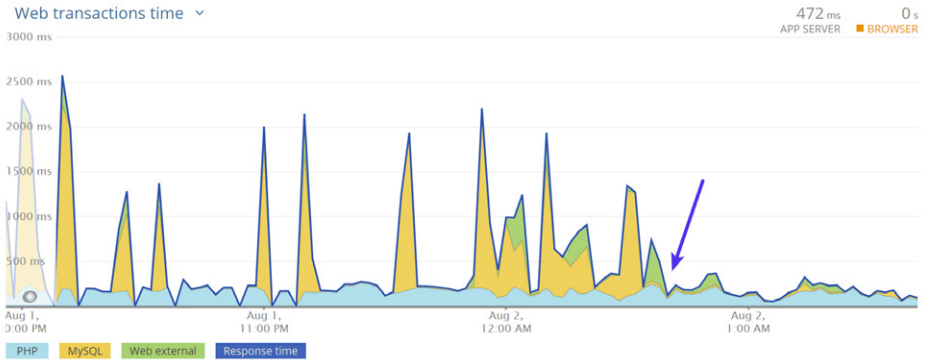
```
DELETE FROM `wp_options`
WHERE `option_name` LIKE '_wp_session_%'
```

Na het opruimen van alle overgebleven `_wp_session_` rijen had de tabel minder dan 1.000 rijen en werd deze teruggebracht tot 11 MB.

wp_links	0	InnoDB	utf8mb4_unicode_ci	32	KiB
wp_options	672	InnoDB	utf8mb4_unicode_ci	11.5	MiB
wp_postmeta	12,986	InnoDB	utf8mb4_unicode_ci	9.1	MiB
wp_posts	2,367	InnoDB	utf8mb4_unicode_ci	4	MiB



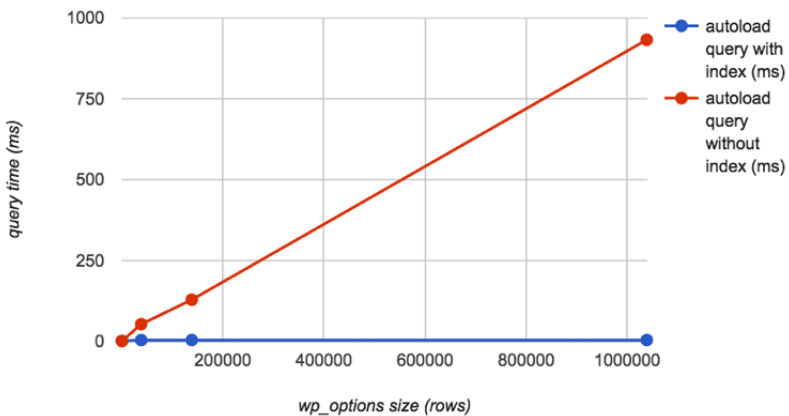
Dit repareerde ook de pieken die de site in MySQL kreeg.



## Een index toevoegen aan Autoload

Als het opruimen van je `wp_options` tabel niet genoeg was, kan je proberen een “index” toe te voegen aan het autoload-veld. Dit kan in wezen helpen om efficiënter te zoeken. Het geweldige team van 10up, voerde enkele testscenario's uit op een `wp_options` tabel met een typisch aantal autoload-records om te laten zien hoe het toevoegen van een autoload-index aan `wp_options`-queries de prestaties kan verbeteren.

## wp\_options size vs query time



Bron: 10up

## Gebruik Redis als een Persistent Object Cache voor WordPress

Redis is open-source, in-memory data structure opslag. In de context van WordPress kan Redis worden gebruikt om de waarden die door de native object cache van WordPress worden gegenereerd blijvend op te slaan, zodat objecten in de cache tussen de pagina's kunnen worden hergebruikt.

Het gebruik van een persistente object cache zoals Redis maakt het mogelijk om **objecten in de cache te hergebruiken** in plaats van te vragen dat de MySQL-



database een tweede keer wordt geraadpleegd voor hetzelfde object. Het resultaat is dat Redis de belasting van de MySQL database van een website kan verminderen, waardoor tegelijkertijd de reactietijd van de site afneemt en de mogelijkheid van de site om extra verkeer te verwerken.

Zeer dynamische websites (WooCommerce, lidmaatschapssites, forums, discussiefora, blogs met extreem actieve commentaarsystemen) die geen goed gebruik kunnen maken van paginacaching zijn potentiële kandidaten voor een persistente object caching optie zoals Redis.

Als je een Kinsta klant bent, bieden wij je een Redis-add-on aan.

## Gebruik Elasticsearch om het zoeken in WordPress te versnellen

Elasticsearch is een open-source full-text zoekmachine. Het wordt gebruikt om gegevens te indexeren en ongelooflijk snel te doorzoeken.

In de context van WordPress kan Elasticsearch worden gebruikt om het **raadplegen van de WordPress database te versnellen**. Dit wordt gedaan door een index op te stellen van de inhoud van de database van je site waarmee vervolgens, met behulp van Elasticsearch, veel sneller gezocht kan worden dan dat een MySQL-query in staat is om dezelfde zoekactie uit te voeren.



Als je de tijd en de mogelijkheden hebt, kan Elasticsearch worden geïntegreerd in een WordPress site door een zeer kundige WordPress- en Elasticsearch-ontwikkelaar. Als je site relatief standaard gebruik maakt van WP\_Query, kan Elasticsearch ook worden geïntegreerd door het installeren van ElasticPress, een gratis WordPress plugin van IOup, beschikbaar op WordPress.org, die automatisch integreert met het WP\_Query object om query resultaten te genereren met Elasticsearch in plaats van MySQL.

Elke site die veel gebruik maakt van WP\_Query kan profiteren van Elasticsearch. Voorbeelden van sites die kunnen profiteren van Elasticsearch:

- Sites waar de zoekopdracht het belangrijkste navigatiemiddel is.
- WooCommerce sites met een enorm aantal bestellingen waar site admins regelmatig de lijst met bestellingen moeten kunnen doorzoeken.
- Elke site met een groot aantal berichten waar MySQL-query's onaanvaardbaar trage resultaten opleveren.

Net als bij Redis hebben we bij Kinsta ook een Elasticsearch add-on.

## Schakel niet-kritieke functies uit die database intensief zijn.

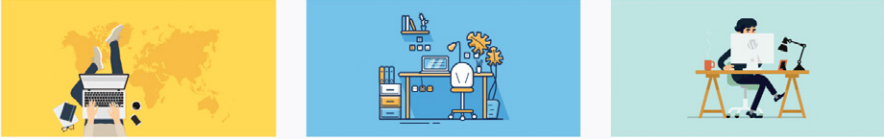
Dit lijkt misschien een beetje voor de hand liggend, maar het kan een wereld van verschil betekenen als je niet-kritische plugins en themafuncties die database-intensief zijn uitschakelt.

- Populaire- en/of gerelateerde postwidgets en plugins zijn verschrikkelijk. Ze hebben meestal zware sitewide query's.
- Afbeeldingsoptimalisatie-plugins die beelden comprimeren met behulp van je server. Gebruik altijd een plugin voor afbeeldingsoptimalisatie die de beelden extern optimaliseert.

Als je de Kinsta blog bezoekt en naar beneden scrollt naar het einde van een bericht, zul je merken dat we handgekozen gerelateerde artikelen hebben. Deze worden door ons

handmatig geselecteerd en aan de post toegewezen. Dit reduceert de query tot bijna niets en zal de prestaties van je gehele site niet schaden. Is er meer werk nodig? Ja, maar het is beter, want je kunt kiezen wat je wilt dat de lezers zien.

### 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

Hoe hebben we dit bereikt? We gebruikten de geweldige Advanced Custom Fields plugin (beschikbaar op [WordPress.org](https://www.wordpress.org)) en voegen deze velden vervolgens toe aan ons blogposttype. Dit stelt ons in staat om te zoeken en toe te wijzen welke gerelateerde inhoud we willen koppelen aan elk van onze blogberichten (zoals hieronder te zien is).


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.*

Search...

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



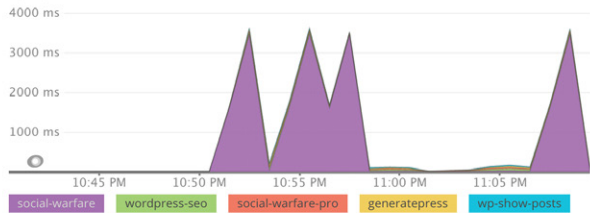
We raden ook aan om uit de buurt te blijven van plugins die een view/post counter aan je site toevoegen, tenzij je deze absoluut nodig hebt. Vermijd bijvoorbeeld dingen als “792 berichten” naast de avatar van een gebruiker in forumberichten of “5.243 weergaven” bij het vermelden van forumberichten. Wanneer je een lange discussie hebt, zullen deze tellers een enorme tol eisen aan je database. In het algemeen, minimaliseer het gebruik van tellers en gebruik ze alleen als dat nodig is.

Dit geldt ook voor veel sociale tellers. Zo kun je op deze site hieronder zien dat de responstijd van de populaire Social Warfare plugin 30x meer is dan de volgende plugin eronder. Caching is ingeschakeld, maar uiteraard heeft deze plugin een behoorlijke impact op de prestaties. Na het uitschakelen van de plugin op de site zijn de laadtijden direct verbeterd en is de reactiesnelheid van het WordPress admin dashboard verbeterd.

Sort by

social-warfare	18.9 sec
wordpress-seo	624 ms
social-warfare-pro	449 ms
generatepress	425 ms
wp-show-posts	389 ms
aaawp	356 ms
gp-premium	229 ms
nogluten	160 ms
imagify	115 ms
wp-review	107 ms

Top 5 plugins and themes  
by response time



HOOFDSTUK 11:

# Gebruik van een Content Delivery Network (CDN)



**CDN is een afkorting voor content delivery network. Dit is een netwerk van servers (ook wel POP's genoemd) die zich over de hele wereld bevinden. Ze zijn ontworpen om kopieën van de statische (en soms dynamische) inhoud van je WordPress site te hosten en af te leveren, zoals afbeeldingen, CSS, JavaScript en videostreams.**

Ten eerste wil je een CDN niet verwarren met je WordPress host. Dit zijn volledig gescheiden diensten. Een CDN is geen vervanging voor je hostingprovider, maar eerder een extra manier om de snelheid van je site te verhogen. Terwijl onze hosting hier bij Kinsta razendsnel is, kan een CDN je site nog sneller maken.

## Hoe een CDN werkt

Hoe werkt een CDN precies? Wel, wanneer je bijvoorbeeld je website bij Kinsta host, moet je een fysieke datacenterlocatie kiezen, zoals de VS, Europa, Azië-Pacific of Zuid-Amerika.

Laten we zeggen dat je kiest voor US Central. Dit betekent dat je website zich fysiek bevindt op een "host server" in Council Bluffs, Iowa. Wanneer mensen in Europa je website bezoeken, zal het langer duren voordat deze wordt geladen dan iemand die hem bezoekt vanuit bijvoorbeeld Dallas, TX.

Waarom? Omdat de gegevens een grotere afstand moeten afleggen. Dit is wat bekend staat als latency. Latency verwijst naar de tijd en/of vertraging die betrokken is bij de overdracht van gegevens over een netwerk. Hoe groter de afstand, hoe groter de latency.

## Soorten CDN's

Er zijn twee verschillende soorten content delivery netwerken:

1. Traditionele Pull CDN
2. Reverse Proxy CDN

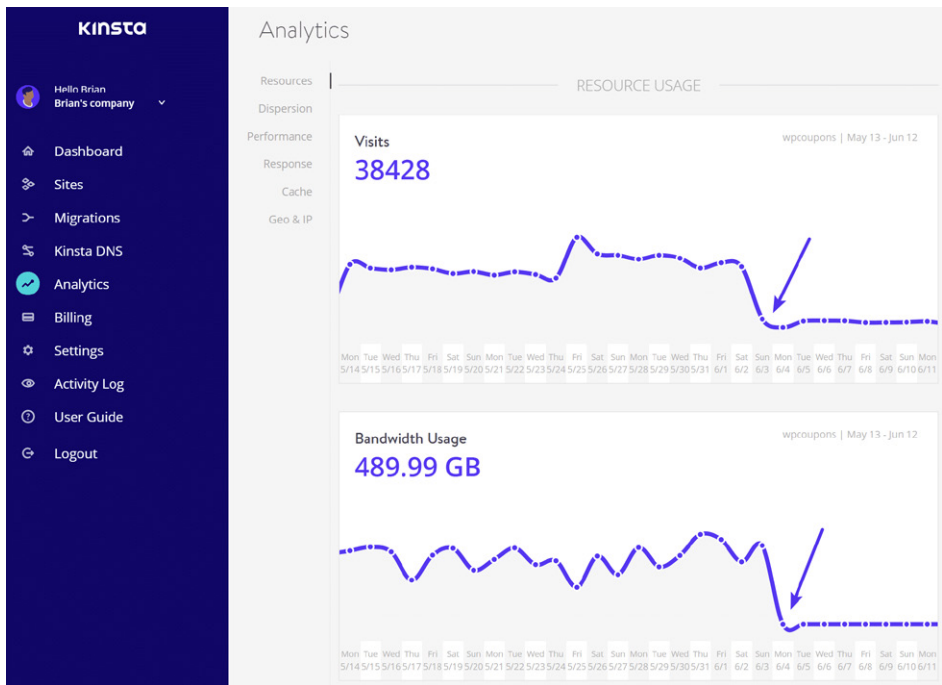
Traditionele pull CDN's cachen een kopie van al je inhoud en media, maar een verzoek van de klant komt nog steeds rechtstreeks bij je hosting provider terecht. KeyCDN en CDN77 zijn voorbeelden van traditionele CDN's.

Een reverse proxy CDN is iets anders. Terwijl het zich nog steeds gedraagt als een CDN, onderschept het alle inkomende verzoeken en fungeert het als een intermediaire server tussen de client en je host. Cloudflare en Sucuri zijn voorbeelden van reverse proxy CDN's. Dit is één van de redenen waarom je je DNS rechtstreeks naar deze providers moet wijzen in plaats van naar je host.

Het voordeel hiervan is dat ze fungeren als een tussen-server, ze kunnen zorgen voor sterke webapplicatie firewalls die kunnen helpen om het slechte verkeer te blokkeren van het ooit raken van je WordPress site en of hosting provider. Een nadeel hiervan is dat ze met een beetje extra overhead in termen van prestaties komen, in vergelijking met een traditionele Pull CDN. Maar met extra prestaties en veiligheidskenmerken kan dit als verwaarloosbaar worden beschouwd.

Hieronder staat een voorbeeld van wat er gebeurde na het inschakelen van Sucuri op de site van een klant. Zoals je kunt zien had het een dramatische impact op de hoeveelheid slecht verkeer dat doorkwam. Uiteindelijk kunnen dit soort diensten je helpen om te besparen op je hostingkosten.





## CDN-snelheidstests

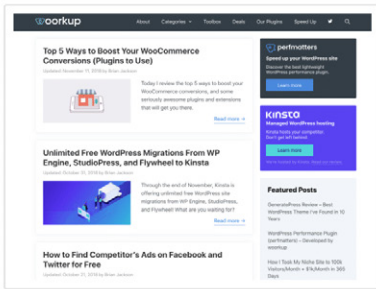
Eerder hadden we het al over de enorme voordelen van WordPress caching. Nou, CDN caching is ook super krachtig. Dit komt omdat CDN's meestal veel meer serverlocaties hebben dan hostingproviders. Dit betekent dat ze al je bestanden (afbeeldingen, JS, CSS) dichterbij je bezoekers kunnen cachen en ze razendsnel kunnen versturen.

Laten we een paar snelle tests doen om te zien hoeveel sneller je site zou kunnen zijn met een CDN.

## Zonder CDN

Onze testwebsite wordt gehost bij Kinsta en is fysiek gevestigd in het datacenter in Iowa, USA. We hebben eerst vijf snelheidstesten uitgevoerd in Pingdom (zonder de

CDN-functie) en hebben het gemiddelde genomen. Belangrijk: we gebruiken de locatie Europa - Verenigd Koninkrijk - Londen in Pingdom om de echte kracht van een CDN aan te tonen. De totale laadtijd was **1,03 s**.



Performance grade

**B 89**

Page size

**231.0 KB**

Load time

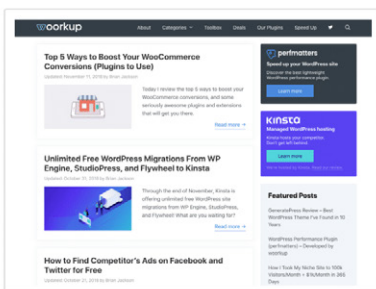
**1.03 s**

Requests

**40**

## Met CDN

We hebben toen onze CDN ingeschakeld en vijf extra snelheidstesten uitgevoerd in Pingdom. Onze totale laadtijd is nu **585 ms** van de testlocatie Europa - Verenigd Koninkrijk - Londen Pingdom. Door gebruik te maken van het CDN konden we de **laadtijd van onze pagina's dus met 43,2% verminderen!** Dat is enorm!



Performance grade

**B 89**

Page size

**230.6 KB**

Load time

**585 ms**

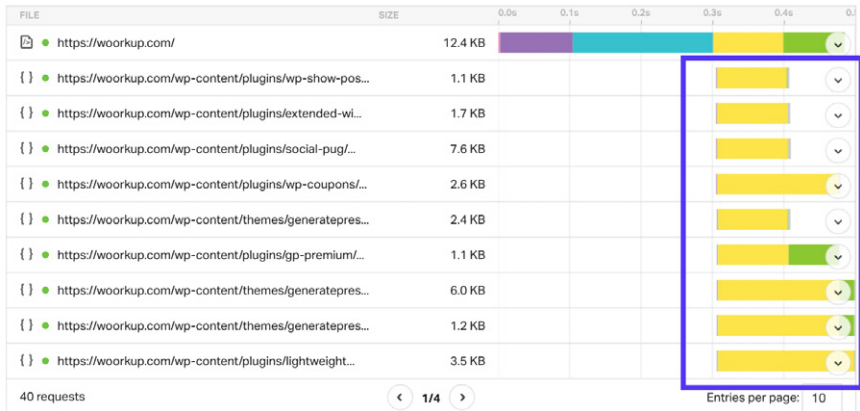
Requests

**40**

De reden voor zo'n drastisch verschil is dat het CDN een datacenter in Londen heeft. Dit betekent dat alle bestanden op die locatie worden gecached en klaar zijn om met minimale vertraging te worden verstuurd.

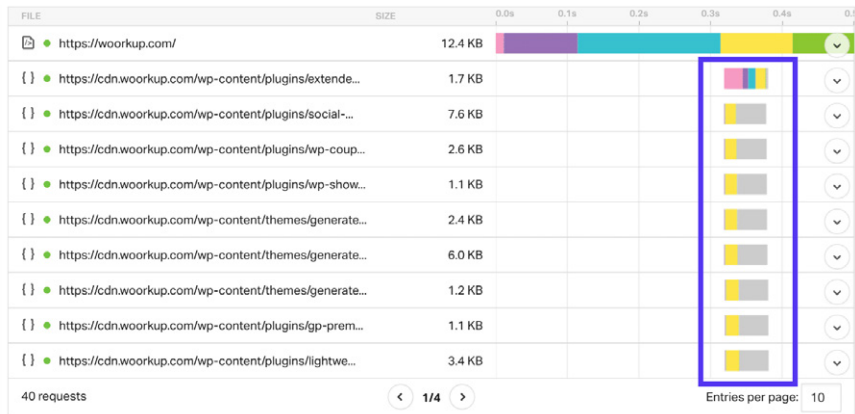
# TTFB zonder CDN

Vergeet niet dat de gele balk in Pingdom staat voor wachttijd, dat is tijd voor de eerste byte (TTFB). Op onze snelheidstesten zonder dat het CDN draait was de gemiddelde TTFB op activa ongeveer 98 ms.



# TTFB met CDN

Toen we het CDN inschakelden, daalde de gemiddelde TTFB op de bestanden tot gemiddeld **15 ms**. Door het gebruik van een CDN **daalde onze gemiddelde TTFB dus met 84,69%**. Dit komt vooral omdat de bestanden rechtstreeks vanuit de cache van het CDN werden verstuurd.



*Een CDN verminderde de laadtijd  
van onze pagina's met 43,2%!  
Bekijk waarom je er één zou moeten  
gebruiken.*

## Hoe schakel je een CDN in?

Het inschakelen van een CDN op je WordPress site hoeft niet moeilijk te zijn, het is vrij eenvoudig! Volg gewoon deze stappen.

### Stap 1

Selecteer een CDN-provider en abonneer je op hun dienst. Deze worden doorgaans maandelijks gefactureerd of op basis van het gegevensgebruik. De meeste providers hebben een calculator om je kosten in te schatten.

- Elke CDN-aanbieder zou documentatie op zijn sites moeten hebben om je op weg te helpen.
- We hebben ook uitgebreide tutorials over hoe Cloudflare en Sucuri te installeren beschikbaar op onze website op [kinsta.com](http://kinsta.com).

### Stap 2

Als je een traditionele pull CDN gebruikt, kun je gebruik maken van een gratis plugin zoals CDN Enabler (beschikbaar op [WordPress.org](http://WordPress.org)), WP Rocket (beschikbaar op [wp-rocket.me](http://wp-rocket.me)), of Perfmatters (beschikbaar op [perfmatters.io](http://perfmatters.io)) om het te integreren met je WordPress site. Deze plugins koppelen automatisch je bestanden aan het CDN. Er is geen werk nodig van jouw kant om je content op het CDN te krijgen; dit wordt allemaal geregeld! Reverse Proxy CDN's hebben meestal geen plugins nodig, hoewel ze soms wel extra functies hebben.

- 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** ?

Enable CDN in WordPress with Perfmatters

## Hoe kan ik het Kinsta CDN inschakelen?

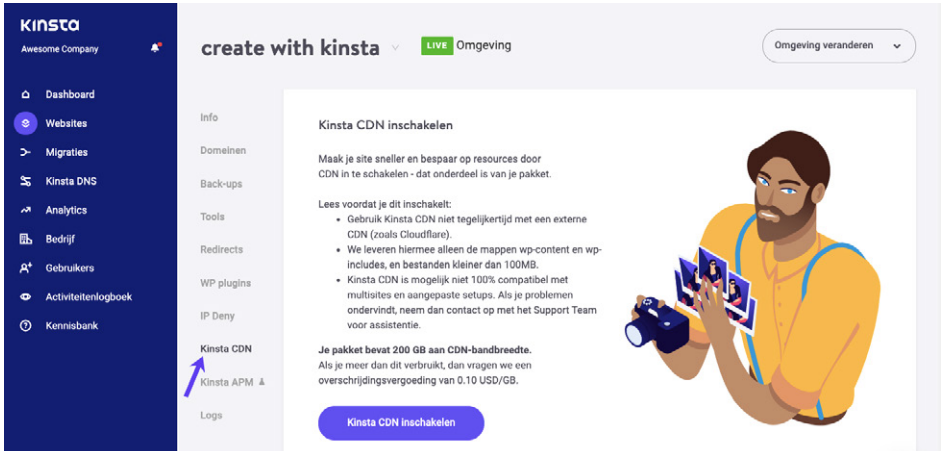
Vond je die CDN-snelheidstesten hierboven leuk? We gebruikten KeyCDN in die tests. Het grote nieuws is dat onze Kinsta CDN wordt aangedreven door KeyCDN. Het is een HTTP/2 en IPv6-geschikt CDN met 34+ locaties, om je bestanden en media over de hele wereld te turbochargen. Momenteel worden de volgende regio's ondersteund: Amerika, Zuid-Amerika, Europa, Afrika, Azië en Australië.



Als je een Kinsta klant bent, nemen we gratis CDN-bandbreedte op in al onze hostingpakketten. Je kunt het Kinsta CDN in twee eenvoudige stappen inschakelen.

# Stap 1

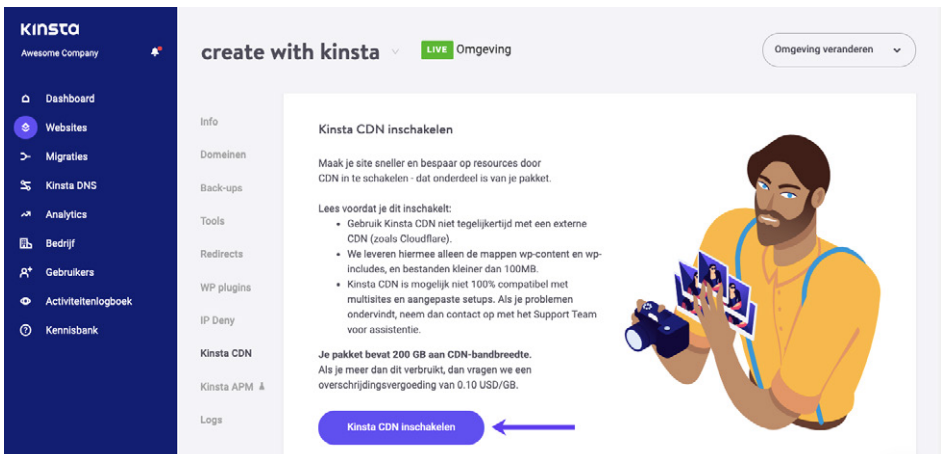
Log eerst in op je MyKinsta-dashboard. Klik op je site en vervolgens op het Kinsta CDN tabblad.



The screenshot shows the Kinsta dashboard interface. On the left is a dark blue sidebar with the Kinsta logo and navigation menu items: Dashboard, Websites, Migrations, Kinsta DNS, Analytics, Bedrijf, Gebruikers, Activiteitenlogboek, and Kennisbank. The 'Websites' item is selected. The main content area has a header with 'create with kinsta', a 'LIVE' status indicator, and an 'Omgeving' dropdown menu. Below the header is a sidebar with menu items: Info, Domeinen, Back-ups, Tools, Redirects, WP plugins, IP Deny, Kinsta CDN (highlighted with a blue arrow), Kinsta APM, and Logs. The main content area displays the 'Kinsta CDN inschakelen' page, which includes an illustration of a man with a camera and a list of instructions for enabling the CDN.

# Stap 2

Klik dan op “CDN inschakelen”. Na een paar minuten wordt het CDN automatisch ingezet, en zal je site vanuit de cache over de hele wereld worden gedistribueerd. Meer is er niet nodig.



This screenshot is identical to the one in Step 1, but with a blue arrow pointing to the 'Kinsta CDN inschakelen' button at the bottom of the main content area. The button is a purple rounded rectangle with white text.

## Extra CDN-optimalisaties

Hier zijn een paar extra CDN-optimalisaties die je misschien wilt bekijken of overwegen:

- Als je veel reacties op je site krijgt, kunnen gravatar's veel verzoeken genereren en laden van `secure.gravatar.com`. In plaats daarvan kun je de gravatar's laden vanaf je CDN. We doen dit op de Kinsta website. Als je een Kinsta klant bent, neem dan contact op met ons supportteam om dit te laten configureren.
- Je kunt ook aangepaste lettertypen, Google-lettertypen en zelfs je favicon op je CDN hosten. Links naar tutorials over hoe dit te doen zijn beschikbaar op onze website op <https://kinsta.com/speed-up-wordpress>.

HOOFDSTUK 12:

# Media en e-mail verwijderen wanneer dat nodig is





Alles wat een request genereert heeft op één of andere manier invloed op de prestaties van je site. Voor sites met honderdduizenden bestanden of grote media bestanden kan het verstandig zijn om dit volledig te offloaden. **Offloading is anders dan het gebruiken van een CDN. Met een CDN** blijft de originele data nog steeds bij je host staan, het CDN heeft er gewoon meerdere kopieën van.

Wanneer de caching op je CDN-bestanden afloopt, vraagt het je host opnieuw om de laatste kopieën van de bestanden. CDN's zijn bedoeld om bestanden voor lange tijd te cachen. Maar vanwege het feit dat ze zoveel POP's hebben, kan er veel re-querying plaatsvinden als de cache in verschillende regio's afloopt.

Wanneer je media of bestanden offload, betekent dit dat je de oorspronkelijke fysieke locatie van de media of bestanden van je hostingprovider moet verplaatsen. Dus terwijl het misschien lijkt dat de bestanden vanaf je site komen, komen ze echt ergens anders vandaan. Naast het verminderen van extra requests naar de host, is dit natuurlijk ook een manier om op schijfruimte te besparen.

## Media offloaden naar Amazon S3

Een van de meest populaire oplossingen is Amazon S3. Amazon S3 is een opslagoplossing en maakt deel uit van de vele producten van Amazon Web Services.

Meestal wordt dit gebruikt voor grote sites die ofwel extra back-ups nodig hebben ofwel grote bestanden (downloads, software, video's, games, audiobestanden, PDF's, etc.) bevatten. Amazon heeft bewezen zeer betrouwbaar te zijn, en door hun enorme infrastructuur kunnen ze zeer lage opslagkosten bieden. Enkele van de klanten van S3 zijn Netflix, Airbnb, SmugMug, Nasdaq, enz.



Omdat ze volledig met bulkopslag werken, kan je bijna garanderen dat de prijsstelling goedkoper zal zijn dan die van je WordPress host. Het offloaden van media naar AWS kan een geweldige manier zijn om geld te besparen en is het eerste jaar gratis (tot 5 GB opslag). Ook omdat de requests voor je media rechtstreeks vanuit Amazon worden bediend, wordt je WordPress site hierdoor minder belast, wat een snellere laadtijd oplevert.

Wij raden aan om hiervoor de WP Offload Media Lite for Amazon S3 plugin (beschikbaar op WordPress.org) te gebruiken. De plugin kopieert automatisch bestanden van je WordPress site naar Amazon S3 als ze worden geüpload naar de mediabibliotheek. Je kunt ook een CDN met de offloaded media gebruiken voor het beste van beide werelden.

## Media offloaden naar Google Cloud Storage



Een andere populaire offloading oplossing is Google Cloud Storage. Aangezien Kinsta wordt aangedreven door Google Cloud Platform, zijn wij grote fans van hun technologie en infrastructuur. Door de enorme infrastructuur van Google en het feit dat ze in bulk met storage werken, kunnen ze zeer lage opslagkosten bieden. Enkele van hun klanten zijn Spotify, Vimeo, Coca-Cola, Philips, Evernote en Motorola.

Wij raden je aan om hiervoor de gratis WP-Stateless plugin te gebruiken (beschikbaar op WordPress.org). De plugin kopieert automatisch bestanden van je WordPress site naar Google Cloud Storage als ze worden geüpload naar de mediabibliotheek. Het biedt dan verschillende modi, zoals het maken van een back-up van je media of zelfs het verwijderen van de media uit WordPress en het verwerken van de media-requests uit Google Cloud Storage.

## Transactie- en marketingmails offloaden

E-mails kunnen wel degelijk een impact op je server en server resources hebben. Bij sommige hosts, met name shared hosts, kan misbruik hiervan zelfs leiden tot schorsing. Dit wordt vooral een probleem bij degenen die proberen om bulkmails te versturen. Dit is de reden waarom er transactionele e-mailproviders bestaan en waarom veel hostingproviders de bezorging van e-mail op standaardpoorten helemaal blokkeren. **We raden af om je hostingprovider te gebruiken voor e-mail.**

Als je nieuwsbrieven of bulkmails verstuurt, raden wij je altijd de volgende alternatieven aan om het beste resultaat te krijgen:

- Gebruik professionele e-mailmarketing software die geen deel uitmaakt van WordPress.
- Gebruik een transactionele e-mail service provider (HTTP API of SMTP) samen met WordPress.

Andere voordelen van het gebruik van een externe dienst zijn onder andere:

- Betere e-mail bezorging. Laat de e-mailproviders doen waar ze goed in zijn!
- Minder kans om geblokkeerd te worden op basis van je domeinnaam of IP-adres
- Het is misschien niet altijd mogelijk om DMARC-records op te zetten bij je hostingprovider.

## E-mailmarketingtools

Enkele voorbeelden van marketing e-mails zijn nieuwsbrieven, product- en functie-aankondigingen, uitverkoop, uitnodigingen voor evenementen, onboarding reminders, enz. Hier zijn enkele e-mailmarketingtools die wij aanbevelen:

- **MailChimp** – Wij gebruiken bij Kinsta MailChimp
- **MailerLite**
- **Drip**

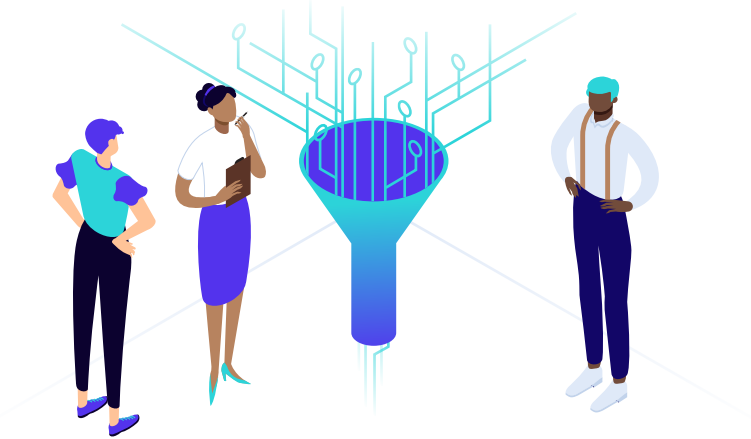
## Transactie-e-maildiensten

Enkele voorbeelden van transactie-e-mails zijn facturen van WooCommerce of EDD, meldingen over het aanmaken van een account, verzendmeldingen, app-foutmeldingen, wachtwoordherstel, enz. Als je een Kinsta klant bent, vertrouwen we op een externe SMTP-provider voor een hoge bezorggarantie. Maar afhankelijk van je volume, raden wij je altijd aan om deze naar een externe partij te verplaatsen. Hier zijn enkele transactie-e-maildiensten die wij aanbevelen:

- **SendGrid** – We gebruiken SendGrid bij Kinsta.
- **Mailgun**
- **SparkPost**

HOOFDSTUK 13:

# Knelpunten en trage plugins vinden



**Nu duiken we in enkele tips over hoe je knelpunten kunt vinden op je WordPress site en wat je er aan kunt doen.**

## **Gebruik New Relic om langzame database queries te vinden**

Er zijn een aantal geweldige tools op de markt die je kunnen helpen bij het lokaliseren en identificeren van langzame database queries en plugins die veel tijd in beslag nemen. Wij zijn grote fans van New Relic en gebruiken het dagelijks. New Relic is een PHP monitoring tool die je kunt gebruiken om gedetailleerde prestatiestatistieken van je website te krijgen.

Als je een Kinsta klant bent, kan je zelfs je eigen New Relic licentiesleutel op ons MyKinsta dashboard invoeren.

Gebruik echter New Relic met zorg, want het heeft invloed op de prestaties van je site. Het voegt JavaScript toe aan je website. We raden je aan het in te schakelen wanneer je problemen met de prestaties moet oplossen en het daarna weer uit te schakelen.

## Het vinden van langzame plugins

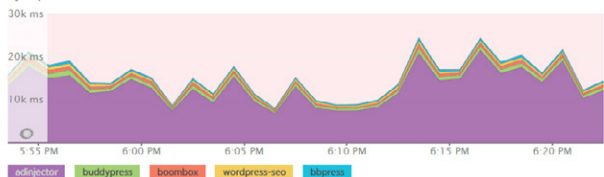
Wanneer een WordPress plugin de een trage site veroorzaakt, zullen de symptomen variëren afhankelijk van de activiteit die de plugin uitvoert. Echter, in veel gevallen zal een langzame plugin invloed hebben op elke pagina van een WordPress site. In het geval van de site waarvan je de gegevens in de onderstaande afbeelding ziet, is de algemene traagheid op elke pagina van de site waargenomen. Hier is wat New Relic te zeggen had over de prestaties van de plugins op de site.

TIME POKER  
Last 30 minutes ending now

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
zombily	4.22 sec

Top 5 plugins and themes  
by response time



Onmiddellijk zie je dat de adinjector plugin meer dan 15 keer zoveel tijd verbruikt als de volgende langzaamste plugin.

Wanneer je dit soort gegevens ziet, kan het verleidelijk zijn om de plugin onmiddellijk af te doen als slecht gecodeerd of op de een of andere manier ineffectief. Hoewel dit soms het geval is, is dit niet altijd het geval. Plugin misconfiguratie, database traagheid, of externe bronnen die traag reageren kunnen ervoor zorgen dat een plugin veel tijd in beslag neemt.

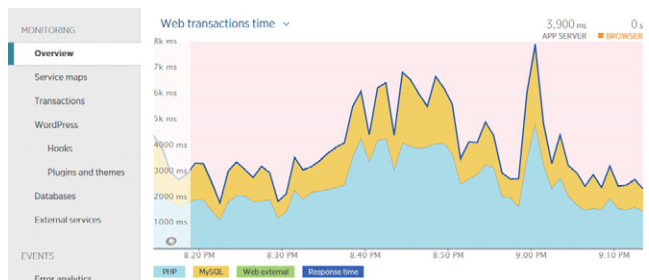
Dus als je een plugin ziet die langzaam reageert, is het een goed idee om verschillende andere schermen in New Relic te bekijken om aanvullende informatie te vinden. De transacties, databases en externe bronnen moeten allemaal worden gecontroleerd voordat je beslist dat het deactiveren van de plugin de beste of enige manier is om verder te gaan.

## Algehele traagheid veroorzaakt door een slecht geoptimaliseerde database

Een slecht geoptimaliseerde database kan leiden tot algemene traagheid op een WordPress site. Eerder hebben we al veel verschillende dingen doorgenomen die je kunt doen om dit op te lossen. In New Relic zal deze database gerelateerde traagheid hoogstwaarschijnlijk op twee plaatsen te zien zijn:

- Ten eerste zie je in het overzicht een buitenproportionele hoeveelheid MySQL-activiteit.
- Ten tweede zie je één of meer databasetabellen die veel tijd in beslag nemen in het databasetabblad.

Beginnend met het overzichtsscherm, zou een site met een worstelende database er misschien zo uit kunnen zien:







# Gebruik de de gratis Query Monitor plugin

Je kunt ook de gratis Query Monitor WordPress plugin gebruiken (beschikbaar op WordPress.org). Gebruik het om langzame databasequeries, AJAX-requests, REST API-requests en nog veel meer te identificeren en te debuggen. Daarnaast rapporteert de plugin website-details zoals script afhankelijkheden, WordPress hooks die getriggerd zijn tijdens het genereren van de pagina, hosting details, voorwaardelijke query tags, en nog veel meer.

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	

De plugin is ontwikkeld door John Blackbourn, een WordPress core comitter die momenteel een ontwikkelaar is bij Human Made en eerder in dienst was van WordPress.com VIP - met andere woorden, iemand die WordPress uitgebreid kent. Query Monitor is in 2013 toegevoegd aan de WordPress plugin directory en heeft momenteel meer dan 10.000 actieve installaties - een indrukwekkend aantal voor een ontwikkelingsplugin. De gebruikers van de plugin geven de plugin vijf van de vijf sterren, wat de populariteit verklaart onder ontwikkelaars.

## Maak gebruik van testomgevingen

We weten niet wat we zouden doen zonder testomgevingen. Deze kunnen van

onschatbare waarde zijn als het gaat om het oplossen van problemen met de prestaties. Gelukkig heeft Kinsta één-klik testomgevingen. Als je WordPress host geen testomgevingen biedt, kan je ook een plugin gebruiken zoals WP Staging (beschikbaar op WordPress.org), hoewel het niet zo eenvoudig op te zetten is.

**create with kinsta** TEST Omgeving Omgeving veranderen

**Live-omgeving**  
Wijzigingen die je hier maakt, via SFTP of de database, zijn zichtbaar. Wees dus alsjeblieft voorzichtig

**Testomgevingen**  
Bedoeld om te testen en voor ontwikkelingsdoeleinden. Wijzigingen zullen niet zichtbaar zijn op je live-website

**Je hebt geen testomgeving**  
Een testomgeving biedt je speelruimte om ni code en andere veranderingen te testen. Wanneer je een website aanmaakt in de testomgeving is dit een kopie van je live-website. Hierin kan je vervolgens wijzigingen aanbrengen en uitvoeren. Als je hiermee klaar bent, kan je alle veranderingen naar je live-site sturen met een druk op de knop.

**Een testomgeving aanmaken**

Nadat je een testomgeving hebt opgezet, is het eerste wat je kunt doen het uitschakelen van al je plugins. Aangezien dit een kopie is van je live site, hoeft je je geen zorgen te maken over het breken van iets. Het is veruit één van de gemakkelijkste manieren om problemen te vinden. Ga gewoon naar Plugins, selecteer ze allemaal en kies “Deactiveren” uit de bulkopties.

**Plugins** Add New

All (33) | Active (28) | Inactive (5) | Recently Active (4) | Update Available (2) | Must-Use (1)

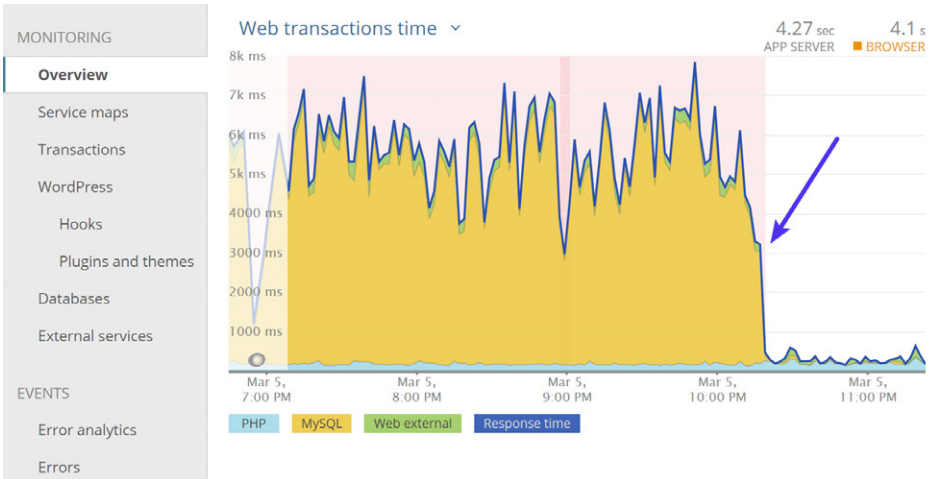
Bulk Actions Apply

Bulk Actions  
Activate  
**Deactivate**  
Update  
Delete Delete

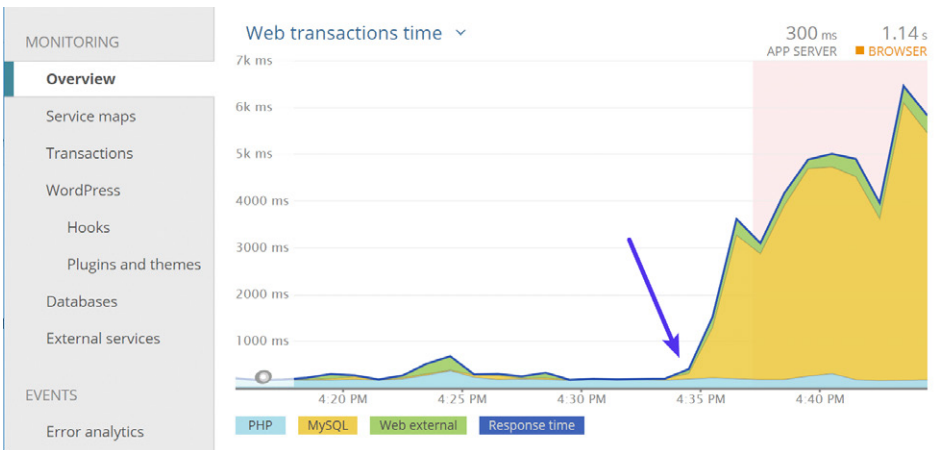
	Description
<input type="checkbox"/>	<b>Anti-Spam</b> Used by millions, Akismet is quite possibly the best wa Version 4.0.3   By Automattic   <a href="#">View details</a>
<input checked="" type="checkbox"/>	<b>Category Sticky Post</b> Deactivate Mark a post to be placed at the top of a specified cate Version 2.10.1   By Tom McFarin   <a href="#">View details</a>
<input checked="" type="checkbox"/>	<b>Code Snippets</b> Snippets   Deactivate An easy, clean and simple way to add code snippets to Version 2.10.1.1   By Shea Bunge   <a href="#">View details</a>   <a href="#">About</a>
<input checked="" type="checkbox"/>	<b>Complete Analytics Optimization Suite (CAOS)</b> Deactivate A plugin that allows you to completely optimize Googl Version 1.44   By Daan van den Bergh   <a href="#">View details</a>

Na dit te hebben gedaan, kan je de reactietijden monitoren in New Relic of Query Monitor en zien wat er gebeurt. In dit voorbeeld hieronder zijn de reactietijden direct

teruggevallen op de site, en nu we weten we dat het dus één van de plugins was die een probleem veroorzaakte. Je kunt ze dan één voor één opnieuw inschakelen, waarbij je hetzelfde proces herhaalt totdat je de oorzaak vindt.



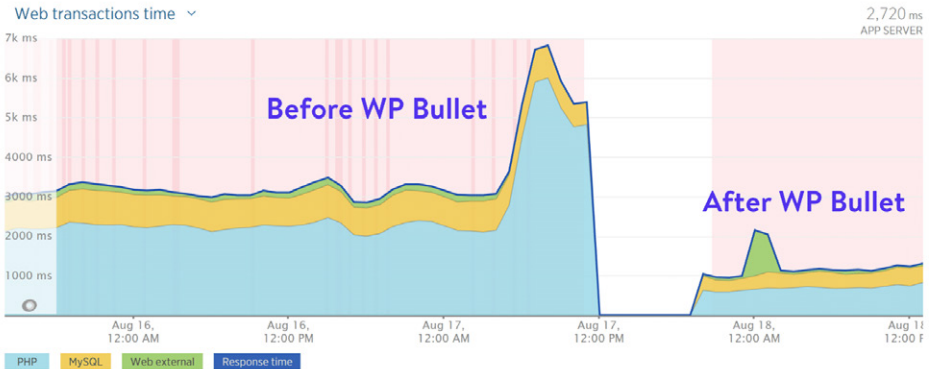
Hier is een voorbeeld van wat er gebeurde toen we de plugin die het probleem veroorzaakte aanzette. Laadtijden (webtransactietijden) gingen meteen weer omhoog.



Wat moet je doen nadat je de plugin hebt gevonden die de traagheid veroorzaakt? Hier is wat we adviseren:

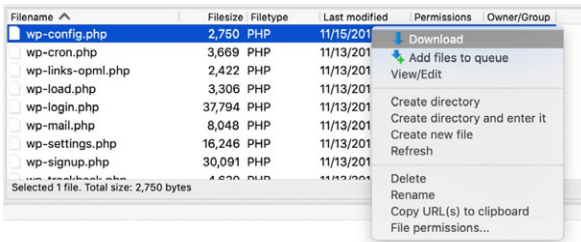
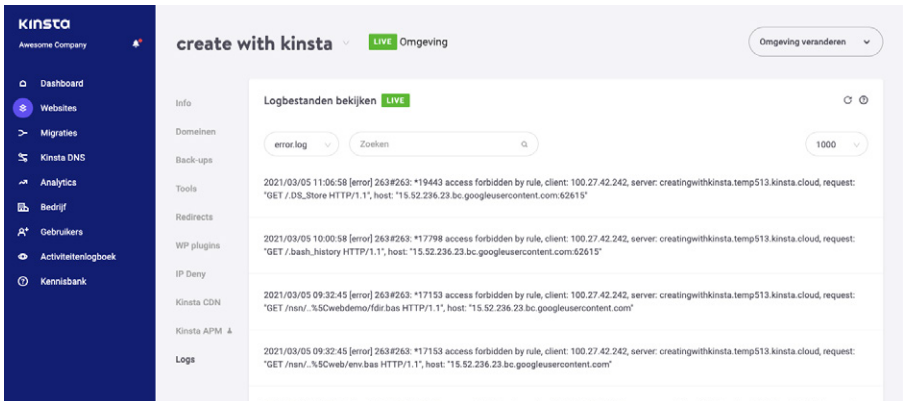
1. Update je plugins en thema's naar de laatste versie als je dat nog niet hebt gedaan.
2. Benader de ontwikkelaar van de plugin of het thema en vraag hen om assistentie.
3. Zoek een alternatieve plugin die dezelfde functionaliteit kan leveren.
4. Misschien veroorzaakt je PHP versie een probleem. Verander je PHP versie naar een lagere versie en kijk of de plugin of het thema dan werkt.

Je kunt ook een WordPress ontwikkelaar inhuren om het probleem op te lossen. Als het betrekking heeft op de prestaties, moeten we een persoonlijke shout-out geven aan Mike Andreason van wp-bullet.com. Hij is een fulltime Codeable ontwikkelaar gespecialiseerd in performance optimalisatie, die veel klanten hier bij Kinsta heeft geholpen om hun site naar een hoger niveau te tillen.



## Controleer je error logs

Het controleren van error logs is nooit leuk, maar kan veel onthullen over prestatieproblemen met WordPress plugins. Als je een Kinsta klant bent, kun je je error logs, cache-logs en access logs eenvoudig bekijken in het MyKinsta dashboard.



Je kunt ook error logs inschakelen door wat code toe te voegen aan je `wp-config.php` bestand. Eerst moet je verbinding maken met je site via SFTP. Download vervolgens je `wp-config.php` zodat je deze

kunt bewerken. Let op: Maak altijd eerst een backup van dit bestand!

Zoek de regel die zegt `/* That's all, stop editing! Happy blogging. */` en voeg vlak daarvoor het volgende toe (zoals hieronder te zien is):

```
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' );

```

Als de bovenstaande code al bestaat in je `wp-config.php`-bestand, maar is ingesteld op “false”, verander deze dan gewoon in “true”. Dit zal de debug-modus inschakelen. Opmerking: Je zult ook waarschuwingen of fouten zien in je WordPress admin als deze code aanwezig is.

Je kunt dan het debug-log inschakelen om alle fouten naar een bestand te sturen door de volgende code toe te voegen net na de `WP_DEBUG`-regel (zoals hieronder te zien is):

```
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');
```

Sla je wijzigingen op en upload deze naar je server. De fouten worden dan gelogd in het `debug.log` bestand in je `/wp-content/` map. Als je dit bestand om één of andere reden niet ziet, kan je deze altijd aanmaken.

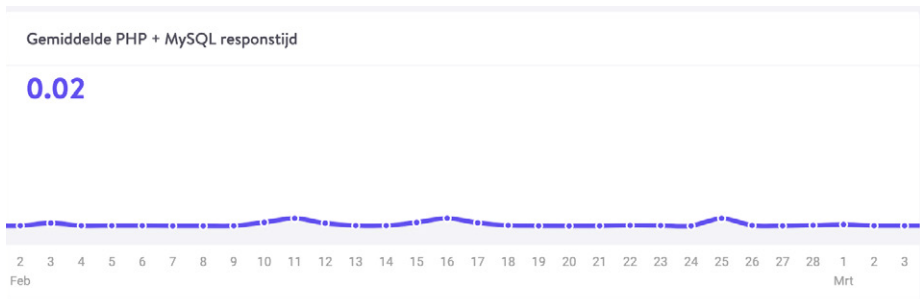
## Gebruik MyKinsta Analytics

Als je een Kinsta klant bent, kan je profiteren van de prestatie-inzichten die we hebben ingebouwd in onze MyKinsta Analytics-tool.

In de prestaties-sectie kun je je gemiddelde PHP + MySQL-reactietijd, PHP-doorvoer, AJAX-gebruik, topgemiddelde upstream-tijd en topmaximum upstream-tijd bekijken.

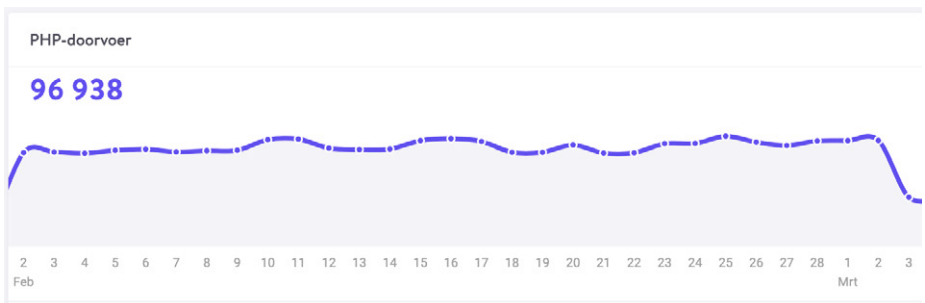
## Gemiddelde PHP + MySQL Responstijd

Wanneer je je WordPress site bezoekt, worden PHP en MySQL gebruikt om de gegevens die je op de pagina ziet te compileren en op te vragen. Deze grafiek toont je de gemiddelde reactietijd van de PHP-engine en de MySQL-engine voor elk niet in het geheugen opgeslagen dynamisch verzoek. Het kennen van deze reactietijden kan je helpen bij het oplossen van problemen met traagheid.



## PHP Doorvoer


















Doorvoer geeft het aantal transacties per seconde aan dat een applicatie kan verwerken, en in dit rapport verwijst het naar PHP doorvoer van je WordPress site. Met andere woorden, het laat je zien hoe vaak een PHP asset is opgevraagd.







# AJAX-gebruik

AJAX is een client-side script dat communiceert van en naar een server/database zonder dat er een postback of een complete pagina verversing nodig is. Als het gaat om WordPress, hebben velen dit waarschijnlijk gezien in snelheidstesten. De top twee van de problemen met AJAX zijn de plugins die ervoor zorgen dat het pieken vertoont en de CPU-problemen op de back-end.

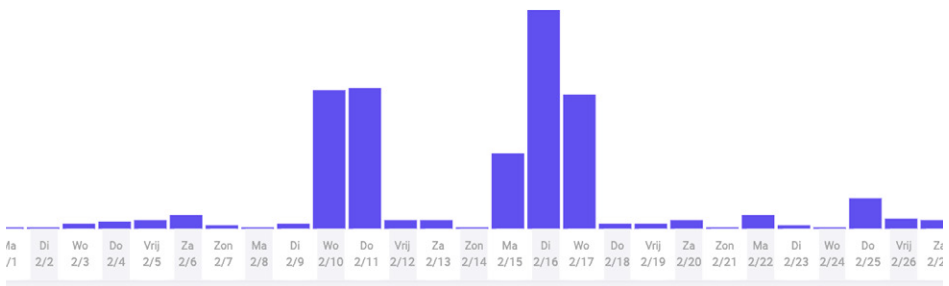
 jquery.scrollTo.js	cdn.wpdev.ink/wp-content/themes/twent...	2.9 kB		
 local-ga.js	cdn.wpdev.ink/wp-content/plugins/host...	12.5 kB		
 open-sans-v13-latin-regular.woff2	cdn.wpdev.ink/fonts/	15.7 kB		
 open-sans-v13-latin-700.woff2	cdn.wpdev.ink/fonts/	16.4 kB		
 admin-ajax.php	wpdev.ink/wp-admin/	0 B		
 collect?v=1& v=149&a=1502808721&t=pag...	www.google-analytics.com/r/	421 B		
18 requests		276.4 kB	640 ms	

 Send 0 ms  
 Wait 791 ms  
Total 791 ms

Het AJAX-gebruiksrapport in MyKinsta analytics kan een geweldige manier zijn om je te helpen bij het oplossen van dit soort problemen, omdat je kunt zien of je bepaalde AJAX-pieken ziet tijdens bepaalde perioden. Deze grafiek toont de telling van de admin-ajax verzoeken. Je kunt dan gebruik maken van een aantal van de tips in het bericht dat we hierboven hebben genoemd om te bepalen waar ze vandaan kunnen komen.

## AJAX-verbruik

703



# Top Gemiddelde PHP + MySQL Responstijd

Deze lijst toont de hoogste gemiddelde reactietijden van PHP en MySQL. Deze getallen kunnen éénmalige pieken zijn, dus het wordt aangeraden om deze lijst te vergelijken met “Top Maximum Upstream Time”.

PAD	AANVRAGEN	TIJD
/google0064c97a60a07595.html	3	19.18 s
/wp-json/regenerate-thumbnails/v1/regenerate/42? only_regenerate_missing_thumbnails=true&delete_unregistered_thumb nail_files=false&update_usages_in_posts=false	1	3.03 s
/wp-json/regenerate-thumbnails/v1/regenerate/43? only_regenerate_missing_thumbnails=true&delete_unregistered_thumb nail_files=false&update_usages_in_posts=false	1	2.81 s
/wp-json/regenerate-thumbnails/v1/regenerate/168? only_regenerate_missing_thumbnails=true&delete_unregistered_thumb nail_files=false&update_usages_in_posts=false	1	2.80 s
/wp-json/regenerate-thumbnails/v1/regenerate/93? only_regenerate_missing_thumbnails=true&delete_unregistered_thumb nail_files=false&update_usages_in_posts=false	1	2.66 s
/wp-json/regenerate-thumbnails/v1/regenerate/45? only_regenerate_missing_thumbnails=true&delete_unregistered_thumb nail_files=false&update_usages_in_posts=false	1	2.60 s
/wp-json/regenerate-thumbnails/v1/regenerate/44? only_regenerate_missing_thumbnails=true&delete_unregistered_thumb nail_files=false&update_usages_in_posts=false	1	2.49 s

# Top Maximale Upstream Tijd

Upstream tijd is de totale tijd die NGINX (en upstream servers) nodig heeft om een verzoek te verwerken en een antwoord te sturen. De tijd wordt gemeten in seconden, met milliseconde resolutie.

PAD	AANVRAGEN	TIJD
/wp-cron.php?server_triggered_cronjob	36 971	123.05 s
/google0064c97a60a07595.html	3	57.51 s
/wp-admin/admin-ajax.php	638	8.25 s
/wp-json/regenerate-thumbnails/v1/regenerate/42? only_regenerate_missing_thumbnails=true&delete_unregistered_thumb nail_files=false&update_usages_in_posts=false	1	3.03 s
/wp-json/regenerate-thumbnails/v1/regenerate/43? only_regenerate_missing_thumbnails=true&delete_unregistered_thumb nail_files=false&update_usages_in_posts=false	1	2.81 s
/wp-json/regenerate-thumbnails/v1/regenerate/168? only_regenerate_missing_thumbnails=true&delete_unregistered_thumb nail_files=false&update_usages_in_posts=false	1	2.80 s

## Je site zou gehackt kunnen zijn

Als je problemen hebt met het opsporen van een prestatieprobleem, kan het heel goed zijn dat je site is gehackt, geïnfecteerd is met malware of een DDoS-aanval ondergaat. Dit kan de snelheid van je site en zelfs het reactievermogen van je WordPress dashboard beïnvloeden. In deze gevallen raden wij het volgende aan:

1. Implementeer een proxyserver en WAF zoals Cloudflare of Sucuri.
2. Blokkeer slechte IP-adressen met behulp van de bovenstaande diensten of als je een Kinsta klant bent, kan je ook IP-adressen in ons MyKinsta-dashboard blokkeren.
3. Je kunt ook geo-blocking implementeren. Sommige landen zijn echt slecht als het gaat om de kwaliteit van het verkeer dat ze genereren. Als je aangevallen wordt, moet je misschien het hele land blokkeren, tijdelijk of permanent.

## Problemen oplossen met foutcodes (HTTP-statuscodes)

HTTP-statuscodes zijn als een korte notitie van de webserver dat boven aan een webpagina wordt geplakt. Het is geen onderdeel van de webpagina. In plaats daarvan is het een bericht van de server dat je laat weten hoe het ging toen het verzoek om de pagina te bekijken door de server werd ontvangen. Deze kunnen van onschatbare waarde zijn als het gaat om het oplossen van problemen!

Hoewel er meer dan 40 verschillende statuscodes zijn, staan hieronder de meest voorkomende waar WordPress gebruikers mee worstelen. We hebben tutorials beschikbaar op [kinsta.com](https://kinsta.com) met instructies voor het oplossen van elk van de volgende foutcodes:

**429: “Too many requests”** Door de server gegenereerd wanneer de gebruiker te veel verzoeken in een bepaalde tijd heeft verzonden (rate limiting). Dit kan soms gebeuren door bots of scripts die proberen toegang te krijgen tot je site. In dit geval kan je proberen je WordPress login URL te wijzigen.

### 429 Too Many Requests

nginx

**500: “There was an error on the server and the request could not be completed.”** Een generieke code die gewoon “interne serverfout” betekent. Er ging iets mis op de server, en de gevraagde bron werd niet geleverd. Deze code wordt meestal gegenereerd door plugins, foutieve PHP of zelfs het verbreken van de verbinding met de database.

### Error establishing a database connection

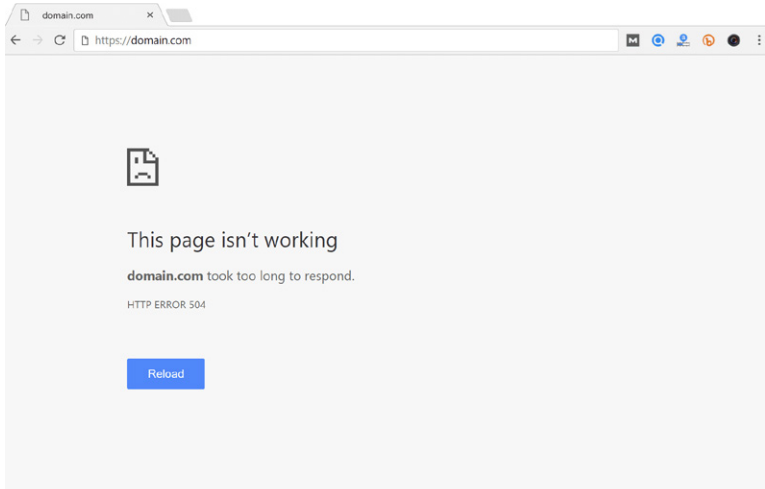
**502: “Bad Gateway.”** Deze foutcode betekent meestal dat de server een ongeldig antwoord heeft ontvangen van een andere. Soms duurt een vraag of verzoek te lang en wordt deze door de server geannuleerd of afgebroken en de verbinding met de database wordt verbroken.



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

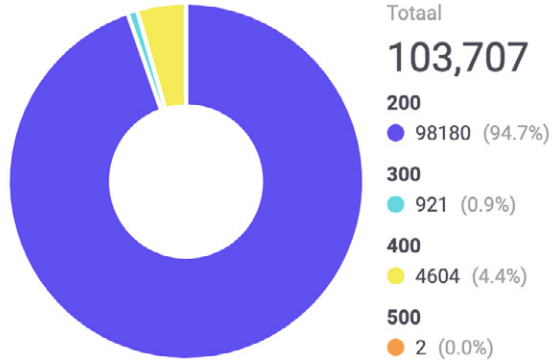
Het verzoek kan nu niet worden verwerkt. Deze code kan worden geretourneerd door een overbelaste server die niet in staat is om extra verzoeken af te handelen.

**504: “The server, acting as a gateway, timed out waiting for another server to respond.”** Deze code krijg je wanneer er twee servers betrokken zijn bij het verwerken van een verzoek, en de eerste server te lang moet wachten op de tweede server om te reageren.



Je kunt meer vinden over deze HTTP-responscodes in onze *MyKinsta Analytics*-tool. Ons rapport over *Responscodelyse* geeft je een overzicht van de verdeling van de HTTP-statuscodes die voor de gevraagde bronnen worden gebruikt.

## Responsecode analyse



Het rapport met de responsstatistieken laat je het totale aantal redirects zien, de fouten, het succespercentage en de foutverhouding. Elke WordPress site heeft doorgaans een kleine foutratio; dit is volkomen normaal.

## Response statistieken

**921**

Redirects

**4 606**

Fouten

**95.6%**

Succespercentage

**4.4%**

Foutpercentage

Vervolgens zijn er voor elk type foutcode breakdown reports, zoals 500 fouten, 400 fouten, redirects, etc.

# 500-error analyse



HOOFDSTUK 14:

# Aanbevelingen over back-end optimalisatie

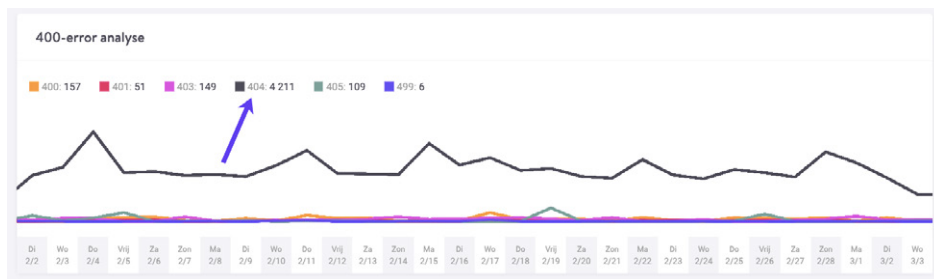




Nu bekijken we een aantal manieren waarop je WordPress kunt versnellen door het optimaliseren van de back-end. Back-end omvat meestal alles wat volledig door de server wordt afgehandeld, zoals PHP, HTTP-cacheheaders, GZIP-compressie, enz.

## Maak een lichte 404 pagina aan

We hebben zelf gezien dat zeer dynamische sites doorgaans veel 404 fouten genereren. Je website genereert misschien wel meer dan je denkt! Onze MyKinsta analysetool kan je helpen bij het bepalen van de exacte hoeveelheid (zie hieronder).



De reden dat deze fouten slecht zijn, is dat veel 404 pagina's zeer veel middelen gebruiken. Voor een zeer dynamische WordPress site, wil je een zware 404 pagina

vermijden. Maak een eenvoudige 404-template die de database zo min mogelijk belast. En repareer de 404-fouten, want dit is niet alleen resource-intensief, het is gewoonweg slecht voor de gebruikerservaring.

## Verhoog PHP workers

Je hebt misschien nog nooit van PHP workers gehoord, maar veel hosts, inclusief Kinsta, zorgen zo voor het beperken van inkomende verzoeken (in plaats van je te beperken door CPU of RAM, wat typisch is voor shared hosting providers).

PHP workers bepalen **hoeveel gelijktijdige aanvragen je site op een bepaald moment kan verwerken**. Simpel gezegd wordt elk ongecacht verzoek voor je website afgehandeld door een PHP worker. Als je bijvoorbeeld 4 verzoeken krijgt die precies op hetzelfde moment op je site komen en je site heeft 2 PHP workers, dan worden twee van die verzoeken verwerkt terwijl de andere twee in de wachtrij moeten blijven staan tot de eerste twee klaar zijn met de verwerking.

Vergeet niet dat we eerder hebben besproken dat één van de grootste problemen met WordPress lidmaatschapssites al die ongecachte verzoeken zijn. Dit is de reden waarom PHP workers erg belangrijk zijn, omdat ze voor elk verzoek werk moeten doen. Daarom zullen deze sites meestal extra PHP workers nodig hebben om ervoor te zorgen dat elk verzoek zonder vertraging wordt verwerkt en met succes wordt afgerond.

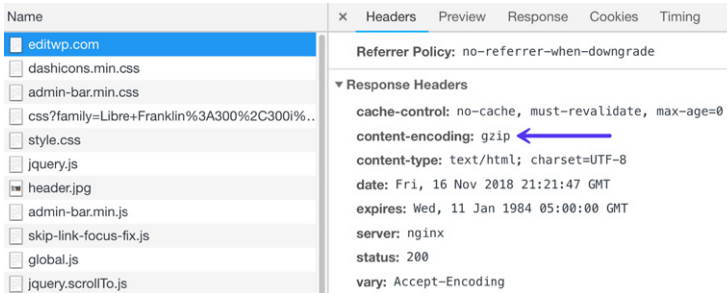
Wat gebeurt er als je je PHP workers continu maximaliseert? In principe begint de wachtrij oudere aanvragen te negeren, wat kan leiden tot 500-fouten op je site. Elk van de hostingpakketten van Kinsta bevat een vooraf bepaald aantal PHP workers. Als je moeite hebt om in te schatten wat je site nodig heeft, kan je altijd met ons verkoop- of supportteam chatten.

## Gebruik maken van GZIP-compressie

GZIP is een bestandsformaat en een softwaretoepassing die wordt gebruikt voor het comprimeren en decomprimeren van bestanden. GZIP compressie is server-

side ingeschakeld, en maakt het mogelijk om de grootte van je HTML-, stylesheet en JavaScript-bestanden verder te verkleinen.

Wanneer een webbrowser een website bezoekt, controleert hij of de webserver GZIP heeft ingeschakeld door te kijken of de content-encoding: gzip HTTP header bestaat. Als de header wordt gedetecteerd, vraagt deze de gecomprimeerde en kleinere bestanden op. Zo niet, dan worden de ongecomprimeerde bestanden opgehaald. Als je GZIP niet hebt ingeschakeld, zie je waarschijnlijk waarschuwingen en fouten in snelheidstestprogramma's zoals Google PageSpeed Insights en GTmetrix.



Het inschakelen van GZIP-compressie kan helpen om de grootte van je webpagina te verminderen, wat de tijd om te downloaden aanzienlijk kan verminderen, het dataverbruik voor de klant kan verminderen en de tijd om je pagina's voor het eerst te renderen kan verbeteren. Dit is nu vrij standaard bij de meeste hostingproviders, maar niets verrast ons meer op dit punt.

## Hotlink-bescherming inschakelen

Het concept van hotlinking is vrij eenvoudig. Je vindt een afbeelding ergens op het internet en gebruikt de URL van de afbeelding direct op je site. Deze afbeelding wordt op je website getoond, maar wordt wel vanaf de oorspronkelijke locatie geladen. Dit is erg handig voor de hotlinker, maar het is eigenlijk diefstal omdat het gebruik maakt van de resources van de hotlinked site. Het is alsof we in onze auto zouden stappen en weg zouden rijden met benzine dat we van de auto van onze buurman hebben overgeheveld.

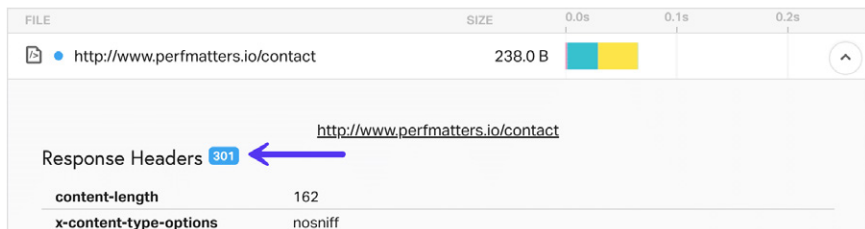
***Hotlinking is als wegrijden met benzine dat je uit de auto van je buurman hebt overgeheveld.***

Hotlinking kan een **groot beslag leggen op de middelen van de server**. Stel je voor dat je op een gedeelde WordPress host zit en Huffington Post plotseling naar je afbeeldingen linkt. Je zou op je site kunnen gaan van een paar honderd aanvragen per uur naar een paar honderdduizend. Dit kan zelfs resulteren in een schorsing van je hostingaccount. Dit is een reden om niet alleen een high-performance host te gebruiken (die dit soort pieken aankan), maar ook om hotlinkbeveiliging in te schakelen, zodat dit niet gebeurt.

## Minimaliseer redirects en voeg ze toe op serverniveau

Te veel redirects zijn altijd iets waar je op moet letten. Eenvoudige redirects zoals een enkele 301 redirect, HTTP naar HTTPS, of www naar non-www (vice versa) zijn prima. En dat is vaak nodig op bepaalde delen van je website. Echter, elk heeft een kostenpost op de prestaties van je site. En als je begint met het op elkaar stapelen van redirects, is het belangrijk om te beseffen hoe ze je site beïnvloeden. Dit geldt voor pagina's en post redirects, image redirects, alles.

Een redirect genereert een 301 of 302 op de response header status.

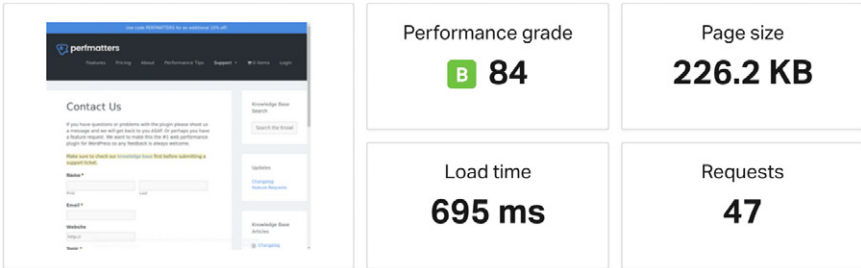


Hoeveel impact hebben redirects op je site? Laten we een kleine test doen. Eerst doen we een snelheidstest op onze contactpagina: `https://perfmatters.io/contact/`. Zoals je hieronder kunt zien, krijgen we een totale laadtijd van **417 ms**.

The image shows a performance test results dashboard for the website `perfmatters.io`. The dashboard includes a screenshot of the 'Contact Us' page and four key performance indicators (KPIs):

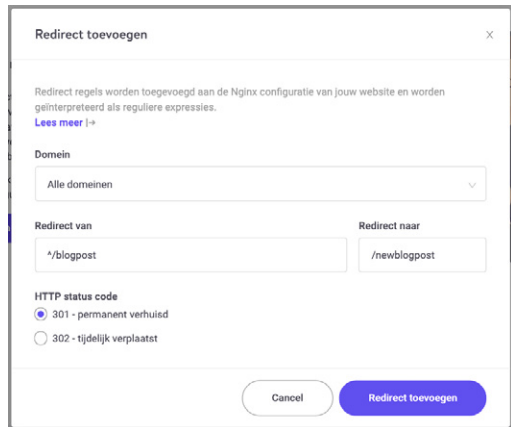
- Performance grade:** B 85
- Page size:** 225.5 KB
- Load time:** 417 ms
- Requests:** 45

We passen dan de URL iets aan en voeren nog een snelheidstest uit om de impact van meerdere redirects te zien. <http://www.perfmatters.io/contact>. Zoals je kunt zien, duurt het nu 695 ms om dezelfde pagina te laden. Dat is een stijging van 66%. Yikes!



Performance grade <b>B 84</b>	Page size <b>226.2 KB</b>
Load time <b>695 ms</b>	Requests <b>47</b>

Het gebruik van gratis WordPress plugins om redirects te implementeren kan soms leiden tot prestatieproblemen omdat de meeste van hen de `wp_redirect`-functie gebruiken, die extra code en bronnen vereist. Sommige van hen voegen ook auto-loaded data toe aan je `wp_options` tabel, wat de database bloot verhoogt. Het toevoegen op serverniveau is waar het moet gebeuren. Wij raden je aan dat te doen via MyKinsta met onze redirect tool.



Redirect toevoegen

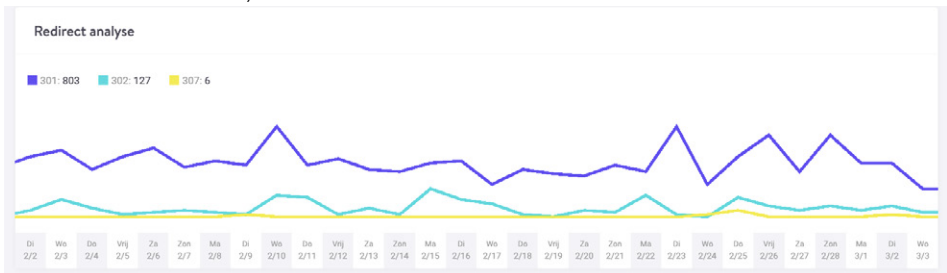
Redirect regels worden toegevoegd aan de Nginx configuratie van jouw website en worden geïnterpreteerd als reguliere expressies.  
[Lees meer](#) |>

Domain:

Redirect van:       Redirect naar:

HTTP status code  
 301 - permanent verhuisd  
 302 - tijdelijk verplaatst

Je kunt ook zien hoeveel redirects er op je sites zijn in onze MyKinsta analysetool. Bekijk het totaal aantal 301, 302 en 304.



## Laat Cronjobs niet uit de hand lopen...

CRONjobs (WP-Cron) worden gebruikt om terugkerende taken voor je WordPress site in te plannen. Na verloop van tijd kunnen deze echter uit de hand lopen en prestatieproblemen veroorzaken. Je kunt de gratis WP Cronrol plugin (beschikbaar op WordPress.org) gebruiken om alle Cronjobs op je site te controleren.

We hebben ook prestatieproblemen gezien met de ingebouwde Cron handler van WordPress: WP-Cron. Als een site niet genoeg PHP workers heeft, en er komt een verzoek binnen, dan zal WordPress de cron laten oppakken maar de cron moet wachten op de PHP worker en zit dus gewoon te wachten. Een betere aanpak is om WP-Cron uit te schakelen en in plaats daarvan de systeem cron te gebruiken. Dit wordt zelfs aanbevolen in het officiële Plugin handboek.

Om WP-Cron uit te schakelen, voeg je het volgende toe aan je `wp-config.php` bestand, net voor de regel die zegt “That’s all, stop editing! Happy blogging.” Opmerking: Dit schakelt het uit wanneer een pagina laadt niet wanneer je het direct via `wp-cron.php` aanroept.

```
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

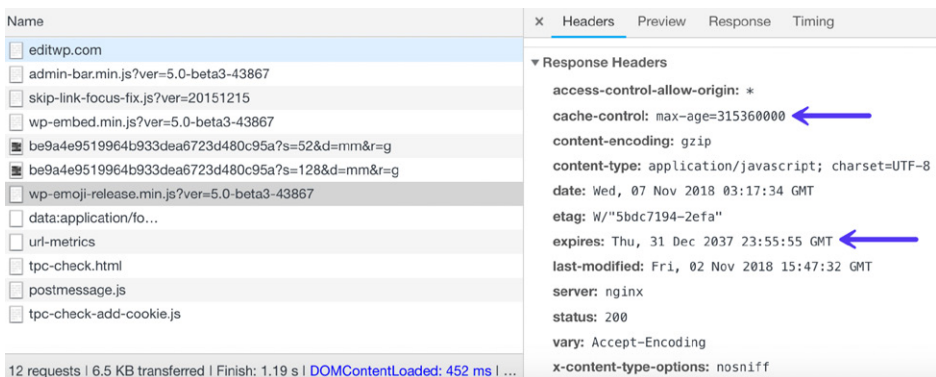
Je zult dan `wp-cron.php` moeten inplannen vanaf je server. Als je een Kinsta klant bent, zijn de systeemcrons al ingeschakeld en draaien ze standaard om de 15 minuten. Je kunt de frequentie ook laten verhogen door een beroep te doen op ons supportteam. Als je bekend bent met SSH, kan je de servercrons vanaf de commandoregel beheren.

# Cache-Control toevoegen en de kopteksten laten verlopen (Cache Lengte bepalen)

Elk script op je WordPress site moet een HTTP-cache header hebben (of zou dit moeten hebben). Dit **bepaalt wanneer de cache van het bestand verloopt**. Om dit te verhelpen, zorg je ervoor dat je WordPress host de juiste cache-control- en expires-headers heeft. Als je dit niet doet, zul je waarschijnlijk waarschuwingen zien over het toevoegen van expiratie headers of het gebruik van browser caching in snelheidstestprogramma's.

Terwijl de `cache-control` header de client-side caching inschakelt en de `max-age` van een bron instelt, wordt de `expires` header gebruikt om een specifiek punt in de tijd te specificeren dat de bron niet meer geldig is. Hoewel beide headers samen kunnen worden gebruikt, hoef je niet per se beide headers toe te voegen. `cache-control` is nieuwer en meestal de aanbevolen methode.

Kinsta voegt automatisch HTTP-cacheheaders toe op alle serververzoeken, en als je een CDN gebruikt, zullen ze waarschijnlijk ook deze headers voor je toevoegen.



Name	Headers	Preview	Response	Timing
editwp.com				
admin-bar.min.js?ver=5.0-beta3-43867				
skip-link-focus-fix.js?ver=20151215				
wp-embed.min.js?ver=5.0-beta3-43867				
be9a4e9519964b933dea6723d480c95a?s=52&d=mm&r=g				
be9a4e9519964b933dea6723d480c95a?s=128&d=mm&r=g				
wp-emoji-release.min.js?ver=5.0-beta3-43867				
data:application/fo...				
url-metrics				
tpc-check.html				
postmessage.js				
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

Als je server deze headers mist, kan je ze handmatig toevoegen.

## Cache-Control header toevoegen in Nginx

Je kunt `cache-control` headers in Nginx toevoegen door het volgende toe te voegen aan de serverconfiguratie van je serverlocatie of `-blok`.

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

## Toevoegen expires header in Nginx

Je kunt `expires` headers in Nginx toevoegen door het volgende toe te voegen aan je serverblok. In dit voorbeeld kan je zien hoe je verschillende expiratietijden kunt opgeven op basis van bestandstypen.

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

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

## Cache-Control Header toevoegen in Apache

Je kunt `cache-control` headers in Apache toevoegen door het volgende toe te voegen aan je `.htaccess` bestand. Fragmenten van code kunnen bovenaan of onderaan het bestand worden toegevoegd (voor `# BEGIN WordPress` of na `# END WordPress`).

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



## Toevoegen expires header in Apache

Je kunt `expires` headers in Apache toevoegen door het volgende toe te voegen aan je `.htaccess` bestand.

```
## 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 ##
```

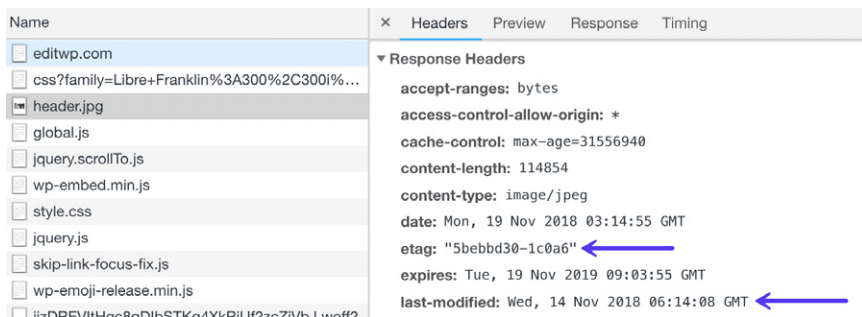
Het is ook belangrijk om op te merken dat je alleen HTTP-cacheheaders kunt toevoegen aan bronnen op je eigen server. Als je een waarschuwing krijgt over dat je misschien gebruik moet maken van browser caching op een verzoek van een derde partij, is er niets dat je kunt doen, omdat je geen controle hebt over hun server. Veelvoorkomende boosdoeners zijn onder andere het Google Analytics-script en marketingpixels, zoals Facebook en Twitter.

Als je dit probeert te verhelpen met het Google Analytics-script, kun je het lokaal hosten of op je CDN (hoewel dit niet officieel wordt ondersteund) met een plugin zoals Perfmatters (beschikbaar op [perfmatters.io](https://perfmatters.io)) of WP Rocket (beschikbaar op [wp-rocket.me](https://wp-rocket.me)).

## Last-Modified en ETag headers toevoegen (Cache valideren)

Vervolgens hebben we nog twee sets van headers, `last-modified` en `etag`.

Terwijl de `cache-control` en `expires` headers helpen de browser te bepalen **of het bestand** is veranderd sinds de laatste keer dat het werd aangevraagd (of beter gezegd, ze valideren de cache). De `last-modified` en `etag` headers **valideren** beide **en stellen de lengte van de cache** in en moeten worden opgenomen in elk antwoord van de origin server. Als deze niet goed zijn ingesteld, zie je misschien een waarschuwing dat je "Specify a cache validator" nodig hebt.



Als de headers niet worden gevonden, zal het elke keer een nieuwe aanvraag voor de resource genereren, wat de belasting van je server verhoogt. Het gebruik van caching headers zorgt ervoor dat latere verzoeken niet van de server hoeven te worden geladen, waardoor bandbreedte wordt bespaard en de prestaties voor de gebruiker worden verbeterd.

Kinsta voegt automatisch de bovenstaande headers toe op alle serveraanvragen, en als je een CDN gebruikt, zullen ze waarschijnlijk ook deze headers voor je toevoegen. Net zoals bij `cache-control` en `expires`, kan je deze HTTP-headers niet handmatig instellen op externe bronnen.

### Last-Modified header

De **last-modified** header wordt over het algemeen automatisch verzonden vanaf de server. Dit is een header die je **in het algemeen niet handmatig hoeft toe te voegen**. Het wordt verstuurd om te zien of het bestand in de cache van de browser is gewijzigd sinds de laatste keer dat het is opgevraagd. Je kunt de headeraanvraag in Pingdom bekijken of Chrome DevTools gebruiken om de waarde van de laatst gewijzigde header te zien.

## Etag Header

De ETag-header is ook zeer vergelijkbaar met de laatst gewijzigde header. Het wordt ook gebruikt om de cache van een bestand te valideren. Als je Apache 2.4 of hoger uitvoert, wordt de ETag-header al automatisch toegevoegd met behulp van de FileETag-richtlijn. En wat NGINX betreft, is de ETag-header standaard ingeschakeld sinds 2016.










Je kunt de ETag-header handmatig in Nginx inschakelen met behulp van de volgende code.

```
etag on
```

## Voeg een Vary: Accept-Encoding header toe

De `vary: Accept-Encoding` header moet worden opgenomen op elke respons van de origin server, omdat het de browser vertelt of de client al dan niet gecomprimeerde versies van de inhoud aankan. Als dit niet goed is ingesteld, zie je misschien een waarschuwing dat je “Specify a Vary: Accept-Encoding Header.”

Laten we bijvoorbeeld zeggen dat je een oude browser hebt zonder gzip-compressie en dat je er een moderne browser bij hebt. Als je geen gebruik maakt van de `vary: Accept-Encoding` header kan je webserver of CDN de ongecomprimeerde versie kunnen cachen en die per ongeluk aan de moderne browser leveren, wat op zijn beurt de prestaties van je WordPress site schaadt. Door gebruik te maken van de header kan je ervoor zorgen dat je webserver en/of CDN de juiste versie levert.

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 voegt automatisch de bovenstaande headers toe op alle serveraanvragen, en als je een CDN gebruikt, zullen ze waarschijnlijk ook deze headers voor je toevoegen. Net zoals bij de andere cacheheaders die we hierboven hebben besproken, kan je deze header niet handmatig instellen op externe bronnen.

## Voeg Vary: Accept-Encoding header in Apache toe

Je kunt de vary: Accept-Encoding header in Apache toevoegen door het volgende toe te voegen aan je .htaccess bestand.

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

## Voeg Vary: Accept-Encoding Header in Nginx toe

Je kunt de vary: Accept-Encoding header in Nginx toevoegen door de volgende code toe te voegen aan je configuratiebestand. Alle Nginx-configuratiebestanden bevinden zich in de map /etc/nginx/. Het primaire configuratiebestand is /etc/nginx/nginx.conf.

```
gzip_vary on
```

## De WordPress-geheugenlimiet in wp-config.php wijzigen

Zoals vermeld in de WordPress Codex, kan je met de optie WP\_MEMORY\_LIMIT vanaf WordPress Versie 2.5 de maximale hoeveelheid geheugen opgeven die door PHP kan worden verbruikt. Deze instelling kan nodig zijn als je een melding "Allowed memory size of xxxxxx bytes exhausted" ontvangt.

Standaard zal WordPress proberen het geheugen dat is toegewezen aan PHP te vergroten tot 40MB voor een enkele site en 64MB voor multisite. Ze definiëren de geheugenlimieten in het bestand `./wp-includes/default-constants.php` op regels 32 - 44.

Er is ook een PHP `memory_limit` op de server van je hostingprovider. Dit zijn twee verschillende dingen. Bij Kinsta stellen we de standaard `memory_limit` in op 256M. Als je tegen de maximale geheugengrootte aanloopt, kan je proberen de PHP-geheugenlimiet in WordPress te verhogen.

Voeg het volgende toe aan je `wp-config.php` bestand, net voor de regel die zegt “That’s all, stop editing! Happy blogging.”

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

Jan Reilink heeft ook een geweldig artikel op zijn website op [saotn.org](http://saotn.org) dat in meer detail verwijst naar dit nummer. Hij geeft ook een variatie op de code die je zou kunnen gebruiken. In plaats van het bedrag handmatig in te stellen, kun je het instellen op de PHP `memory_limit` waarde.

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

HOOFDSTUK 15:

# Tips voor front-end optimalisatie



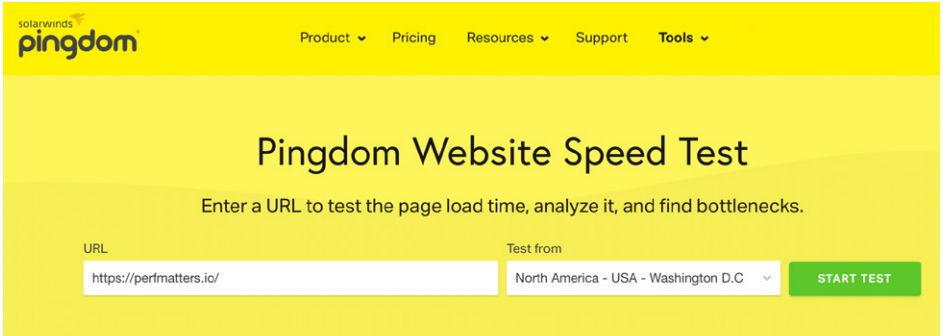
Nu bekijken we een aantal manieren om WordPress te versnellen door de front-end te optimaliseren. Bij de front-end gaat het meestal om alles wat volledig door de browser wordt afgehandeld, zoals CSS, JavaScript, afbeeldingen, enz. Dit omvat ook het analyseren van externe diensten die je op je site hebt geladen en hoe deze van invloed zijn op de totale laadtijd.

Twee van de belangrijkste doelstellingen die je moet hebben als het gaat om front-end optimalisatie zijn:

- **Vermindering van de totale omvang van je webpagina.** De grootte van de CSS, JavaScript, afbeeldingen is van belang. Een website van 4MB wordt meestal veel langzamer geladen dan een website van 1MB. Paul Calvano heeft een geweldig artikel over de impact van het gewicht van de pagina op de laadtijd en waarom het belangrijk is om ervoor te zorgen dat dat niet het enige is wat je bekijkt, want soms kan dit misleidend zijn.
- **Vermindering van HTTP-verzoeken en externe diensten.** Met HTTP/2 kunnen nu meerdere verzoeken en antwoorden tegelijkertijd worden verzonden met behulp van een enkele TCP-verbinding. Hoewel dit al geweldig is voor de prestaties, kan het verminderen van HTTP-verzoeken nog steeds helpen om je WordPress site te versnellen. Dit omvat ook het verminderen van het totale aantal externe verzoeken en diensten. Elk hiervan voegt extra vertragingen toe, zoals DNS-lookups, TLS-verbindingen en netwerk latency.

# Test je WordPress site om een startpunt vast te leggen

Als het gaat om het optimaliseren van de front-end van je site, is het altijd goed om te beginnen met een startpunt. Dit betekent meestal dat je een snelheidstest moet gaan uitvoeren. Er zijn veel verschillende tools die je hiervoor kunt gebruiken. We hebben uitgebreide gidsen over hoe je Pingdom en GTmetrix goed kunt gebruiken op onze website op [kinsta.com](https://kinsta.com).



## Your Results:

DOWNLOAD HAR

SHARE RESULT



Performance grade

**B**

**88**

Page size

**243.0 KB**

Load time

**398 ms**

Requests

**36**

Hier zijn een paar dingen om in gedachten te houden bij het testen van de snelheid:

## 1. Kies één tool en blijf daarbij

Wij zijn grote fans van Pingdom, GTmetrix, WebPageTest, PageSpeed Insights en Chrome DevTools. Het maakt echter niet zozeer uit welke snelheidstest-tool je gebruikt, maar wel of je deze consistent gebruikt. Ze hebben allemaal verschillende manieren om de snelheid te meten en te kwantificeren, dus kies één tool en houd het daarbij voor al je tests en optimalisaties. Zelfs Google zegt dat je er één moet kiezen.





*Het maakt niet uit welke snelheidstest-tool je kiest, zorg dat je er één kiest en daar al je tests mee doet.*

## **2. Ga niet voor de perfecte score.**

Veel van de tools zoals Google PageSpeed Insights hebben allemaal een soort snelheids- of prestatiescore. Het is belangrijk om te onthouden dat de score niet altijd even belangrijk is als de snelheid van je website en de waargenomen prestaties van de gebruiker. De score is er om te helpen meten hoe goed je het doet. Maar het nastreven van een perfecte 100/100 of een A score kan in sommige gevallen een verspilling van tijd zijn. En grotere sites met veel externe scripts en advertenties zullen nooit een perfecte score krijgen, wat perfect in orde is.

## **3. De locatie van je tests zijn belangrijk**

De locatie die je kiest bij het testen van de snelheid doet er nogal wat toe. Zoals we in een eerdere paragraaf al hebben besproken, is de reden hiervoor dat dit allemaal relatief is ten opzichte van de datacenter locatie die je kiest. TTFB, netwerk latency, ze komen allemaal aan bod. Dus test je site zowel vanaf een locatie die dicht bij je datacenter ligt als vanaf een locatie die ver weg is. Dit zal je ook helpen om te zien hoeveel impact een CDN kan hebben op je WordPress site.

## **4. Test meerdere keren vanwege caching**

Zoals we al eerder in de sectie over caching hebben gezegd, als de cache onlangs is gewist of is verlopen op je WordPress host of CDN, gaat het een “MISS” registreren op de HTTP-header. Dit betekent dat je website niet uit de cache komt.

FILE	SIZE	0.0s	0.1s	0.2s
https://perfmatters.io/	9.6 KB			
<a href="https://perfmatters.io/">https://perfmatters.io/</a>				
Response Headers <span style="background-color: #90EE90; 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			

Om de snelheid van je hele site goed te meten moet je alles geladen worden uit de cache, je eerste pagina, en alle bestanden moeten een “HIT” meekrijgen. Dit vereist dat je soms de snelheidstest meerdere malen moet uitvoeren. Je kunt dan het gemiddelde nemen.

FILE	SIZE	0.0s	0.1s	0.2s
https://perfmatters.io/	9.6 KB			
<a href="https://perfmatters.io/">https://perfmatters.io/</a>				
Response Headers <span style="background-color: #90EE90; 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			

Laten we nu kijken naar een aantal front-end optimalisaties die je kunt maken op je WordPress site.

## Querystrings verwijderen

Een veel voorkomende waarschuwing of aanbeveling die mensen zien in snelheidstest-tools is dat je de querystrings moet verwijderen. Waar gaat dit over? Nou, hoe het werkt is dat je CSS- en JavaScript-bestanden meestal de bestandsversie

aan het einde van hun URL's hebben, zoals `https://domain.com/file.min.css?ver=4.5.3`. Sommige servers en proxy servers zijn niet in staat om querystrings te cachen. Dus door ze te verwijderen, kan je soms je caching verbeteren.

Je kunt deze handmatig verwijderen door de volgende code in het bestand `functions.php` van je thema te laten vallen.










```
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');
```










Of je kunt een premium plugin gebruiken zoals Perfmatters (beschikbaar op [perfmatters.io](https://perfmatters.io)) om querystrings met één klik te verwijderen.

## Met querystrings

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
 jizDREVIthGc8qDIbSTKq4XkRIUf2zcZiVbJ.woff2	200

Hier is een voorbeeld van het laden van scripts met querystrings.

# Zonder querystrings

Name	Status	
 editwp.com	200	
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600%2C600i...	200	
 style.css ←	200	Hier is een voorbeeld van het laden van scripts na het verwijderen van de querystrings.
 jquery.js	200	
 header.jpg	200	
 skip-link-focus-fix.js	200	
 global.js	200	
 jquery.scrollTo.js	200	
 jizDREVIthGc8qDlbSTKq4XkRiUf2zcZiVbJ.woff2	200	

Voordat je direct op je site querystrings gaat strippen, is het belangrijk om te weten waarom querystrings worden gebruikt. Versiebeheer op bestanden wordt meestal gebruikt door WordPress-ontwikkelaars om cachingproblemen te omzeilen.

Als een plugin-ontwikkelaar bijvoorbeeld een update uitvoert en `style.css` wijzigt van `?ver=4.6` naar `?ver=4.7`, wordt het behandeld als een volledig nieuwe URL en wordt het niet in de cache geplaatst. Als je de querystrings verwijdert en een plugin bijwerkt, kan dit ertoe leiden dat de gecacheerde versie nog wordt gebruikt. In sommige gevallen zou dit het uiterlijk van je site kunnen breken totdat de cache resource verloopt of de cache volledig wordt geleegd.

Ook kunnen sommige CDN's querystrings cachen. De Kinsta CDN kan en doet dat standaard. Dus als je een Kinsta klant bent, worden bestanden met querystrings al gecached.

## Elimineer Render-Blocking JavaScript en CSS

Een waarschuwing over het blokkeren van JavaScript en CSS kan verschijnen wanneer je bestanden hebt die voorkomen dat de pagina zo snel mogelijk wordt geladen. Specifieke JS en CSS zijn soms voorwaardelijk, wat betekent dat ze niet verplicht zijn om de inhoud boven de vouw weer te geven. Je kunt voorkomen dat ze render-blocking worden door gebruik te maken van `async` en `defer` attributen.

Opportunity	Estimated Savings
1 Eliminate render-blocking resources	<div style="width: 100%;"><div style="width: 80%;"></div></div> 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> .	

Om render-blokkering van JavaScript en CSS te elimineren, moet je het volgende doen:

## Verwijder JS van het critical rendering path

Het verplaatsen van JavaScript uit het critical rendering path wordt meestal gedaan door het toevoegen van ofwel het `defer` ofwel het `async`-attribuut aan het de HTML-elementen die de JavaScript-bestanden aanroepen.

- Het **async-attribuut** vertelt de browser om meteen te beginnen met het downloaden van het bestand zonder HTML-parsing te vertragen. Zodra het bestand beschikbaar is, wordt de HTML-parsing gepauzeerd zodat het bestand kan worden geladen.
- Het **defer-attribuut** vertelt de browser dat hij moet wachten met het downloaden van het bestand totdat de HTML-parsing voltooid is. Zodra de browser klaar is met de HTML zal hij alle uitgestelde scripts downloaden en weergeven in de volgorde waarin ze in het document verschijnen.

## Optimaliseren van de levering van CSS-bestanden

Het optimaliseren van de levering van CSS betekent dat je moet uitzoeken hoe je ervoor kunt zorgen dat het niet blokkerend werkt.

- Identificeer de stijlen die nodig zijn om de inhoud boven de vouw te renderen en lever deze stijlen inline met de HTML.
- Gebruik CSS alleen voorwaardelijk op apparaten wanneer dat nodig is.
- Laad de resterende CSS asynchroon.

Bovenstaande kan soms een lastig proces zijn en vergt zeker wat tweaks op basis van de scripts die je op je site hebt geladen. Hier zijn een paar WordPress plugins die kunnen helpen:

- Autoptimize (Gratis, beschikbaar op WordPress.org)
- Async JavaScript (Gratis, beschikbaar op WordPress.org)
- Hummingbird (Gratis, beschikbaar op WordPress.org)

Voor een meer gedetailleerde uitleg en beschrijving, raden wij je aan onze post over het elimineren van render-blocking JavaScript en CSS te bekijken.










## Combineer externe CSS en JavaScript in WordPress

De waarschuwing over het combineren van externe CSS wordt meestal getoond bij het gebruik van een CDN omdat je je CSS-bestanden op een extern domein, zoals `cdn.domein.com`, host. In het verleden is een snelle manier om dit te verhelpen het koppelen van je CSS-bestanden, of ze te combineren zodat ze in één enkele aanvraag worden geladen.

Als je HTTPS gebruikt met een provider die HTTP/2 ondersteunt, is deze waarschuwing niet meer zo relevant als vroeger. Met HTTP/2 kunnen nu meerdere CSS-bestanden parallel worden geladen over een enkele verbinding. En meer dan 86% van de browsers ondersteunt HTTP/2.

Maar dat betekent niet noodzakelijkerwijs dat deze optimalisatie volledig dood is. In sommige gevallen hebben we gezien dat dit de WordPress sites nog steeds versnelt. Het hangt af van de grootte van de bestanden en hoeveel er zijn. Daarom is dit een optimalisatie die we nog steeds aanraden om te testen op je site.

Een van de eenvoudigste manieren om je externe CSS- en JavaScript-bestanden te combineren is met de gratis Autooptimize-plugin (beschikbaar op WordPress.org). Na het combineren zie je een “`autooptimize_XXXX.css`” of “`autooptimize_XXXX.js`” bestand. Het ondersteunt ook CDN. Je kunt dit ook doen met de WP Rocket plugin (verkrijgbaar bij `wp-rocket.me`).

Name	Status	Domain
 editwp.com	200	editwp.com
 autooptimize_23d90d8a1eaacef4363d9e02c50ae363.css ←	200	mk0editwpn6...
 css?family=Libre+Franklin%3A300%2C300%2C400%2C400...C600...	200	fonts.googlea...
 jquery.js?ver=1.12.4	200	mk0editwpn6...
 header.jpg	200	mk0editwpn6...
 autooptimize_3fde0f2c0341cdc5afd7cc0c0e403f92.js ←	200	mk0editwpn6...
 jizDREVIHgc8qDlbSTKq4XkRIUf2zcZIVbj.woff2	200	fonts.gstatic.c...
 jizAREVIHgc8qDlbSTKq4XkRI24_Sl0q1vjitOh.woff2	200	fonts.gstatic.c...
 jizAREVIHgc8qDlbSTKq4XkRI20-Sl0q1vjitOh.woff2	200	fonts.gstatic.c...

# Gebruik minimalisatie op HTML, CSS en JavaScript

We kunnen de hoeveelheid gegevens die de browser moet downloaden verminderen door HTML-, CSS- en JavaScript-bronnen te minimaliseren. Minimalisatie is het proces van het verwijderen van onnodige karakters zoals commentaar en witruimte uit de broncode. Deze karakters zijn uiterst nuttig in de ontwikkeling, maar ze zijn nutteloos voor de browser om de pagina te renderen.

## Niet-geminimaliseerde HTML

Hier is een voorbeeld van niet-geminimaliseerde HTML-code.

```
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' />
```

## Geminimaliseerde HTML

Hier is een voorbeeld van geminimaliseerde HTML-code.

```
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://mk0editwpn6qbfuw5o9.kinstacdn.com/wp-
7 content/cache/autoptimizate/css/autoptimizate_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-
```

Je kunt de gratis Autoptimize plugin of WP Rocket gebruiken om je bestanden eenvoudig te minimaliseren.

## Gebruik Cookie-Free domeinen

Wanneer je inhoud zoals afbeeldingen, JavaScript, CSS verstuurt, is er geen reden om hierbij een HTTP-cookie mee te sturen, omdat het extra overhead creëert. Zodra de server een cookie voor een bepaald domein instelt, moeten alle volgende HTTP-verzoeken voor dat domein het cookie bevatten. Deze waarschuwing wordt meestal gezien op sites met een groot aantal verzoeken.

Vaak kun je deze waarschuwing negeren omdat nieuwe protocollen zoals HTTP/2 dit nu minder belangrijk maken. De kosten van een nieuwe verbinding zijn meestal duurder dan het streamen van alles over dezelfde verbinding.

Een eenvoudige manier om deze waarschuwing te verhelpen is door gebruik te maken van een CDN-aanbieder die zowel cookies kan negeren als verwijderen, waardoor de client de header van de Set-Cookie response header niet meer ontvangt. KeyCDN is een CDN-aanbieder die deze functie wel aanbiedt. Standaard kan je zien dat de volgende twee opties zijn ingeschakeld. Dit is een eenvoudig alternatief zonder dat je hoeft te rommelen met het verplaatsen en configureren van je site om statische bestanden te serveren vanuit een apart subdomein.

### 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.**

Als je Cloudflare gebruikt, kan je geen cookies uitschakelen op bestanden die via hun netwerk worden bediend. CloudFlare neemt hun eigen beveiligingscookie op in je header. Ook hier zijn deze cookies zeer klein en de gevolgen voor de prestaties zijn uiterst minimaal. Maar als je CloudFlare gebruikt, is er geen manier om deze waarschuwing te omzeilen.











Een tweede manier om dit te omzeilen is het herconfigureren van je WordPress site om de statische bestanden van een nieuw domein of subdomein te leveren.



## Embedden in WordPress uitschakelen

Toen WordPress 4.4 uitkwam, was de oEmbed-functie toegevoegd aan WordPress core. Hierdoor kunnen gebruikers YouTube-video's, tweets en vele andere bronnen op hun sites insluiten door simpelweg een URL te plakken, die WordPress automatisch omzet in een insluiting en een live-voorvertoning in de visuele editor biedt. Met de update is WordPress zelf een oEmbed-provider geworden.

Deze functie is nuttig voor veel mensen, en daarom wil je deze misschien ingeschakeld houden. Dit betekent echter dat het ook een extra HTTP-verzoek genereert op je WordPress site om het `wp-embed.min.js` bestand te laden. En dit laadt op je hele site. Terwijl dit bestand slechts 1,7 KB is, tellen dit soort dingen in de loop van de tijd op. Het verzoek zelf is soms groter dan de downloadgrootte van de inhoud.

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

Er zijn een paar verschillende manieren om embedden in WordPress uit te schakelen. Je kunt dit doen met de volgende plugins:












- Disable Embeds (Gratis, beschikbaar op [WordPress.org](https://wordpress.org/plugins/disable-embeds/))
- Perfmatters (verkrijgbaar bij [perfmatters.io](https://perfmatters.io/))

Of je kunt embedden met code uitschakelen. Vanwege de lengte is deze te vinden op onze website <https://kinsta.com/knowledgebase/disable-embeds-wordpress/>.

## Emojis uitschakelen in WordPress

Vergelijkbaar met embeds, in WordPress 4.2 werd ondersteuning voor emojis

toegevoegd in WordPress core, ook voor oudere browsers. Het grote probleem hierbij is dat het een extra HTTP-verzoek genereert op je WordPress site om het `wp-emoji-release.min.js` bestand te laden. En dit laadt op de hele site. Hoewel dit bestand slechts 10,5 KB is, is het nutteloos als je geen emojis op je site gebruikt.

Name	Status	Domain
 editwp.com	200	editwp.com
 global.js	200	mk0editwpn...
 css?family=Libre+Franklin%3A300%2C300%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
 jizDREVIHgc8qDlbSTKq4XkRiUf2zcZiVbJ.woff2	200	fonts.gstatic...

Er zijn een paar verschillende manieren om emojis uit te schakelen in WordPress. Je kunt het doen met de volgende plugins:

- Disable Emojis (Gratis, beschikbaar op WordPress.org)
- Perfmatters (verkrijgbaar bij perfmatters.io)

Of je kunt emojis met code uitschakelen. Vanwege de lengte is deze te vinden op onze website <https://kinsta.com/knowledgebase/disable-emojis-wordpress/>.

## Zo maak je WordPress reacties sneller of schakel je ze uit

Een drukbezette reactie-sectie op een site kan veel prestatieproblemen veroorzaken. Denk maar eens aan de middelen die nodig zijn om reacties te leveren:

- De database wordt opgevraagd om bestaande reacties op te halen.
- Voor elke nieuwe reactie worden database-items aangemaakt.
- Reacties en reactie-metadata worden ontvangen en verwerkt door de browser van een bezoeker.
- Externe bronnen, zoals Gravatars, worden opgevraagd, gedownload en geladen (waarvoor een aparte DNS-lookup nodig is).
- In veel gevallen moeten grote JavaScript en jQuery bronnen worden gedownload en verwerkt om het reactiesysteem te laten werken zoals het hoort.

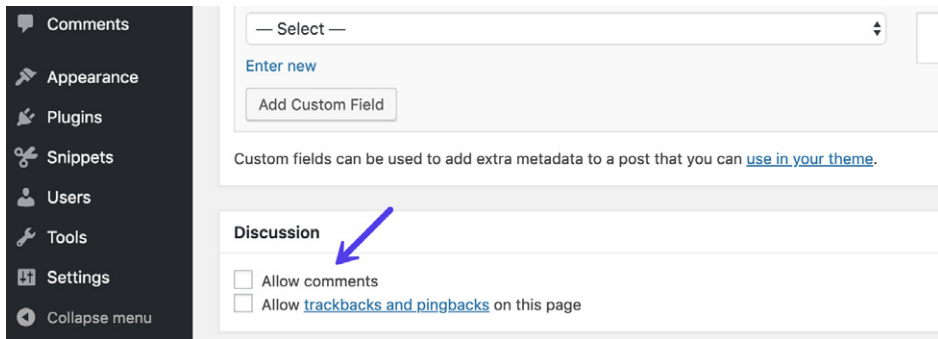
Hier zijn vier verschillende opties die je kunt uitvoeren om de reacties van WordPress te versnellen:

## Optie 1 - Uitschakelen van reacties

Als je site niet veel reacties krijgt en je denkt dat ze geen waarde toevoegen, kan het beter zijn om de reacties helemaal uit te schakelen. Vergeet niet dat reacties je SEO kunnen beïnvloeden, aangezien Google deze meestal als extra inhoud op de pagina zal crawlen, dus je moet enkel reacties van hoge kwaliteit goedkeuren. Bekijk deze drie eenvoudige manieren om reacties uit te schakelen:

### 1. Schakel reacties uit in de WordPress instellingen

Je kunt reacties op een pagina of een bericht individueel uitschakelen door deze te bewerken en de optie “Reacties toestaan” onder de sectie “Discussie” uit te schakelen. Als je deze sectie niet ziet, moet je misschien naar boven scrollen en deze onder de “Schermopties” inschakelen.



Je kunt ook overall reacties uitschakelen. Je kunt dit doen door in “Reacties” te klikken onder Instellingen in je WordPress dashboard en dan de optie “Mensen toestaan om reacties op nieuwe artikelen te plaatsen” uit te schakelen. Het nadeel van deze methode is echter dat het geen invloed heeft op oudere artikelen, alleen op nieuwe.

Als je meer controle nodig hebt en gewoonweg alle opmerkingen in één keer wilt uitschakelen, raden wij je aan één van de volgende methoden te gebruiken.

Discussion Settings Help ▾

**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.)*

**Other comment settings**

- Comment author must fill out name and email
- Users must be registered and logged in to comment
- Automatically close comments on articles older than  days
- Show comments cookies opt-in checkbox.
- Enable threaded (nested) comments  levels deep
- Break comments into pages with  top level comments per page and the  page displayed by default
- Comments should be displayed with the  comments at the top of each page

## 2. Reacties uitschakelen met Plugin

Een van de gemakkelijkste manieren om reacties uit te schakelen is het gebruik van een gratis plugin zoals Disable Comments (beschikbaar op WordPress.org). Je kunt reacties overal uitschakelen met één klik, of ze uitschakelen op berichten, pagina's of media-posttypes.

## 3. Uitschakelen reacties met code

Je kunt reacties met code uitschakelen door eenvoudigweg de commentaarsectie te verwijderen uit het verschijnen in je WordPress thema met eenvoudige HTML-commentaren.

In je `page.php`-bestand vervang je eenvoudigweg de volgende code:

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

Met deze code:

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

Dat haalt het van je pagina's. En om het te verwijderen van je blogberichten, navigeer je naar je single.php bestand en vervang de volgende code:

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

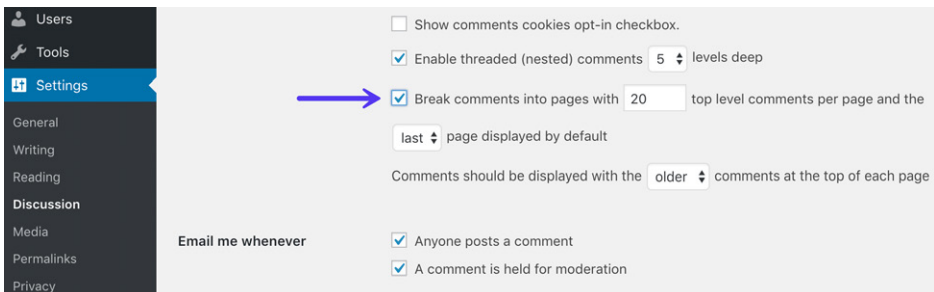
Met deze code:

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

## Optie 2 - Optimaliseren van standaard WordPress reacties

De tweede optie zou zijn om het standaard WordPress reactiesysteem te optimaliseren. Een optie is het aantal reacties verminderen dat met het laden van de pagina's getoond wordt.

1. Ga naar Instellingen / Discussie in het WordPress dashboard.
2. Zoek de sectie Overige reactie-instellingen.
3. Schakel het selectievakje naast "Reacties over meerdere pagina's verdelen met" in en voeg een waarde toe voor het aantal reacties dat je wilt weergeven bij het laden van de pagina.

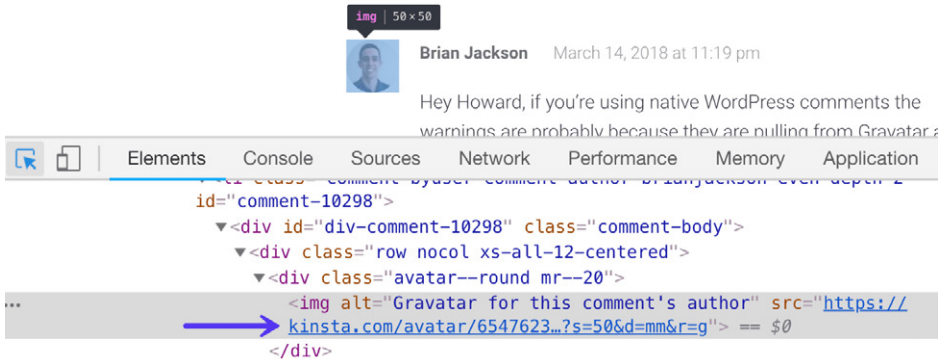


The screenshot shows the WordPress Settings - Discussion section. The 'Break comments into pages with' checkbox is checked and highlighted with a blue arrow. The value '20' is entered in the adjacent input field. Other settings include 'Enable threaded (nested) comments' (checked), 'Comments should be displayed with the' set to 'older', and 'Email me whenever' (checked).

Een andere optie die je hebt is het hosten van Gravatars op je CDN. Dit is de aanpak die we bij Kinsta hanteren.

Normaal gesproken, wanneer WordPress reacties worden geladen, zorgt elke unieke Gravatar voor een HTTP request. Dus als een pagina wordt geladen met reacties van 50 verschillende mensen, zullen 50 HTTP requests nodig zijn om al die Gravatars te downloaden. Zoals je je kunt voorstellen, kan dit invloed hebben op de snelheid van je pagina. Om nog maar te zwijgen van het feit dat we hebben gezien dat de externe DNS lookup naar gravatar.com soms traag is en in sommige gevallen zelfs een time-out geeft

Als je kijkt naar Gravatars op de Kinsta blog, kan je zien dat ze vanaf kinsta.com (inclusief onze CDN) laden.



## Optie 3 - Gebruik van een reactiesysteem van een tussenpartij

De derde optie is het gebruik van een reactiesysteem van een tussenpartij. Als je site wordt gehost op een goedkope shared server, dan kan het gebruik van een reactiesysteem van een tussenpartij het laden van pagina's met veel reacties versnellen. Het is hetzelfde idee als afbeelding optimalisatie, het werk uitbesteden. Als je site wordt gehost bij Kinsta of een andere webhost van uitstekende kwaliteit, zal het overschakelen naar een tussenpartij niet veel verschil maken om de snelheid van je website te helpen en kan het zelfs vertragen.



Bekijk deze twee verschillende manieren om RSS-feeds uit te schakelen in WordPress:

## 1. RSS-feed uitschakelen met Plugin

De eerste manier om een WordPress RSS-feed uit te schakelen is door gebruik te maken van een gratis plugin zoals Disable Feeds (beschikbaar op WordPress.org). Deze plugin schakelt alle RSS/Atom-feeds op je WordPress-website uit door alle aanvragen door te sturen.

Je kunt ook gebruik maken van een premium plugin zoals Perfmatters (ontwikkeld door een teamlid van Kinsta, beschikbaar op perfmatters.io), waarmee je zowel RSS-feeds als RSS-feedlinks kan uitschakelen, samen met andere optimalisaties voor je WordPress site.

## 2. RSS-feed met code uitschakelen

Als je geen plugin wilt gebruiken, kan je ook de RSS-feeds met code uitschakelen. Zet de volgende code in het bestand `functions.php` van je thema.

```
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 genereert ook links naar de RSS-feeds in de header van je webpagina. Je kunt nog een stap verder gaan en deze links verwijderen uit de HTML-code van je pagina's. Zet de volgende code in het bestand `functions.php` van je thema.

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



# Gebruik Prefetch en Preconnect

Resource tips en directives zoals prefetch en preconnect kunnen een geweldige manier zijn om WordPress achter de schermen te versnellen.

## Prefetch

Met DNS prefetch kan je domeinnamen prefetchen (een DNS-lookup op de achtergrond uitvoeren) voordat een gebruiker op een link klikt, wat op zijn beurt kan helpen om de prestaties te verbeteren. Dit wordt gedaan door een `rel="dns-prefetch"` tag toe te voegen in de header van je WordPress site.

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

Enkele veel voorkomende items waar je DNS prefetching voor kan gebruiken is je CDN URL, Google fonts, Google Analytics, etc.

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

Prefetch wordt ook ondersteund door de meeste moderne browsers. Er zijn een paar manieren om dit te implementeren.

1. Je kunt de code aan je header toevoegen met behulp van een gratis plugin zoals Head, Footer en Post Injections (beschikbaar op WordPress.org).
2. Je kunt de code handmatig toevoegen aan het `header.php` bestand van je thema.
3. Implementeer DNS met behulp van een premium plugin zoals Perfmatters (beschikbaar op perfmatters.io). Klik gewoon op het tabblad "Extra's" in de Perfmatters plugin en voeg domeinen toe. Formaat: `//domain.tld` (één per regel)

**Perfmatters Settings**

Options | CDN | Google Analytics | **Extras** | License | Support

### Extras

*Extra options that pertain to Perfmatters plugin functionality.*

Script Manager ?

DNS Prefetch ?

## Preconnect

Met Preconnect kan de browser verbindingen opzetten voordat een HTTP-verzoek wordt ingediend, waardoor er geen round-trip-latency meer is en gebruikers tijd kunnen besparen.

Preconnect is een belangrijke tool in je optimalisatie toolbox... Het kan vele dure roundtrips van je request path elimineren - in sommige gevallen vermindert het de request latency met honderden en soms zelfs duizenden milliseconden. - Ilya Grigorik



Ilya Grigorik

Dit wordt gedaan door een `rel="preconnect"` tag toe te voegen in de header van je WordPress site.

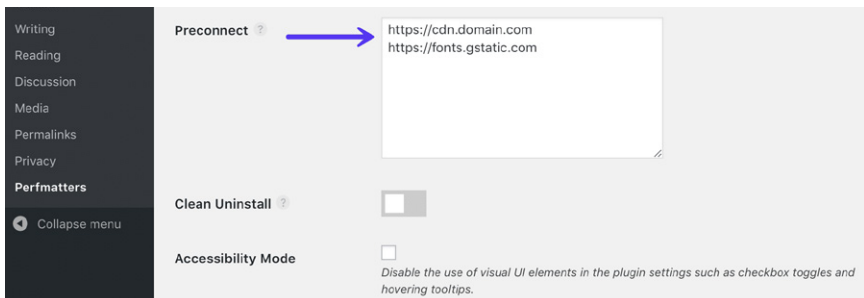
```
<link rel="preconnect" href="//domain.com">
```

Een paar voorbeelden van dingen die je misschien hiervoor wilt gebruiken zijn je CDN URL of Google Fonts.

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

Preconnect wordt ondersteund door de meeste moderne browsers, met uitzondering van Internet Explorer, Safari, IOS Safari en Opera Mini. Er zijn een paar manieren om dit te implementeren.

1. Je kunt de code aan je header toevoegen met behulp van een gratis plugin zoals Head, Footer en Post Injections (beschikbaar op WordPress.org).
2. Je kunt de code handmatig toevoegen aan het `header.php` bestand van je thema.
3. Implementeer preconnect met een plugin zoals Perfmatters (beschikbaar op [perfmatters.io](https://perfmatters.io)). Klik gewoon op de “Extra’s” tab in de Perfmatters plugin en voeg domeinen toe. Formaat: `schema://domain.tld` (één per regel).



## Scripts per pagina/post uitschakelen

Een andere zeer krachtige manier om WordPress te versnellen is het doorzoeken van elk verzoek dat op je pagina's en berichten wordt geladen. Je zult waarschijnlijk uiteindelijk scripts vinden die op je hele site worden geladen en dat zou niet zo moeten zijn.

Je kunt gebruik maken van een premium plugin zoals Perfmatters (beschikbaar op [perfmatters.io](https://perfmatters.io)) die een “Script Manager” functie heeft ingebouwd. Hiermee kan je scripts (CSS en JavaScript) per pagina/post of zelfs op je hele site uitschakelen met

een enkele klik. Deze plugin is ontwikkeld door een teamlid van Kinsta.

Een paar voorbeelden van waar dit voor kan worden gebruikt:

- De populaire Contact Form 7 plugin laadt zichzelf op elke pagina en op elke post. Je kunt dit gemakkelijk overal uitschakelen met één klik en alleen op je contactpagina inschakelen.
- Social media sharing plugins moeten alleen worden geladen voor je berichten. Je kunt deze eenvoudig overal uitschakelen en alleen laden op bepaalde (aangepaste) berichttypes.
- De Table of contents plugin (TOC) wordt op elke pagina en op elk bericht geladen. Met de scriptsmanager kan je eenvoudig controleren waar je deze wilt laden.
- Als je een upgrade naar WordPress 5.0 hebt uitgevoerd en geen gebruik maakt van de Gutenberg block editor, gebruik je misschien nog steeds de klassieke editor of een andere editor, dan zijn er twee extra front-end scripts die op je hele site worden toegevoegd en die je kan uitschakelen:

```
/wp-includes/css/dist/block-library/style.min.css en /wp-includes/css/  
dist/block-library/theme.min.css.
```

## Waarom zijn sommige plugins op deze manier geprogrammeerd?

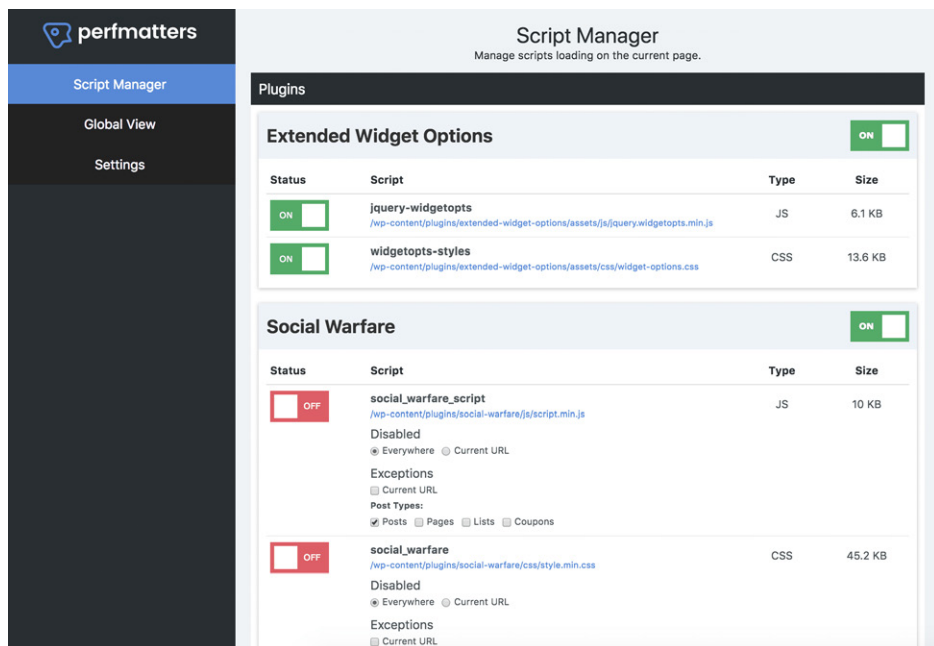
Je vraagt je misschien af waarom alle plugin ontwikkelaars hun scripts niet gewoon laden als de plugin op de pagina wordt gedetecteerd? Nou, het is iets ingewikkelder dan dat. Bijvoorbeeld, als je een plugin hebt zoals Contact Form 7, dan heeft het ook shortcodes waarmee je het overal kunt plaatsen. Die kan je ook in een widget gebruiken. Met WordPress is het veel moeilijker om gegevens hiervan op te vragen als je scripts niet laadt, in tegenstelling tot het opvragen van gegevens van de bericht of pagina metadata.

Dit is vaak te wijten aan het proberen te voorkomen van gebruikersproblemen. Hoe minder kans op fouten, hoe minder tickets en ondersteuning de plugin ontwikkelaars zullen hebben. Echter, veel plugins op de markt zijn gemaakt voor prestaties. Helaas maakt het aantal downloads en aantal gebruikers het coderen voor gebruiksvriendelijkheid soms tot een prioriteit.

# De Script Manager nader bekijken

We geven je een kleine rondleiding door de Script Manager. Na het aanklikken in je werkbalk krijg je alle scripts te zien die op die huidige URL geladen zijn, zowel JavaScript- als CSS-bestanden. Je hebt dan de volgende opties:

1. **Status Aan** (standaard instelling)
2. **Status Uit:** overall uitschakelen (je kunt dan kiezen op welke berichttypes je het wilt inschakelen, samen met de huidige URL)
3. **Status Uit:** alleen op de huidige URL uitschakelen (dit is erg handig voor gebruik op je homepage)
4. **Status Uit:** Uitzonderingen (huidige URL, posttype of archief)



The screenshot shows the 'Script Manager' interface. The sidebar on the left contains 'perfmatters', 'Script Manager', 'Global View', and 'Settings'. The main content area is titled 'Script Manager' and 'Manage scripts loading on the current page.' It displays 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' script has additional options for 'Disabled' (Everywhere, Current URL), 'Exceptions' (Current URL), and 'Post Types' (Posts, Pages, Lists, Coupons).

Alles is **gegroepeerd onder de plugin of de themanaam**. Dit maakt het super eenvoudig om een hele plugin in één keer uit te schakelen. Normaal gesproken heeft een WordPress plugin zowel een JavaScript- als een CSS-bestand. Een WordPress

thema kan 10+ bestanden hebben. De Script Manager ondersteunt zelfs regex voor wanneer je een meer gecompliceerde URL-structuur of installatie hebt.

Nadat je de instellingen hebt geselecteerd en gewijzigd, moet je onderaan op “Opslaan” drukken. Je kunt dan testen in een website snelheidstool om er zeker van te zijn dat de scripts niet meer geladen worden op de pagina of op de post. Zorg ervoor dat je eerst je cache leegt! En als er visueel iets misgaat op je site, kun je het altijd opnieuw activeren in de instellingen om terug te keren naar normaal.

In een snelheidstest door [workup.com](http://workup.com) konden ze **de totale laadtijden met 20,2% verlagen**. Alleen al op hun homepage konden ze het aantal HTTP-verzoeken terugbrengen van 46 naar 30. Hun paginagrootte is ook afgenomen van 506,3 KB naar 451,6 KB.

## Analyse van de prestaties van tussenpartijen

In principe heeft alles wat je extern laadt vanaf je site gevolgen voor de laadtijd. Wat dit probleem nog erger maakt, is dat sommige van hen slechts soms traag zijn, wat de identificatie van het probleem nog moeilijker maakt.

Een externe dienst van een tussenpartij kan worden beschouwd als alles wat met je WordPress site communiceert van buiten je eigen server. Hier zijn een paar veel voorkomende voorbeelden die we regelmatig tegenkomen:

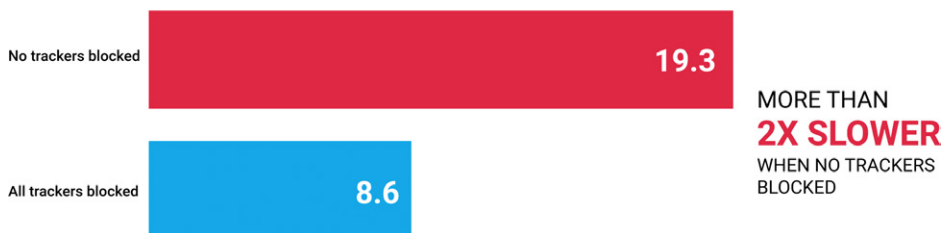
- Sociale-mediaplatforms zoals Twitter, Facebook en Instagram (widgets of conversiepixels)
- Advertentienetwerken zoals Google Adsense, Media.net, BuySellAds, Amazone-Associates
- Website analytics en tracking scripts zoals Google Analytics, Crazy Egg, Hotjar, AdRoll
- A/B-test tools zoals Optimizely, VWO, Unbounce
- WordPress reactiesystemen zoals Disqus, Jetpack, Facebook reacties
- Back-up- en beveiligingstools zoals VaultPress, Sucuri, CodeGuard
- Social sharing tools zoals SumoMe, HelloBar

- CDN netwerken zoals KeyCDN, Amazon CloudFront, CDN77 en StackPath.
- Extern gehoste Javascript

Hoeveel invloed hebben sommige van deze trackers op de prestaties? In onze eigen case study zagen we dat scripts van tussenpartijen **de laadtijden van de pagina's met 86,08% verhoogden**.

Ghostery heeft ook de top 500 Amerikaanse domeinen in Alexa gemeten, en de resultaten waren verbluffend, hoewel voor ons niet verrassend. Websites waren 2x langzamer toen er helemaal geen trackers werden geblokkeerd. Wat betekent dat deze derde partij tracking scripts één van de primaire bijdragers zijn aan het vertragen van de laadsnelheid van de pagina's op het web.

## AVERAGE PAGE LOAD TIME WITH vs. WITHOUT TRACKERS

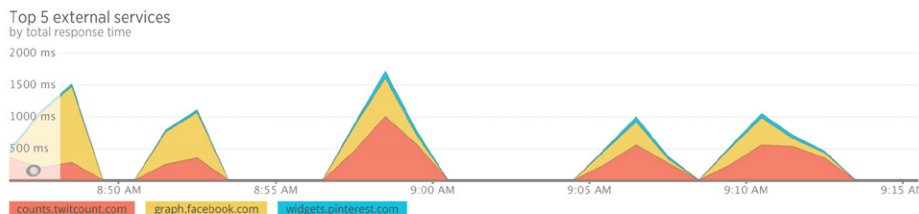


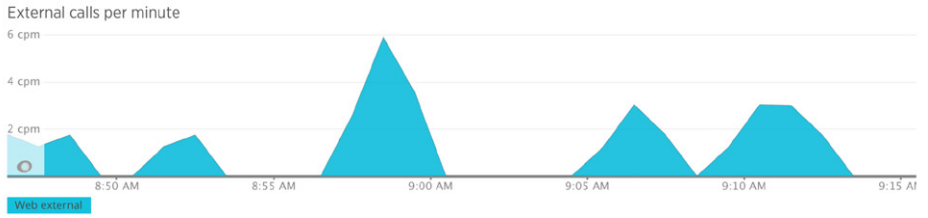
Bron: Ghostery

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

Je moet heel voorzichtig zijn op je WordPress site. Slechts één slechte API-call van een tussenpartij kan je hele site vertragen! Ja, zo zou het niet moeten werken, maar in veel gevallen gebeurt dat wel. We hebben het echt vaak gezien.

New Relic biedt een uitstekende en eenvoudige manier om je externe diensten te monitoren. In dit voorbeeld hieronder zien we dat er externe calls worden gedaan naar [twitcount.com](https://counts.twitcount.com), [graph.facebook.com](https://graph.facebook.com) en [widgets.pinterest.com](https://widgets.pinterest.com).



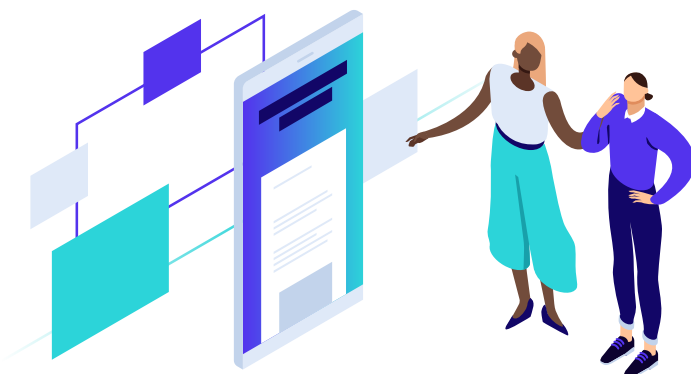


Het is belangrijk dat wanneer je een nieuwe functie of plugin aan je site toevoegt, je de externe bronnen onderzoekt die van de site worden geladen. Hoe minder hoe beter!



HOOFDSTUK 16:

# Optimaliseer altijd met Mobile-First in gedachten



**Google begon met het uitrollen van zijn mobile-first index op 26 maart 2018. Voorheen gebruikte Google's crawl-, indexerings- en rankingsystemen de desktop versie van websites. Mobile-first indexering betekent dat Googlebot nu de mobiele versie van je WordPress site gebruikt voor het indexeren en rangschikken. Dit helpt de zoekervaring voor mobiele gebruikers te verbeteren.**

Als het gaat om het mobile-first optimaliseren van je site, **is snelheid één van de belangrijkste factoren om je op te focussen**. Snelheid speelt een grote rol in alles, van usability tot bounce rates en het bepalen of potentiële kopers al dan niet terugkomen op je site. In feite is snelheid nu een landingspagina factor voor Google Search en Ads voor mobiele zoekopdrachten.

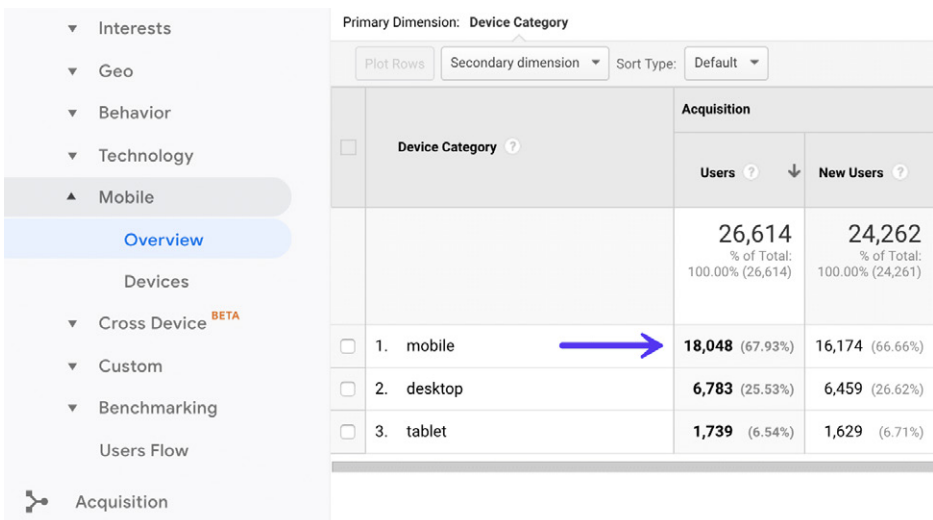
Slechte mobiele ervaringen zullen ertoe leiden dat de meerderheid van de gebruikers nooit meer terugkomt. Volgens het laatste Google page speed report was de gemiddelde tijd die een mobiele site nodig had om te laden in 2018 15 seconden. Kun je je voorstellen dat je zo lang moet wachten om een enkele pagina te laden? Verbazingwekkend.

Gebruikers vragen (en verdienen) beter. Volgens hetzelfde page speed report **verlaat 53% van de mobiele websitebezoekers pagina's die langer dan drie seconden duren om te laden.**

Trage mobiele ervaringen zijn niet dodelijk voor conversies. Ze voorkomen zelfs dat je klanten de kans krijgen om aan de slag te gaan. Als de laadtijd van de pagina's met slechts enkele seconden toeneemt, stijgt de kans dat iemand weggaat exponentieel. Hier zijn een paar dingen om rekening mee te houden bij het optimaliseren voor mobiel.

## Bekijk je mobiele verkeer

Het is altijd belangrijk om te kijken hoeveel mobiel verkeer je krijgt, omdat dit je prioriteiten een beetje kan verschuiven. Je kunt zien hoeveel mobiele apparaten je site bezoeken in Google Analytics onder "Audience / Mobile / Overview". Zoals je kunt zien op deze site, is meer dan 67% van al het verkeer van mobiel. Dat is veel!

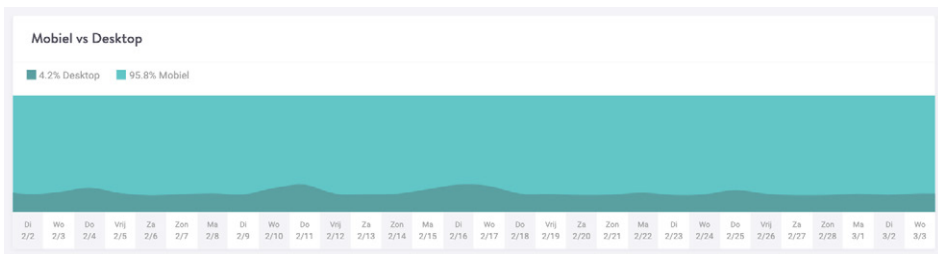


Primary Dimension: Device Category

Plot Rows Secondary dimension Sort Type: Default

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

Als je een Kinsta klant bent, kan je ook je mobiele vs. desktop verkeer bekijken in MyKinsta Analytics. Zoals je op deze site kunt zien, is meer dan 88% van het verkeer afkomstig van de desktop. Het is altijd belangrijk om te controleren en niet alleen maar te veronderstellen. Het feit dat iedereen zegt dat alles op mobiel gaat, betekent niet altijd dat het ook voor jouw site geldt. Kijk naar de data.



## Zorg ervoor dat je site responsive is

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

In 2019 kan je website maar beter responsieve zijn! Dit betekent dat het gebruik maakt van media-queries om dingen automatisch te verkleinen op mobiele apparaten. Als je dit nog steeds niet hebt gedaan, loop je waarschijnlijk al achter op je concurrentie. Alle WordPress thema's die we eerder in dit bericht hebben genoemd, zijn volledig responsive en zien er op alle apparaten geweldig uit. Gebruik de Mobile-Friendly tool van Google om te testen en ervoor te zorgen dat je website aan alle eisen voldoet.

## Wees er zeker van dat srcset werkt

In het verleden was het erg belangrijk dat je afbeeldingen op schaal uploadde en niet door CSS liet verkleinen. Dit is echter niet meer zo belangrijk aangezien WordPress 4.4 nu responsieve afbeeldingen ondersteunt (niet geschaald door CSS). WordPress maakt automatisch verschillende groottes aan van elke afbeelding die naar de mediabibliotheek wordt geüpload. Door de beschikbare formaten van een afbeelding in een srcset-attribuut op te nemen, kunnen browsers nu kiezen om de meest geschikte grootte te downloaden en de andere te negeren. Bekijk hieronder een voorbeeld van hoe je code er uit ziet.

```

```

Door de vele plugins en maatwerk van derden is het vaak voorgekomen dat dit niet goed werkte. Daarom is het belangrijk om te controleren of je afbeeldingen het srcset-attribuut goed krijgen toegevoegd met verschillende versies voor verschillende schermafmetingen. Afbeeldings optimalisatie is vanaf nu voor altijd belangrijk.

## Google AMP kan een oplossing voor je zijn.

Google AMP (Accelerated Mobile Pages Project) werd oorspronkelijk gelanceerd in oktober 2015. Het project is gebaseerd op AMP HTML, een nieuw open framework dat volledig is opgebouwd uit bestaande webtechnologieën, waardoor websites lichtgewicht webpagina's kunnen bouwen. Eenvoudig gezegd biedt het een manier om een uitgekledede versie van je huidige webpagina op te dienen. We hebben een soort van liefdes- en haatverhouding met Google AMP, en dat geldt ook voor een groot deel van de gemeenschap. We hebben dit zelf getest en hebben geen goede resultaten gezien. Maar dat betekent niet dat jij dat niet zult zien. Elke website is anders en Google AMP wordt voortdurend verbeterd.

Je kunt snel aan de slag met Google AMP op je WordPress site met één van de volgende plugins:

- AMP for WordPress (Gratis, beschikbaar op WordPress.org)
- AMP for WP (Gratis, beschikbaar op WordPress.org)

## Samenvatting

Zoals je waarschijnlijk kunt zien, zijn we geobsedeerd door alle verschillende manieren waarop je WordPress kunt versnellen. Het hebben van een snelle site helpt je rankings te verhogen, verbetert de crawlability voor zoekmachines, verbetert de conversie, verhoogt de tijd op de site, en vermindert het aantal bounces. Niet te vergeten het feit dat iedereen houdt van het bezoeken van een snelle website!

We hopen dat dit boek nuttig is geweest en dat je een paar dingen hebt kunnen meenemen en toepassen op je WordPress site.

Bekijk ook ons WordPress hosting affiliateprogramma op <https://kinsta.com/nl/affiliates>. Verdien tot \$500 voor elke verwijzing + 10% maandelijks **terugkerende levenslange commissies**. Dit is de beste affiliate-deal die je kunt vinden in de branche. Laten we samen het web versnellen!

Voor meer informatie over Kinsta kan je terecht op <https://kinsta.com/nl>

**Bedankt voor het lezen!**



De snelheid van je WordPress website is belangrijker dan je denkt. Het maakt niet uit wat voor soort site je gebruikt; of het nu een WooCommerce winkel is of een blog, de laadtijden van de pagina's hebben een directe relatie met:

- Betere keyword rankings (snelheid is een factor die Google gebruikt).
- Verhoogde conversie.
- Verbeterde indexatie door zoekmachines zoals Google en Bing.
- Langere gemiddelde tijd doorgebracht op je site.
- Verlaagde bounce rates.

Om nog maar te zwijgen van het feit dat het hebben van een snelle website een betere algemene ervaring voor je bezoekers oplevert.

WordPress heeft nu meer dan een derde van alle websites op het internet in handen! Dit is geweldig, maar het betekent ook dat er duizenden verschillende thema's, plugins en technologieën naast elkaar moeten bestaan. Voor de alledaagse WordPress-gebruiker kan dit al snel veranderen in een nachtmerrie wanneer een site een knelpunt begint te vertonen. Velen weten niet waarom of zelfs niet waar ze moeten beginnen met het oplossen van problemen.

Het team van Kinsta houdt zich dagelijks bezig met dit soort problemen en heeft het allemaal gezien! In deze gids kijken we naar de stappen die je nu kunt nemen om je WordPress sites te versnellen. Alles van beginnerstips tot geavanceerde strategieën, je gaat zeker iets nuttigs vinden in deze gids. Het bevat meer dan 15 jaar ervaring en harde lessen die we hebben geleerd.