

WIE MAN SEINE

WORDPRESS

SEITE

BESCHLEUNIGT



Brian Jackson

KINSTA

Managed WordPress Hosting

Wie man seine WordPress Seite beschleunigt

(DER ULTIMATIVE GUIDE)



Für die aktuellste Version scanne bitte den QR-Code oben oder gehe zu:
<https://kinsta.com/de/lernen/wordpress-beschleunigen/>

Veröffentlicht von Kinsta

Übersetzt von: Alexander Gerstendörfer

Buchdesign von: Maja Szakadat



Über den Autor - Brian Jackson

Brian ist Chief Marketing Officer bei Kinsta. Er konzentriert sich auf alles, von der Entwicklung neuer Online-Wachstumsstrategien, der Erstellung von Inhalten, technischem SEO und der Öffentlichkeitsarbeit in der Community. Er schreibt gerne ausgiebige Artikel und hat schon über 2.500 Blogbeiträge veröffentlicht.

Eine von Brians größten Leidenschaften ist WordPress, das er seit über einem Jahrzehnt verwendet. Nachdem er sowohl in der Content-Delivery-Netzwerkbranche als auch heute im High-Performance-Hosting-Sektor tätig war, hat er sich nun auf die Optimierung von Web-Performance und strategischen Wegen zur Beschleunigung von Webseiten spezialisiert.

Wie man seine WordPress Seite beschleunigt

Inhaltsverzeichnis

10

WordPress
Seitentypen: statisch
oder dynamisch

13

Entscheide dich für
leistungsstarkes WordPress
Hosting

25

Wähle einen Server, der
deinen Besuchern am
nächsten gelegen ist.

34

Ein premium DNS
ist besser als ein
kostenloser DNS

38

Dein WordPress
Theme ist
entscheidend

47

Ein Überblick
über WordPress
Plugins

52

Die optimalen
WordPress
Einstellungen

58

Warum ist
Caching
so wichtig

69

Bildoptimierung
ist ein
muss

78

Die
Datenbank
optimieren

97

Nutze ein Content
Delivery Network
(CDN)

106

Medien und
Emails auslagern
falls nötig

111

Wie man Engpässe
und langsame
Plugins findet

131

Empfehlungen
zur Back-End
Optimierung

146

Tipps für die Front-End
Optimierung und für
externe Services

175

Optimiere alles mit
der Prämisse "Mobile
First" im Hinterkopf

Wir haben im Laufe der Jahre viele Tutorials veröffentlicht, die sich mit Möglichkeiten zur Optimierung und Beschleunigung von WordPress befassen. Aber manchmal kann es verwirren, wenn man versucht, alles, was man braucht, an einem Ort zu finden. Deshalb werden wir heute mit dir alles, was wir über die Beschleunigung von WordPress wissen – über 15 Jahre Erfahrung und viele Lektionen – teilen, alles in einem ultimativen Leitfaden! Egal, ob du gerade erst anfängst WordPress zu verwenden oder ein erfahrener Entwickler bist, wir versprechen dir, dass du in diesem Buch etwas Nützliches finden wirst!

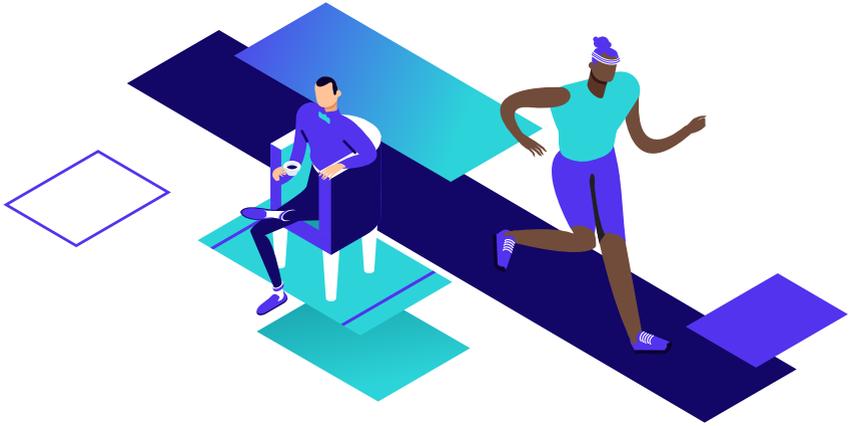
Über 33% des Webs werden heute mit WordPress betrieben. Obwohl das ein sehr guter Wert ist, bedeutet es auch, dass es Tausende von verschiedenen Themes, Plugins und Technologien gibt, die alle nebeneinander existieren müssen. Das kann Seiten oft verlangsamen. Für den alltäglichen WordPress-Benutzer kann dies schnell zu einem Albtraum werden, wenn die eigene Webseite darunter leidet. Viele wissen nicht, warum das der Fall ist oder wo sie mit der Fehlersuche beginnen sollen.

In unserem vorherigen Leitfaden zur Seitengeschwindigkeit haben wir viele Grundlagen von Performance erläutert und erfahren, wie sie einen großen Einfluss auf den Erfolg deines Unternehmens haben können. Aber heute werden wir in anwendbare Schritte eintauchen, die du im Moment umsetzen kannst, um Verbesserungen auf deinen eigenen WordPress-Seiten zu sehen. Wir werden auch einige Ressourcen teilen, die für uns von unschätzbarem Wert waren.

KAPITEL 01:

WordPress

Seitentypen: statisch oder dynamisch



Bevor wir in die Optimierungen eintauchen, ist es wichtig, zuerst zu verstehen, dass nicht alle WordPress-Seiten gleich sind. Deshalb haben viele Benutzer Probleme, da man nicht jedes Problem auf die gleiche Weise lösen kann. Wir geben WordPress-Seiten immer eine Klassifizierung: statisch oder dynamisch. Lass uns also zunächst die Unterschiede zwischen diesen beiden Webseitenarten untersuchen.

Hauptsächlich statische Seiten

Statisch sind typischerweise Webseiten wie Blogs, Webseiten für kleine Unternehmen, Webseiten für Nachrichten mit geringerem Volumen, persönliche Webseiten, fotografische Webseiten, usw.. Unter statisch verstehen wir, dass sich die Daten auf diesen WordPress-Seiten **nicht sehr oft** (vielleicht ein paar Mal am Tag) **ändern**. Selbst die meisten unserer Kinsta-Seiten würden als statische Webseiten bezeichnet werden.

Dies wird unglaublich wichtig, da viele der Anfragen direkt aus dem Cache auf dem Server mit rasanter Geschwindigkeit bedient werden können! Keine Sorge, wir werden weiter unten auf das Thema Caching in breiter Länge eingehen. Das bedeutet, dass du weniger Datenbankaufrufe haben wirst und nicht so viele Ressourcen benötigt werden, um Google Performance zu erreichen.

Hochdynamische Seiten

Auf der anderen Seite haben wir hochdynamische Seiten. Dazu gehören Webseiten wie eCommerce (WooCommerce oder Easy Digital Downloads), Community, Mitgliedschaft, Foren (bbPress oder BuddyPress) und Learning Management Systeme (LMS). Unter dynamisch verstehen wir, dass sich die Daten auf diesen WordPress-Seiten **häufig ändern** (Servertransaktionen finden alle paar Minuten oder sogar jede Sekunde statt). Das bedeutet, dass nicht alle Anfragen an den Server direkt aus dem Cache bedient werden können und zusätzliche Serverressourcen und Datenbankabfragen erforderlich sind.

Diese Webseiten haben in der Regel auch **eine große Anzahl von gleichzeitigen Besuchern und Sitzungen**. Auf einer Informations- oder Firmen-WordPress-Seite, die meist statisch ist, kann ein Besucher fünf oder zehn Minuten bleiben bis er gefunden hat, was er braucht (und das ist eine hohe Zahl, normalerweise sind die Bounce-Raten viel höher). Auf dynamischen Webseiten geschieht das Gegenteil. Besucher kommen in der Regel auf die Webseite, um sich mit etwas oder jemandem zu beschäftigen. Wenn sie einen Online-Kurs absolvieren, ist es nicht ungewöhnlich, dass sie stundenlang auf derselben Seite verweilen.

Du kannst sehen, wohin das führt. Die gleichzeitigen Besucher, die mit deinem WordPress-Host verbunden sind, summieren sich schnell. Um es noch schlimmer zu machen, hast du dann eine große Anzahl gleichzeitiger Besucher zusätzlich zu dem Problem mit „nicht zwischenspeicherbaren Inhalten“.

Du kannst nicht alle WordPress-Seiten gleich behandeln, wenn es um die Leistung geht. Statische und hochdynamische Seiten sind zwei sehr unterschiedliche Dinge.

KAPITEL 02:

Entscheide dich für leistungsstarkes WordPress Hosting



Ein WordPress Host ist ein Unternehmen, das alle Daten deiner Webseite speichert. Man meldet sich für einen Plan an und alle Bilder, Inhalte, Videos usw. werden auf einem Server gespeichert, der sich im Rechenzentrum des Hosts befindet. Der WordPress Host bietet eine einfache Möglichkeit, auf die Daten zuzugreifen, sie zu verwalten und an deine Besucher weiterzuleiten. Ziemlich einfach, oder? Nun, nicht ganz.

Es gibt drei sehr unterschiedliche Arten von WordPress Hosts, die man im Internet antrifft. Tauchen wir ein in die Vor- und Nachteile jedes Einzelnen. Es ist wichtig, dass man von Anfang an die richtige Wahl trifft, sonst verursacht das Ganze einfach nur Kopfschmerzen und man verschwendet seine Zeit.

1. Shared WordPress Hosting

Die erste und beliebteste Art von WordPress Hosting ist das, was wir „Shared Hosting“ nennen. Dazu gehören die größten Hosts der Branche wie IIG-Unternehmen wie Bluehost und HostGator sowie Anbieter wie Siteground, GoDaddy, Checkdomain, HostEurope und InMotion Hosting. Sie nutzen typischerweise cPanel, und der Durchschnittskunde zahlt in der Regel zwischen 3 und 25 EUR pro Monat.

Jeder, der diese Art von Hosting nutzt, wird irgendwann die Nachteile erleben. Es ist nur

eine Frage der Zeit. Warum? Weil **Shared Hosts dazu neigen, ihre Server zu überlasten**, was wiederum die Leistung deiner Webseite beeinträchtigen kann. Standortausfälle oder häufige Fehler vom Typ 500 sind alltägliche Umstände, die du erleben wirst, da diese Anbieter allem Limits setzen und Ressourcen bündeln müssen, um zu überleben. Oder noch schlimmer: Ausfallzeiten deiner Webseite. Auch wenn es einem nicht bewusst ist, befindet sich die eigene WordPress-Seite höchstwahrscheinlich auf demselben Server wie die 200+ anderer Kunden. Alle Probleme, die mit anderen Webseiten auftreten, können deine Seite beeinträchtigen.



Shared WordPress Hosting

Egal, wie man es dreht, abzüglich der Ausgaben, generiert kein Hosting Unternehmen mit \$3 pro Monat Gewinne. Vor allem, wenn man noch die Supportleistungen miteinbezieht. Ein Supportfall und das Ganze rentiert sich nicht mehr. Die Art und Weise, wie diese und ähnliche Unternehmen ihr Geld verdienen, ist durch Upselling und versteckte Gebühren. Diese Upsells beinhalten Vorgänge wie Migrationen, Domainregistrierungen, SSL-Zertifikate, etc. Eine weitere gängige Methode ist es, satte Anmelde-Rabatte anzubieten. Aber sobald die Verlängerung ansteht, erhält man eine hohe Rechnung.

Die meisten dieser Hosts bieten etwas an, was sie ihren „unbegrenzten Ressourcen“-Plan nennen. Ihr habt das wahrscheinlich alle schon mal gesehen. Nun, es gibt in der realen Welt nichts dergleichen wie unbegrenzte Ressourcen. Was Hosts hinter den Kulissen

tun, ist die Clients zu drosseln, die eine Menge Ressourcen verbrauchen. Dies wiederum führt dazu, dass diese wütenden Kunden gehen und Platz für mehr Kunden schaffen, die eben nicht so viele Ressourcen verbrauchen. Das führt schließlich zu einem Teufelskreis, angetrieben durch den Hosting Anbieter, der weiter billige Angebote durchdrückt und Kunden anmeldet, von denen er hofft, dass sie nicht so viele Ressourcen verbrauchen werden und denen er dann weitere Upsells verkaufen wird.

Kundenservice und Support mit Shared Hosting sind fast immer unterdurchschnittlich, da die Anzahl der Webseiten im Vergleich zu den Supportmitarbeitern schier unüberschaubar ist. Shared Hosts müssen sich sehr oft schlecht aufstellen, um überhaupt einen Gewinn zu erzielen, was in der Regel zu einem nicht zufriedenstellendem Erlebnis für den Kunden führt.



“Wenn es um Shared Hosting geht, bekommt man normalerweise das, wofür man bezahlt.”

2. DIY VPS WordPress Hosting

Die zweite Art von WordPress Hosting ist DIY VPS, oder „Do it yourself on a virtual private server“. Dieser Typus besteht üblicherweise aus Bootstrap-Startups und Benutzern mit etwas mehr Entwicklungs-, Servermanagement- und WordPress-Erfahrung. Sie sind die DIY-Fraktion. Diese Leute versuchen in der Regel immer noch, Geld zu sparen, aber sie sind auch in der Regel mit Leistung beschäftigt und erkennen ihre Bedeutung für den Erfolg ihres Unternehmens an. Gängige Methoden können beispielsweise die Verwendung eines VPS-Anbieters, eines Drittanbieters wie Digital Ocean, Linode oder Vultr sowie eines Tools wie ServerPilot zur einfacheren Verwaltung darstellen.

Ein kleines VPS von DigitalOcean beginnt bei 5 Dollar im Monat und der beliebteste Plan bei ServerPilot bei 10 Dollar im Monat. Abhängig von deinem Setup kannst du also

mit Kosten zwischen 5 und 15 EUR oder mehr pro Monat rechnen. Der DIY-Ansatz kann Kosten senken, bedeutet aber auch, dass man verantwortlich ist, wenn etwas schief läuft, und dass man selbst den Server auf Leistung optimieren muss.

Der DIY-Ansatz kann großartig sein, aber er kann auch nach hinten losgehen, wenn man nicht vorsichtig ist. Bediene dich dieser Variante nicht, wenn du nicht technisch versiert bist oder nur, weil du basteln möchtest! Deine Zeit ist Geld wert und du solltest dich auf das Wachstum deines Unternehmens konzentrieren.

3. Verwaltetes WordPress Hosting

Die dritte Form des Hostings ist das, was wir bei Kinsta anbieten, und das ist Managed WordPress Hosting. Ein Host dieser Art übernimmt für dich alle Aufgaben im Zusammenhang mit dem Backend-Server und bietet Unterstützung, wenn du sie benötigst. Letzterer ist in der Regel auf die Arbeit mit WordPress abgestimmt und beinhaltet meist Funktionen wie Staging-Umgebungen mit einem Klick und automatische Backups. Dein Support-Team wird besser informiert sein, wenn es darum geht, sich im CMS zurechtzufinden, da sie sich täglich auf eine Plattform konzentrieren.

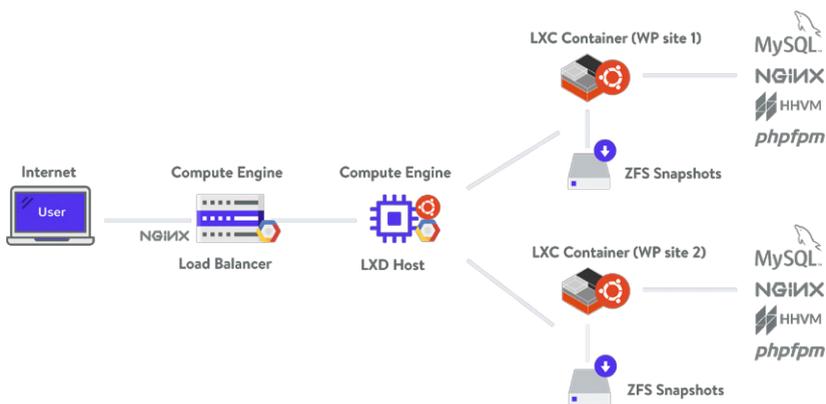
***Wenn du Zeit sparen willst, ist das Managed
WordPress Hosting der richtige Weg!***

Pläne für Managed WordPress Hosting reichen in der Regel zwischen 25 EUR und 150 EUR pro Monat oder mehr, je nach Größe deiner Webseite und der Art deiner Bedürfnisse. Große Unternehmen wie jQuery, Intuit, Plesk, Dyn, NGINX und sogar das weiße Haus verwenden WordPress, um ihre Webseite zu hosten. Einige beliebte Managed WordPress Hosts, mit denen du wahrscheinlich vertraut bist oder die du vielleicht auch gerade verwendest, sind WP Engine, Flywheel, Pressable, Media Temple, Pressidium oder Pagely.

Kinsta verfolgt einen anderen Ansatz

Kinsta bringt jedoch das Managed WordPress Hosting auf das nächste Level. Unsere Hosting-Plattform fällt in keine der traditionellen Hosting-Kategorien. Unsere gesamte Infrastruktur basiert auf der Google Cloud Plattform und unterscheidet sich von der traditionellen Shared, VPS oder dedizierten Infrastruktur.

Jede WordPress-Seite auf unserer Plattform läuft in einem isolierten Software-Container, der alle für den Betrieb der Seite erforderlichen Softwareressourcen (Linux, NGINX, PHP, MySQL) enthält. Das bedeutet, dass die Software, die jede Webseite betreibt, völlig privat ist und nicht einmal zwischen deinen eigenen Webseiten geteilt wird.



Kinsta architecture

Jeder Standortcontainer läuft auf virtuellen Maschinen in einem von mehreren GC-Rechenzentren. Jede Maschine verfügt über bis zu 96 CPUs und Hunderte von GB RAM. Hardware-Ressourcen (RAM/CPU) werden jedem Standortcontainer von unseren virtuellen Maschinen bei Bedarf automatisch zugewiesen (ein geschicktes Feature, das wir als Auto-Scaling bezeichnen).

Jedes Jahr veröffentlicht Review Signal seine WordPress Hosting-Leistungsbenchmarks, und wir sind stolz darauf, dass Kinsta sich vier Jahre in Folge als das beste Unternehmen auf allen Ebenen erwiesen hat! Und das nicht nur für ein oder zwei unserer Pläne, sondern für jeden Plan, von Starter bis zu Enterprise.

Kinsta hatte im Wesentlichen perfekte LoadStorm und Blitz Tests. Sie hatten auch keine Fehler in anderen Tests. Ich bin sprachlos was ihre Leistung anbelangt.



Kevin Ohashi
Gründer und WP Berater, ReviewSignal

Wir haben auch keine Level 1 oder Level 2 Support-Reps. Unser gesamtes Support-Team besteht aus WordPress-Entwicklern und Linux-Hosting-Technikern, von denen viele ihre eigenen Server verwalten, Themes und Plugins erstellen und auch Code für WordPress Core beigesteuert haben. Auf diese Weise erhältst du kompetente Beratung von jemandem, der mit WordPress aktiv arbeitet und entwickelt.

Du kannst mit den gleichen Support-Teammitgliedern chatten, die auch unsere Fortune-500- und Unternehmenskunden unterstützen. Wir sind so wählerisch in Bezug auf die Qualität unseres Support-Teams, dass wir nur weniger als 1% der Bewerber einstellen, die sich bewerben. Besseren Support gibt es nirgendwo sonst!

Um mehr darüber zu erfahren, warum du dich für Kinsta als deinen Managed WordPress Hoster entscheiden solltest, besuche unsere Webseite unter kinsta.com. Aber unabhängig davon, wen du als deinen Hosting-Provider wählst, solltest du immer nach diesen folgenden Serverfunktionen suchen, um sicherzustellen, dass deine Webseite so schnell wie möglich läuft.

Mit WP Engine werden grundlegende Probleme in der Regel schnell gelöst. Bei allen komplexen Fragen wird eine Lösung jedoch einige Zeit in Anspruch nehmen, und es wird viel hin und her gehen. Dies ist ein Problem, wenn man eine High-End WordPress-Site betreibt, und es gibt ein dringendes Problem, das schnell gelöst werden muss. Wenn du eine endgültige Empfehlung zwischen den beiden möchtest, ist Kinsta meiner Meinung nach besser. Sie bieten viel mehr, als sie versprechen. Sie müssen sich keine Sorgen um die Geschwindigkeit der Webseite, Ausfallzeiten, qualitativ hochwertigen Support oder andere Probleme im Zusammenhang mit dem Hosting machen.



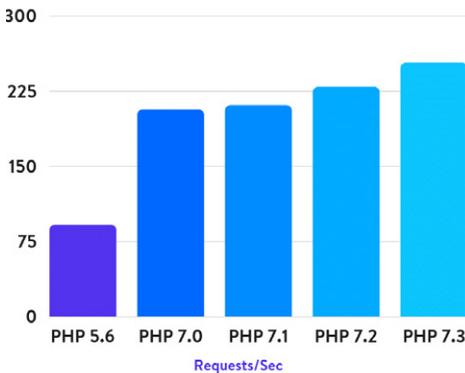
Harsh Agrawal
Preisgekrönter Blogger, ShoutMeLoud

PHP 7 oder höher für die beste Performance

PHP ist eine Open-Source-, serverseitige Scripting- und Programmiersprache, die hauptsächlich für die Webentwicklung verwendet wird. Der Großteil der Kern-Software WordPress ist in PHP geschrieben, zusammen mit deinen Plugins und Themes, was PHP zu einer sehr wichtigen Programmiersprache für die WordPress-Community macht. Du solltest sicherstellen, dass dein WordPress Host mindestens PHP 7 oder höher anbietet.

Es gibt verschiedene Versionen von PHP, die dir dein Host auf deinem Server zur Verfügung stellt, wobei das neuere PHP 7.3 enorme Leistungssteigerungen bietet.

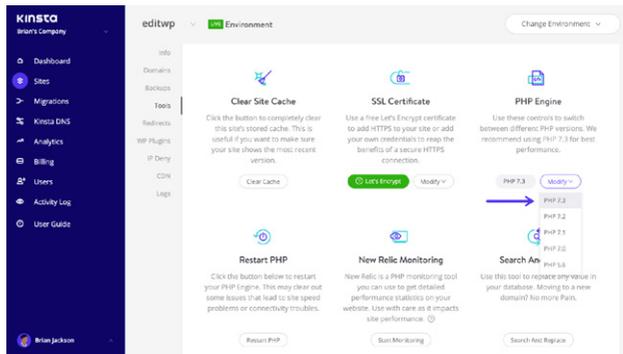
WordPress 5.0



In der Tat, in unseren aktuellen PHP-Vergleichstests, kann PHP 7.3 im Gegensatz zu PHP 5.6 dreimal so viele Anfragen (Transaktionen) pro Sekunde verarbeiten! PHP 7.3 ist auch im Durchschnitt 9% schneller als PHP 7.2. Das kann sich auch auf die Geschwindigkeit deines WordPress Admin-Dashboards auswirken.

Schnellere Geschwindigkeiten bei verbesserter Sicherheit, deshalb bietet Kinsta immer die neuesten Versionen von PHP an. Du kannst die PHP-Versionen mit einem einzigen Klick ändern.

Und halte dich fern von WordPress Hosts, die HHVM als Alternative zu PHP anbieten. HHVM ist keine geeignete Lösung mehr für das WordPress Hosting.



Suche dir einen Host der **NGINX** unterstützt

Hinter den Kulissen verwendet jeder WordPress Host einen Webserver, um deine WordPress-Seiten zu betreiben. Die gängigsten Optionen sind NGINX und Apache.

Wir empfehlen dringend, sich für einen Host zu entscheiden, der NGINX verwendet, da NGINX **seine Wurzeln in der Performance-Optimierung und in der Skalierung hat**. NGINX übertrifft in Vergleichstests oft andere gängige Webserver, insbesondere in Situationen mit statischen Inhalten oder hohen gleichzeitigen Anforderungen, weshalb Kinsta NGINX einsetzt.

Einige namhafte Unternehmen, die NGINX einsetzen, sind Autodesk, Atlassian, Intuit, T-Mobile, GitLab, DuckDuckGo, Microsoft, IBM, Google, Adobe, Salesforce, VMWare, Xerox, LinkedIn, Cisco, Facebook, Target, Citrix Systems, Twitter, Apple, Intel und viele mehr.

Laut W3Techs versorgt Apache 45,9% aller Webseiten und ist damit die am weitesten verbreitete Option. Aber wenn man sich den beliebtesten Webserver unter den hochfrequentierten Webseiten (Top 10.000) ansieht, werden 64,2% von ihnen von NGINX gehostet. Es wird von einigen der ressourcenintensivsten Webseiten verwendet, die es gibt, darunter Netflix, NASA und sogar WordPress.com.

Das Netzwerk deines Hosts macht den Unterschied

Wenn man sich für einen WordPress Host entscheidet, denkt man meistens nicht einmal daran zu fragen oder zu recherchieren, welches Netzwerk sie verwenden, aber das solltest du tun. Das Netzwerk kann einen großen Einfluss auf die Leistung deiner Webseite und sogar auf die Geschwindigkeit deines WordPress Dashboards haben. Viele Hosts werden das nicht in ihr Marketing einbeziehen, da sie sich für das günstigste Netzwerk entscheiden, um Kosten zu sparen.

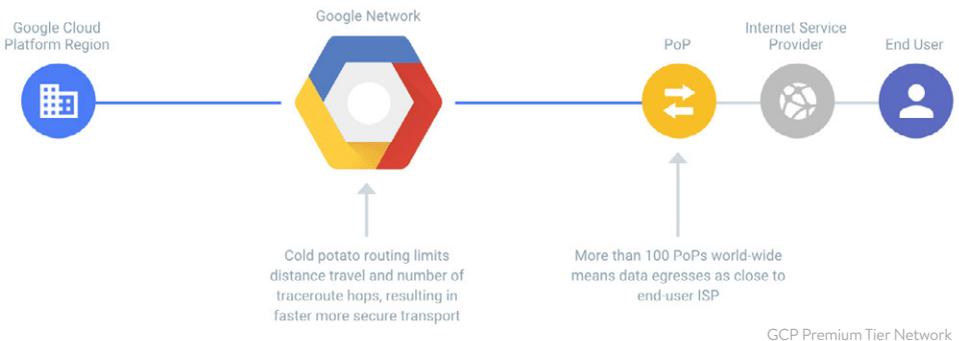
Hier sind ein paar Fragen, die du stellen solltest:

- **Über welche Netzwerke übertragen sie Daten?** Liegt der Großteil davon über öffentliche ISP-Netzwerke oder private Infrastrukturen wie Google oder Microsoft? Diese großen Anbieter verfügen über Netzwerke, die auf geringe Latenz und Geschwindigkeit ausgelegt und optimiert sind. Sie haben sogar ihre eigenen Internet-Kabel unter dem Meer!
- **Sind die von ihnen verwendeten Netzwerke redundant?** Was passiert, wenn ein Kabel versehentlich durchtrennt wird? Das passiert öfter, als man denkt.

Bereits 2017 kündigte Google sein Standard-Tier-Netzwerk an, das ein langsames Netzwerk ist, aber zu einem günstigeren Preis zu haben ist. Bei Kinsta nutzen wir das **Premium-Tier-Netzwerk** für alle unsere Hosting-Angebote. Dies ist zwar für uns ein zusätzlicher Kostenfaktor, stellt aber sicher, dass du blitzschnelle Geschwindigkeiten erhältst.

Kinsta nutzt das Premium-Tier-Netzwerk der Google Cloud Platform. Weil deine Webseiten nur das Beste verdienen.

Laut Google erreicht das Premium-Tier-Netzwerk eine verbesserte Netzwerkleistung, indem es die "Reisedauer" im öffentlichen Internet verkürzt; Pakete gelangen (und verlassen) das Netzwerk von Google um so nah wie möglich am Nutzer zu sein und reisen dann über das Backbone von Google, bevor sie zur VM gelangen. Die Standardebene liefert ausgehenden Datenverkehr von GCP ins Internet über öffentliche Verkehrsmittel (ISP-Netze) anstelle des Google-Netzwerks.



Um es anders auszudrücken, wäre es vielleicht leichter so zu verstehen:

- **Premium-Tier-Pakete verbringen mehr Zeit im Netzwerk von Google**, wobei sie weniger “herumhüpfen” und somit eine bessere Leistung erbringen (aber mehr kosten).
- Standard-Tier-Pakete verbringen weniger Zeit im Google-Netzwerk und mehr Zeit damit, in öffentlichen Netzwerken zu reisen, und liefern somit schlechtere Ergebnisse (aber weniger Kosten).

Wie stark wirkt sich das aus? Nun, für Daten, die über Kontinente transferiert werden, ist ihr Premium-Tier-Netzwerk im Durchschnitt etwa **41% schneller** als das Standard-Tier-Netzwerk. Bei Daten, die in eine nahegelegene Region (denselben Kontinent) reisen, ist die Premium-Ebene etwa **8% schneller**. Während die Vernetzung nur einen Bruchteil deiner gesamten Seitenladezeiten ausmacht, summiert sich jede Millisekunde!

Für Daten, die über Kontinente hinweg reisen, ist das Premium-Tier-Netzwerk von Google Cloud im Durchschnitt 41% schneller!

Auch die Redundanz ist entscheidend, und deshalb verwendet Google mindestens drei unabhängige Pfade (N+2-Redundanz) zwischen zwei beliebigen Standorten im Google-Netzwerk, um sicherzustellen, dass der Datenverkehr auch im Falle einer Unterbrechung zwischen den Standorten weiterläuft.

Wie du wahrscheinlich schon jetzt feststellen kannst, geht hinter den Kulissen viel vor sich, wenn es um Netzwerke geht. Stelle sicher, dass dein WordPress Host ein seriöses verwendet und sich nicht aus Kostengründen für ein langsames entscheidet.

HTTP/2 ist ein Must-Have

HTTP/2 ist ein im Jahr 2015 veröffentlichtes Webprotokoll, das entwickelt wurde, um die Bereitstellung von Webseiten zu beschleunigen. Aufgrund der Browserunterstützung

ist HTTPS (SSL) erforderlich. Wenn dein WordPress Host HTTP/2 nicht unterstützt, solltest du nach einem neuen Provider suchen. Mit der Umstellung des gesamten Webs auf HTTPS ist das nicht mehr nur ein nettes Feature, sondern eine Notwendigkeit.

Die Verbesserung der Performance mit HTTP/2 ist auf eine Vielzahl von Gründen zurückzuführen, wie z.B. die Unterstützung von besserem Multiplexing, Parallelität, HPACK-Kompression mit Huffman-Kodierung, die ALPN-Erweiterung und Server-Push. Früher gab es beim Ausführen über HTTPS einen ziemlichen TLS-Overhead, aber dank HTTP/2 und TLS 1.3 ist dies nun nicht mehr so oft der Fall.

Ein weiteres großes Plus mit HTTP/2 ist, dass du dir bei den meisten WordPress-Seiten keine Sorgen mehr um die Konkatenation (Konkatenation von Dateien) oder das Domain-Sharding machen musst. Das sind nun veraltete Optimierungen.

KAPITEL 03:

**Wähle einen
Server, der deinen
Besuchern am
nächsten ist**



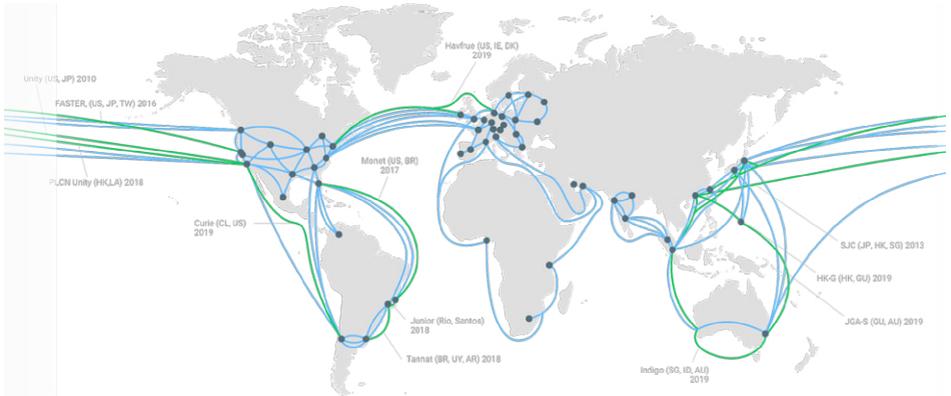
Eines der ersten Dinge, die du tun solltest, wenn du deine WordPress-Seite hostest, ist festzustellen, woher die Mehrheit deiner Besucher oder Kunden kommt. Warum ist das wichtig? Der Standort, an dem du deine Webseite hostest, spielt eine wichtige Rolle bei der Bestimmung deiner gesamten Netzwerklatenz und TTFB. Es wirkt sich auch auf SFTP-Geschwindigkeiten und die Reaktionsfähigkeit des WordPress Admin Dashboards aus.

Netzwerklatenz: Das bezieht sich auf die Zeit und/oder Verzögerung, die bei der Übertragung von Daten über ein Netzwerk anfällt, mit anderen Worten, wie lange es dauert, bis ein Datenpaket von einem Punkt zum anderen geht. Heutzutage wird dies typischerweise in Millisekunden gemessen, je nach Netzwerk können es jedoch Sekunden sein. Je näher an der Null, desto besser.

TTFB: Das steht für Zeit bis zum ersten Byte. Einfach ausgedrückt, ist das ein Maß dafür, wie lange der Browser warten muss, bis er sein erstes Byte an Daten vom Server erhält. Je länger es dauert, diese Daten zu erhalten, desto länger dauert es, deine Seite anzuzeigen. Auch hier gilt: Je näher an der Null, desto besser.

Wir werden dich nicht mit allen technischen Details in diesem Beitrag langweilen, alles, was du wissen musst, ist, **dass deine Netzwerklatenz und TTFB so niedrig wie möglich**

sein sollten. Eine der einfachsten Möglichkeiten, das zu erreichen, ist die Wahl eines Servers, der deinen Besuchern am nächsten ist. Du kannst den besten Standort bestimmen, indem du die folgenden Tipps befolgst.



Es gibt eine Reihe von verschiedenen Faktoren, die zu Verzögerungen bei der Netzwerklatenz beitragen, einschließlich:

- **Übertragungsverzögerung:** Verschiedene Arten von Medien, wie drahtlose oder Glasfaserverbindung, führen alle zu einer gewissen Verzögerung, da sie nur eine begrenzte Menge an Bits senden können. So hat beispielsweise eine Glasfaserverbindung eine geringere Übertragungsverzögerung als eine T1-Leitung.
- **Propagationsverzögerung:** Dies ist die Zeit, die das Datenpaket benötigt, um zu reisen. Dies kann Entfernungen, Netzwerkverzögerungen usw. beinhalten.
- **Warteverzögerung:** Bei Überschreitung der Bandbreite kann es zu Wartezeiten kommen, bei denen die Daten auf dem Host oder Router warten müssen. Dies kann durch Netzwerküberlastung beeinträchtigt werden.

Einige mögen argumentieren, dass die Netzwerklatenz nicht sehr wichtig ist; es kann aber Auswirkungen darauf haben, wo du deine WordPress-Seite hosten möchtest. Für Medien und Assets (wie Bilder, JavaScript, CSS, Video) kann ein CDN eine Menge der zusätzlichen Latenzzeiten beheben, indem es eine Kopie von einem näher gelegenen Server liefert. In den meisten Konfigurationen müssen Sie jedoch noch Ihren Hostingserver nach dem anfänglichen DOC-Load fragen. Und deshalb ist die Latenz wichtig!

Latenzzeiten sind besonders wichtig, wenn es um Unternehmen geht, die Besucher an einem bestimmten geografischen Standort bedienen. Nehmen wir zum Beispiel an, man hat einen E-Commerce-Shop in Sydney, und 90% der Kunden kommen aus Australien. Dein Unternehmen würde davon profitieren, die Webseite auf einem Server in Australien zu platzieren, anstatt sie in Europa oder den Vereinigten Staaten zu hosten.

Ein häufiger Irrtum ist, dass dies nach DNS-Lookup-Zeiten berechnet wird, jedoch beinhaltet die ursprüngliche Berechnung von TTFB im Netzwerk immer die Netzwerklatenz. Dies beinhaltet einen dreistufigen Prozess und Verzögerungen und Latenzen können überall dazwischen auftreten, was sich zu Ihrer gesamten TTFB summiert.

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
Explanation		298.87 ms

1. Anfrage an den Server: Wenn jemand deine Webseite besucht, wird als erstes eine HTTP-Anfrage vom Client (Browser) an den Server gesendet. In diesem Schritt gibt es eine Vielzahl von Faktoren, die zu Verzögerungen führen können. Langsame DNS-Lookup-Zeiten können zu einer längeren Zeit für die Anfrage beitragen. Wenn sich der Server geografisch weit entfernt befindet, kann dies zu Latenzzeiten auf Basis der Entfernung führen, die die Daten zurücklegen müssen. Auch wenn du komplexe Firewall-Regeln hast, kann dies die Routingzeit erhöhen. Und vergiss nicht die Internetgeschwindigkeit des Kunden.

2. Server-Verarbeitung: Nachdem die Anfrage gesendet wurde, muss der Server sie nun verarbeiten und eine Antwort generieren. Dies könnte zu einer Reihe von verschiedenen Verzögerungen führen, wie z.B. langsame Datenbankaufrufe, zu viele Skripte von Drittanbietern, die nicht Ihre erste Antwort zwischenspeichern, schlecht optimierter Code oder WordPress-Themes und ineffiziente Serverressourcen wie Festplatten-I/O oder Speicher.

3. Antwort an den Client: Nachdem der Server die Anfrage bearbeitet hat, muss er sie an den Client zurücksenden (oder besser gesagt, das erste Byte zurücksenden). Dies wird sowohl von der Netzwerkgeschwindigkeit des Servers als auch des Clients stark beeinflusst. Wenn der Kunde langsames Internet von einem Wi-Fi-Hotspot hat, wird es sich im TTFB widerspiegeln.

Im Allgemeinen ist alles unter 100 ms großartig. Google PageSpeed Insights empfiehlt eine Antwortzeit des Servers unter 200 ms. Wenn du dich im Bereich von 300-500 ms befindest, ist das ziemlich normal. Wenn es über 600 ms sind, hast du möglicherweise etwas falsch konfiguriert oder es ist an der Zeit, auf einen besseren Webstack umzusteigen. Oder folge unseren Empfehlungen unten, wie du den TTFB reduzieren kannst. Und denke daran, dass auch die SSL/TLS-Verhandlung ein Faktor sein kann.

Wir werden dich hier nicht mit allen technischen Details langweilen, alles, was du wissen musst, ist, dass deine Netzwerklatenz und TTFB so niedrig wie möglich sind. Eine der einfachsten Möglichkeiten, dies zu erreichen, ist die Wahl eines Servers, der deinen Besuchern am nächsten ist. Du kannst den besten Standort bestimmen, indem du die folgenden Tipps befolgst.

Tipp 1 – Überprüfe via Google Analytics woher deine Besucher kommen

Eines der ersten Dinge, die du tun kannst, ist die Geolokalisierung deiner Besucher in Google Analytics zu untersuchen. Du findest das unter „Publikum / Geo / Ort“.

In diesem Beispiel unten sieht man, dass über 90% des Traffics aus den Vereinigten Staaten kommt. In den meisten Fällen würdest du also deine WordPress-Seite auf einem Server in den Vereinigten Staaten platzieren wollen. Du kannst die Daten auch noch weiter nach Städten filtern. Das ist besonders wichtig, wenn du ein lokales Unternehmen bist. Aber typischerweise empfehlen wir einen zentralen Standort wie Iowa, USA.

Primary Dimension: Country		City		Continent		Sub Continent	
Secondary dimension							
Country ?		Acquisition					
		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%)				

Tip 2 – Überprüfe die eCommerce Daten

Wenn du einen eCommerce-Shop betreibst, überprüfe auch, **woher deine Kunden kommen**. Diese Kategorie an Kunden generieren deine Einnahmen, weshalb sie auch am wichtigsten sind. Das sollte mit deinem Traffic oben übereinstimmen, ist aber nicht immer der Fall. Wenn du über eine eCommerce-Dateneinrichtung oder Ziele in Google Analytics verfügst, kann man diese Informationen leicht über die Geolokalisierungsdaten legen, um eine fundiertere Entscheidung zu treffen. Oder überprüfe die in der Datenbank deiner E-Commerce-Plattform gespeicherten Standortinformationen.

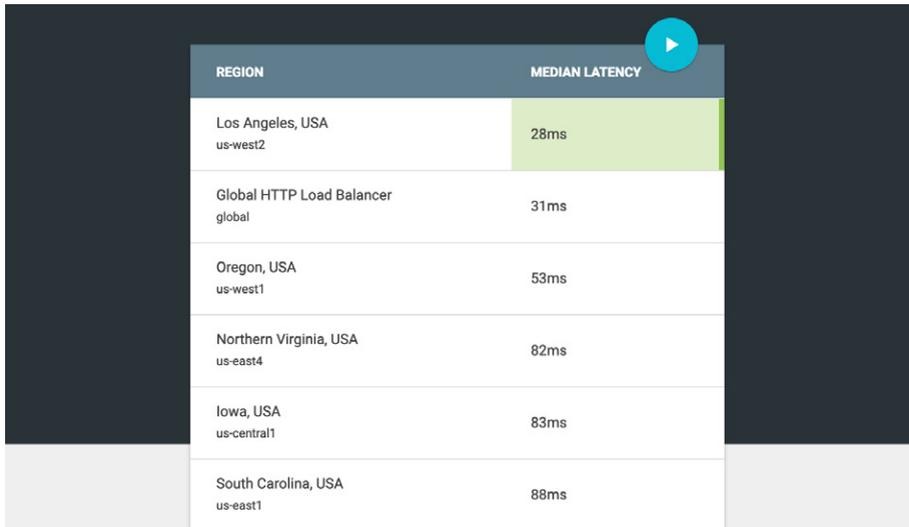
Tip 3 – Mache einen schnellen Latenz Test

Es gibt viele praktische kostenlose Tools, um die Latenzzeit von deinem aktuellen Standort für verschiedene Cloud-Anbieter zu messen. Das kann dir helfen, schnell zu beurteilen, welche Region die beste Wahl für deine Webseite sein könnte.

- **GCP Ping** (Messung der Latenzzeit für Regionen der Google Cloud Platform, einschließlich Kinsta-Server)
- **CloudPing.info** (Messung der Latenzzeiten für Amazon Web Services Regionen)
- **Azure Latency Test** (Messung der Latenzzeit für Azure-Regionen)

In diesem Beispiel unten können wir sehen, dass Oregon, USA (us-west1) der schnellste von unseren Standorten ist. Wenn du jedoch Kunden in den gesamten Vereinigten Staaten bedienst, könnte es besser sein, Iowa, USA (us-central1) zu wählen, um eine geringe Latenzzeit für Besucher von der West- und Ostküste zu gewährleisten.

Messe deine Latenzzeiten für GCP Regionen



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

Hier bei Kinsta bieten wir 20 verschiedene Rechenzentren auf der ganzen Welt an. Man kann sich leicht für eine Webseite entscheiden, die sowohl eine geringe Latenz als auch eine niedrige TTFB hat! Das trägt auch dazu bei, den Netzwerk-Hop zu reduzieren.



Weitere Möglichkeiten wie du deine Latenzzeit verringern kannst

Neben der Wahl eines nahegelegenen Serverstandortes gibt es hier noch einige andere Möglichkeiten, die Latenz zu reduzieren.

- Implementiere Caching auf deiner WordPress-Seite. In unseren Tests reduzierte das Caching unser TTFB um satte 90%!
- Nutze ein Content Delivery Network (CDN), um zwischengespeicherte Assets von POPs auf der ganzen Welt bereitzustellen. Das hilft, die Netzwerklatenz für Besucher zu minimieren, die sich möglicherweise nicht in der Nähe Ihres Hostservers befinden.
- Profitiere vom HTTP/2-Protokoll, um die Anzahl der Rundreisen durch Parallelisierung zu minimieren. HTTP/2 ist auf allen Kinsta-Servern aktiviert.
- Reduziere die Anzahl der externen HTTP-Requests. Jeder von ihnen kann seine eigene zusätzliche Latenzzeit haben, basierend auf dem Standort seines Servers.
- Verwende Prefetch und Prerender, um Aufgaben hinter den Kulissen auszuführen, während die Seite geladen wird.

Keine Sorge, wir werden alle oben genannten Empfehlungen weiter unten in diesem Buch behandeln.

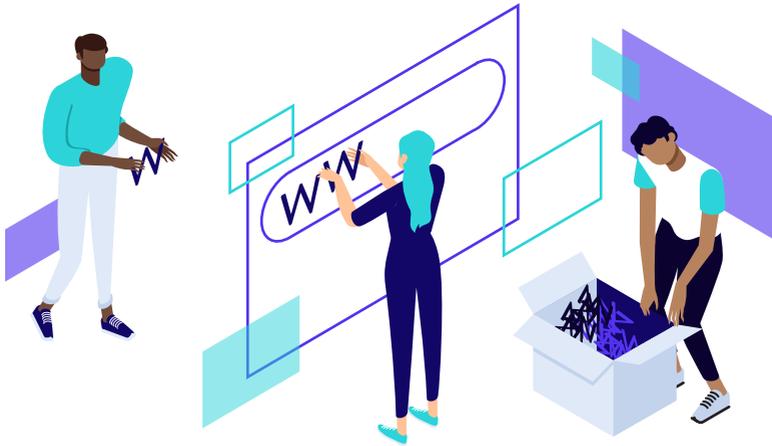
SFTP Geschwindigkeiten und das WordPress Admin Dashboard

Deine Besucher und Kunden sollten immer deine Priorität sein. Aber ein weiterer Aspekt der Leistung, über den viele nicht sprechen, ist, **wie sich einige dieser Entscheidungen auf deine tägliche Arbeit auswirken**. Der von dir gewählte Standort des Rechenzentrums hat einen Einfluss darauf, wie schnell deine SFTP-Download- und Upload-Geschwindigkeiten (Übertragung von Dateien mit einem FTP-Client) sind, sowie auf die Reaktionsfähigkeit deines WordPress-Admin-Dashboards.

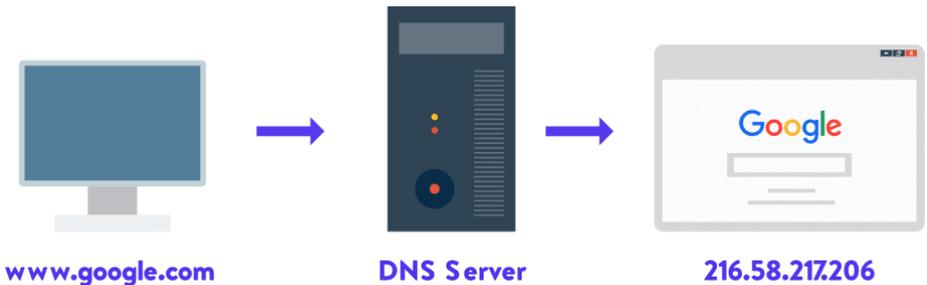
Während du also sicherstellen möchtest, dass du einen Ort wählst, der für deine Besucher am besten geeignet ist, denke auch daran, dass er die Seitenverwaltung beeinflussen kann. Aufgaben wie das Hochladen von Dateien in die WordPress-Medienbibliothek werden schneller erledigt, wenn deine Webseite in einem Rechenzentrum in deiner Nähe gehostet wird.

Wir hören immer wieder von Kunden bei Kinsta, dass sie überrascht sind, wie viel schneller ihr Admin-Dashboard bei uns ist. Es gibt eine Vielzahl von Faktoren, die das beeinflussen, aber 20 verschiedene Rechenzentren zu haben, ist ein großer! Wähle einen Ort, der sowohl für deine Besucher als auch für dich geeignet ist! Schließlich bist du derjenige, der wahrscheinlich Tausende von Stunden damit verbringen wird, an deiner Webseite zu arbeiten.

Ein premium DNS ist besser als ein kostenloses DNS



DNS, kurz für Domain Name System, ist eine der häufigsten, aber meist missverstandenen Komponenten der Web-Landschaft. Vereinfacht ausgedrückt, hilft DNS bei der “Verkehrsleitung” im Internet, indem es Domainnamen mit echten Webservern verbindet. Im Wesentlichen nimmt es eine menschenfreundliche Anfrage – einen Domainnamen wie `kinsta.com` – und übersetzt sie in eine computerfreundliche Server-IP-Adresse – wie `216.58.217.206`.



Wie DNS funktioniert

Es gibt sowohl kostenloses DNS als auch Premium-DNS. Alle Kinsta-Kunden erhalten über die Amazon Route 53 Zugang zu Premium-DNS. Und im Allgemeinen glauben wir, dass Premium-DNS in der heutigen Welt eine Notwendigkeit ist.

Ein wichtiger Grund für die Wahl von Premium-DNS sind **Geschwindigkeit und Zuverlässigkeit**. Das Nachschlagen von DNS-Einträgen und das Leiten von Datenverkehr

braucht Zeit, auch wenn es nur eine Frage von Millisekunden ist.

Normalerweise ist das kostenlose DNS, das du von deinem Domain-Namen-Registrator erhältst, vergleichsweise langsam, während Premium-DNS oft eine bessere Leistung bietet. In unseren Tests haben wir beispielsweise festgestellt, **dass das kostenlose NameCheap DNS 33% langsamer** ist als das Amazon Route 53 Premium DNS. Darüber hinaus kann Premium-DNS eine bessere Sicherheit und Verfügbarkeit bieten, insbesondere, wenn man unter einem DDoS-Angriff steht.

Du kannst ein Tool wie SolveDNS Speed Test verwenden, um deine DNS-Lookup-Zeiten zu überprüfen. DNSPerf liefert auch hervorragende Leistungsdaten über alle führenden DNS-Anbieter.

Für einen guten Mittelweg zwischen dem kostenlosen DNS deines Domain-Anbieters und dem Premium-DNS ist Cloudflare DNS, ein kostenloser Dienst, der immer noch viele Vorteile des Premium-DNS bietet. Und sie sind rasend schnell, mit durchschnittlichen Reaktionszeiten von unter 20 ms rund um den Globus (siehe unten).

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

Ein Vorbehalt bei Cloudflare ist jedoch, dass es auch mehr Ausfallzeiten hat als viele andere Anbieter. Wenn du hauptsächlich Besucher in den USA bedienst, ist DNS Made Easy ein weiterer großartiger Premium-DNS-Anbieter, den du vielleicht ausprobieren möchtest. Es hat den Ruf, einige der besten DNS-Verfügbarkeiten der letzten zehn Jahre angeboten zu haben.

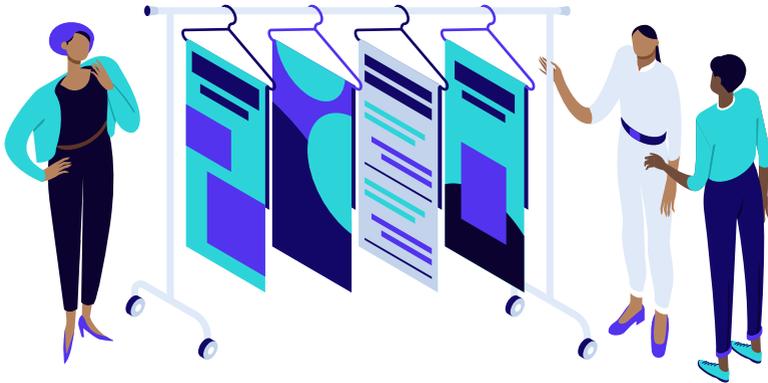
In den letzten 30 Tagen zeigt DNSPerf die nachfolgende Betriebszeit dieser Anbieter an:

- **DNS Made Easy:** 99,99%, was 4 min und 23 sek an monatlichen Ausfallzeiten entspricht
- **Amazon Route 53:** 99,88%, was 52min 35,7s monatlichen Ausfallzeiten entspricht
- **Cloudflare:** 99,85%, was 1h 5min 44,6s monatlichen Ausfallzeiten entspricht

Sind Ausfallzeiten bei DNS-Anbietern so wichtig? Die Antwort darauf ist wirklich ja und nein. DNS wird typischerweise mit ISPs gecached, die den Time To Live Value (TTL) im DNS-Eintrag verwenden. Wenn also ein DNS-Provider für 10 Minuten ausfällt, wirst du höchstwahrscheinlich nichts bemerken. Ausfallzeiten spielen jedoch eine Rolle, wenn der Provider immer wieder längere und häufige Ausfälle hat, oder wenn Deine ISP- und DNS-Einträge beide wirklich niedrige TTL-Werte verwenden.

KAPITEL 05:

Dein WordPress Theme ist entscheidend



Jeder liebt ein brandneues WordPress-Motiv, aber sei vorsichtig, bevor du rausgehst und dir das mit all den neuen, glänzenden Funktionen schnappst. Zuerst solltest du unseren Artikel über die Unterschiede zwischen kostenlosen und kostenpflichtigen Themes lesen. In Bezug auf die Leistung hat jedes Element, das du in einem Theme siehst, einen gewissen Einfluss auf die Gesamtgeschwindigkeit deiner Webseite. Und leider gibt es bei Tausenden von Themes dort draußen sowohl gute als auch schlechte.

Dein WordPress-Theme ist entscheidend für die Leistung. Wähle von Anfang an das Richtige.

Also, woher willst du wissen, welches du wählen sollst? Wir empfehlen, sich für eine der beiden folgenden Optionen zu entscheiden:

- Ein schnelles, leichtes WordPress-Theme, das **nur mit den Funktionen erstellt wurde, die du benötigst**, mehr nicht.
- Ein funktionsreicheres WordPress-Theme, aber du kannst **Funktionen deaktivieren**, die du nicht benötigst.

Dinge wie Google-Fonts, Font Awesome Symbole, Schieberegler, Galerien, Video- und Parallaxenskripte etc. das sind nur einige der vielen Dinge, die du abschalten können solltest, wenn du sie nicht verwendest. Du möchtest wirklich nicht versuchen, diese nachträglich manuell zu optimieren. Die 50 verschiedenen Möglichkeiten, diese zu entfernen, werden wir hier kaum aufgreifen können, aber stattdessen empfehlen wir, dass du mit einem WordPress-Theme startest oder zu einem anderen wechselst, das entweder von Anfang an leichtgewichtig ist oder dir diese Optionen bietet.

Unten sind ein paar WordPress-Themes, die wir empfehlen und mit denen du nichts falsch machen kannst! Vertrau uns, du wirst es uns später danken.

Jedes der unten genannten Themes ist voll kompatibel mit WooCommerce und Easy Digital Downloads, WPML, BuddyPress und bbPress. Wir führen einige Geschwindigkeitstests mit jedem Theme unter Verwendung der folgenden Konfiguration durch:

- Gehostet auf Kinsta, mit WordPress 4.9.8
- PHP 7.3 and SSL (HTTPS)
- Kinsta CDN
- Imagify wurde verwendet, um Bilder automatisch zu komprimieren

GeneratePress

GeneratePress ist ein schnelles, leichtes (weniger als 1 MB gezippt), mobiles WordPress-Theme, das mit Blick auf Geschwindigkeit, SEO und Benutzerfreundlichkeit entwickelt wurde. Erbaut von Tom Osborne, einem Entwickler aus Kanada. Es wird aktiv aktualisiert und gut unterstützt. Selbst einige Kinsta-Teammitglieder nutzen GeneratePress für ihre Projekte.

Es gibt sowohl eine kostenlose als auch eine Premium-Version. Wenn du dir das WordPress Repository ansiehst, verfügt die kostenlose Version derzeit über mehr als 100.000 aktive Installationen, mehr als 1,6 Millionen Downloads und eine beeindruckende 5 von 5 Sternen Bewertung (über 700 Personen haben ihm 5 Sterne gegeben).

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

Install

Premium

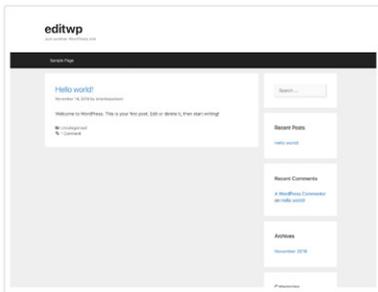
1,643,956+
Downloads

700+
★★★★★

100,000+
Active websites

40,000+
Happy customers

Wie schnell ist es? Wir haben GeneratePress neu installiert, fünf Geschwindigkeitstests in Pingdom durchgeführt und den Durchschnitt ermittelt. Die **Gesamtladezeit betrug 305 ms** bei einer **Gesamtseitengröße von nur 16,8 KB**. Es ist immer gut, einen Basistest durchzuführen, um zu sehen, wozu das Motiv in Bezug auf die Rohleistung fähig ist.



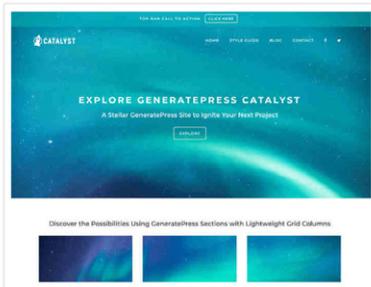
Performance grade
A 97

Page size
16.8 KB

Load time
305 ms

Requests
7

Anschließend führten wir eine weitere Reihe von Tests mit einem der vorgefertigten Designs aus der GeneratePress Bibliothek durch. Das beinhaltet Bilder, Hintergründe, neue Abschnitte etc. Ein Vorteil von GeneratePress ist, dass es viele vorgefertigte Designs hat, die kein Page Builder Plugin erfordern. Man sieht, dass es noch unter **400 ms** getaktet hat.



Performance grade
B 88

Page size
837.5 KB

Load time
396 ms

Requests
36

In einer realen Umgebung, kannst du natürlich andere Dinge wie Google Analytics, Facebook Remarketing Pixel, Hotjar etc. ausführen lassen. Aber du solltest in der Lage sein, leicht unter der 1-Sekunden-Marke zu bleiben.

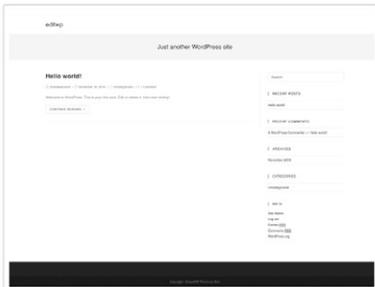
Wir zeigen dir weitere Möglichkeiten, wie du WordPress optimieren und beschleunigen kannst.

OceanWP

Das OceanWP-Theme ist leicht und gut erweiterbar und ermöglicht es dir, fast jede Art von Webseite wie einen Blog, ein Portfolio, eine Business-Webseite und ein WooCommerce Storefront mit einem schönen und professionellen Design zu erstellen. Es wurde von Nicolas Lecocq entworfen und wird aktiv aktualisiert und gut unterstützt.

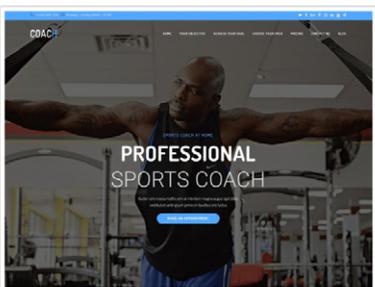
Genau wie bei GeneratePress gibt es sowohl eine kostenlose als auch eine Premium-Version. Wenn du dir den WordPress Repository ansiehst, hat die kostenlose Version derzeit über 200.000 aktive Installationen und weitere beeindruckende 5 von 5 Sterne Bewertungen (über 1.200 Nutzer haben 5 Sterne vergeben).

Wie schnell ist es? Wir haben OceanWP neu installiert, fünf Geschwindigkeitstests in Pingdom durchgeführt und den Durchschnitt ermittelt. Die Gesamtladezeit betrug **389 ms bei einer Gesamtseitengröße von nur 230,8 KB.**



Performance grade A 92	Page size 230.5 KB
Load time 389 ms	Requests 16

Anschließend führten wir eine weitere Reihe von Tests mit einem der Demo-Themes aus der OceanWP-Bibliothek durch. Diese enthält Bilder, Hintergründe, neue Abschnitte und erfordert das Elementor Page Builder Plugin. Man sieht, dass es noch unter **600 ms** läuft.



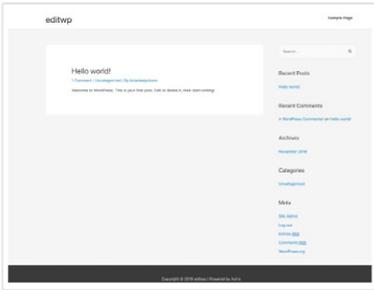
Performance grade B 84	Page size 941.1 KB
Load time 548 ms	Requests 48

Astra

Astra ist ein schnelles, vollständig anpassbares und schönes Design, das sich für Blogs, persönliche Portfolios, Geschäftswebseiten und WooCommerce-Schaufenster eignet. Es ist sehr leicht (weniger als 50 KB am Frontend) und bietet eine beispiellose Geschwindigkeit. Es wurde vom Team von Brainstorm Force entwickelt und wird aktiv aktualisiert und gut unterstützt. Du erkennst sie möglicherweise als die Entwickler des beliebten All In One Schema Rich Snippets Plugins wieder, das es seit vielen Jahren gibt.

Genau wie bei GeneratePress und OceanWP gibt es sowohl eine kostenlose als auch eine Premium-Version. Wenn du dir den WordPress Repository ansiehst, hat die kostenlose Version derzeit über 100.000 aktive Installationen, 1,6+ Millionen Downloads und darüber hinaus eine beeindruckende 5 von 5 Sterne Bewertung (über 850 Nutzer haben 5 Sterne vergeben).

Wie schnell ist es? Wir haben eine Neuinstallation von Astra durchgeführt, fünf Geschwindigkeitstests in Pingdom durchgeführt und den Durchschnitt ermittelt. Die Gesamtladezeit betrug **243 ms** bei einer **Gesamtseitengröße von nur 26,6 KB**.



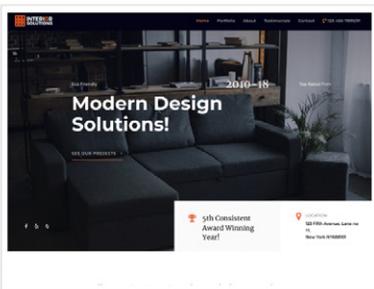
Performance grade
A 98

Page size
26.6 KB

Load time
243 ms

Requests
5

Wir führten dann einen weiteren Satz von Tests mit einem der Demo-Designs aus der Astra Starter Kit Bibliothek durch. Diese enthält Bilder, Hintergründe, neue Abschnitte und erfordert das Elementor Page Builder Plugin. Man sieht, dass es noch unter **700 ms** gelaufen ist. Hinweis: Die Bilder in dieser Demo wurden vollständig komprimiert, aber sie wählten von Anfang an sehr hochauflösende Bilder.



Performance grade
B 86

Page size
1.6 MB

Load time
652 ms

Requests
54

Es ist wichtig, die Unterschiede zwischen den Geschwindigkeitstests mit diesen drei Themes mit Vorsicht zu genießen. Das Problem ist, dass es fast unmöglich ist, einen völlig genauen Seitenvergleich durchzuführen. Das Wichtigste, was wir dir zeigen wollten, ist, dass all diese WordPress-Themes rasant schnell arbeiten, sowohl direkt „von der Stange“ als auch mit vollständigen Demos!

Eine Warnung wenn es um PageBuilder geht.

Wie du wahrscheinlich bemerkt haben wirst, benötigten OceanWP und Astra beide Seitenersteller, um ihre jeweiligen Bibliotheksthemes zu verwenden. Hier sind ein paar Dinge, die du bei der Verwendung eines Page Builder Plugins beachten solltest:

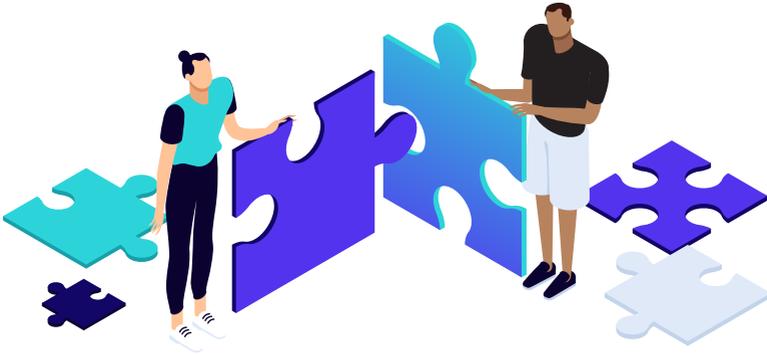
- Einige Seitenersteller können die Ladezeit auf deiner Webseite verlängern. Das liegt daran, dass sie zusätzliche CSS und JS laden müssen, damit die Prozesse für dich ohne Code funktionieren. So funktioniert die Magie! Wir empfehlen immer, die Geschwindigkeit der WordPress-Seite vor und nach der Installation eines Page Builders zu testen.
- Du bindest dich mit diesem Seitenersteller an ein einziges Design. Achte darauf, dass du dich für einen entscheidest, der regelmäßig aktualisiert wird und alles hat, was du langfristig benötigst.

Allerdings sind wir immer noch große Fans von Pagebuildern wie Elementor und Beaver Builder. In den meisten Fällen werden sie unter dem Gesichtspunkt der Leistung entwickelt und fügen nur wenig Aufwand hinzu. Für die meisten lohnt sich die Funktionalität und Benutzerfreundlichkeit, denn mit diesen Plugins kannst du alles erstellen, was du dir vorstellst! Sie könnten auch in einigen Fällen schneller sein, da sie ein Ersatz für mehr als 5 andere Plugins sein könnten, die du sonst hättest verwenden müssen.

Wenn du jedoch kein Page Builder Plugin benötigst, installiere auf jeden Fall nicht einen nur für den Kick. Es wird auch interessant sein zu sehen, wie der neue Gutenberg-Editor in den nächsten Jahren eine Rolle im Webseite-Design spielen wird.

KAPITEL 06:

Ein Überblick über WordPress Plugins



Nun zu den Informationen über WordPress-Plugins. Es wurde dir vielleicht gesagt, dass zu viele Plugins nicht das Gelbe vom Ei sind, da das deine WordPress-Seite verlangsamen würde. Obwohl das manchmal wahr ist, ist es nicht der kritischste Faktor. Die Anzahl der Plugins ist nicht so wichtig wie die Qualität derselben. So, jetzt haben wir es gesagt.

Genau wie bei Themes spielt es eine Rolle, wie das Plugin entwickelt wird und ob es mit Blick auf die Performance entwickelt wurde. Wir haben viele Kunden bei Kinsta, die 30-40 Plugins betreiben und deren Seiten immer noch weit unter einer Sekunde laden.

Es macht zwar Spaß, deiner Webseite Code hinzuzufügen, aber das ist aus den folgenden Gründen nicht immer praktisch:

- 1.** Man muss den Code selbst pflegen und ihn bei Änderungen der Standards auf dem neuesten Stand halten. Warum dann nicht auf die fantastischen Entwickler vertrauen, die die Standards besser kennen als die meisten anderen?
- 2.** Meistens wird ein gut programmiertes Plugin nicht viel mehr Overhead einbringen als der Code selbst.
- 3.** Beachte, dass die Mehrheit der WordPress-Community nicht so technisch versiert ist wie die Entwickler. Plugins sind Lösungen, die helfen, Probleme zu lösen.
- 4.** Es gibt natürlich auch nicht so tolle Plugins da draußen, von denen man sich

fernhalten sollte. Vertraue uns, wir haben bei Kinsta das Schlimmste vom Schlimmsten gesehen. Viele, nicht alle, der Plugins, die wir bei Kinsta verbannen, verursachen Performance-Probleme aus erster Hand. Wir werden vertiefend weiter unten darüber reden.

Es steht jedoch fest, dass eines der Dinge, die Menschen an WordPress lieben, seine riesige Bibliothek von Drittanbieter-Plugins ist. Aber mit mehr als 56.000 kostenlosen Plugins, die allein auf WordPress.org aufgelistet sind, und Tausenden weiteren, die anderswo aufgelistet sind, kann es schwierig sein, das eine Plugin zu finden, das man benötigt. Apropos Nadel im Heuhaufen! Schau dir die Liste an, die wir nur aus den besten WordPress-Plugins auf dem Markt zusammengestellt haben.

Wir versuchen nur Dinge, die wir täglich benutzen, zu teilen. Und ja, wir verwenden WordPress-Plugins auf unserer Webseite, genau wie Ihr auch. Viele der Teammitglieder bei Kinsta entwickeln und verkaufen sogar Plugins.

Ein großes Problem bei WordPress Plugins

Ein großes Problem bei WordPress-Plugins ist der Deinstallationsprozess. Wann immer du ein WordPress-Plugin oder -Theme installierst, speichert es die Daten in der Datenbank. Das Problem ist, dass, wenn du ein Plugin mit einer der Standardmethoden löschst, es normalerweise Tabellen und Zeilen in deiner Datenbank hinterlässt. Im Laufe der Zeit kann das zu einer Menge von Daten führen und sogar anfangen, deine Webseite zu verlangsamen. In unserem Beispiel haben wir das Wordfence Security Plugin deinstalliert und es hinterließ 24 Tabellen in unserer Datenbank (siehe unten). Es ist noch schlimmer, wenn sie hinter den Daten in der Tabelle `wp_options` liegen.

<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	KiB
<input type="checkbox"/>	wp_wfNet404s	★	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8_general_ci	32	KiB
<input type="checkbox"/>	wp_wfNotifications	★	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8_general_ci	16	KiB
<input type="checkbox"/>	wp_wfPendingIssues	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	16	KiB
<input type="checkbox"/>	wp_wfReverseCache	★	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	latin1_swedish_ci	16	KiB
<input type="checkbox"/>	wp_wfScanners	★	Browse	Structure	Search	Insert	Empty	Drop	618	InnoDB	latin1_swedish_ci	64	KiB
<input type="checkbox"/>	wp_wfSNIPCache	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	64	KiB
<input type="checkbox"/>	wp_wfStatus	★	Browse	Structure	Search	Insert	Empty	Drop	40	InnoDB	utf8_general_ci	48	KiB
<input type="checkbox"/>	wp_wfThrottleLog	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8_general_ci	32	KiB
<input type="checkbox"/>	wp_wfVulnScanners	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	latin1_swedish_ci	16	KiB

WordFence Tabellen

Und neben der Datenbank hinterlassen viele Plugins auch zusätzliche Ordner und Dateien. Nach unserer Erfahrung wird das häufig bei Sicherheits- und Caching-Plugins beobachtet, die zusätzliche Verzeichnisse für die Protokollierung erstellen. Nachdem beispielsweise das Wordfence-Plugin gelöscht wurde, wurde uns ein Ordner „wflogs“ in unserem Verzeichnis wp-content zur Verfügung gestellt. Aber wir versuchen nicht, uns auf Wordfence zu konzentrieren da die meisten Plugins und Themes auf dem Markt auch so funktionieren.

Remote site: /www/ /public/wp-content/wflogs

- upgrade
- uploads
- wflogs

Filename	File...	Filetype	Last modified
..			
.htaccess	133	HTACCESS File	7/24/2017 10:54:49 AM
attack-data.php	40,0...	PHP File	7/24/2017 10:54:49 AM
config.php	594,...	PHP File	7/24/2017 10:57:30 AM
ips.php	51	PHP File	7/24/2017 10:54:50 AM
rules.php	101,...	PHP File	7/24/2017 10:54:59 AM
wafRules.rules	44,2...	RULES File	7/24/2017 10:54:59 AM

Logs WordFence

Warum tun Entwickler das?

Du fragst dich also wahrscheinlich, warum haben Entwickler keine Selbstreinigungsoptionen, wenn sie ein Plugin deinstallieren und löschen? Nun, das tun sie. Aber, hier sind ein paar Gründe, warum sie wahrscheinlich nicht so offensichtlich sind.

- 1. Sie wollen die Einstellungen für den Benutzer beibehalten.** Wenn du ein WordPress-Plugin löschst und dich entscheidest, es später noch einmal zu versuchen, sind alle deine Einstellungen und Daten noch vorhanden. Dies ist zwar sehr praktisch, aber nicht der effizienteste Weg.
- 2. Sie kümmern sich nicht um die Leistung.** Einige Entwickler mögen argumentieren, dass das Hinterlassen von Tabellen keinen Einfluss auf die Performance hat. Aber stelle dir eine Webseite im Laufe von zehn Jahren vor, die Hunderte von Plugins verwendet hat, die möglicherweise Tausende von Zeilen oder Tabellen generiert haben. Datenbankabfragen haben einen erheblichen Einfluss auf die Leistung deiner WordPress-Seite, und Plugins können viele dieser Anfragen stellen, wenn der Entwickler nicht vorsichtig war. Im Allgemeinen sollte ein gut geschriebenes Plugin nur die Tabellen oder Zeilen abfragen, an die es gebunden ist, das ist jedoch nicht immer der Fall. Wir haben das bei Kinsta aus erster Hand gesehen, lange Datenbankabfragen, die eine Webseite zum Crawlen bringen, weil unnötige, automatisch geladene Daten in der Tabelle `wp_options` zurückgelassen wurden.
- 3. Sie haben einen Fehler gemacht.** Im WordPress Plugin-Handbuch steht sogar, dass „weniger erfahrene Entwickler manchmal den Fehler machen, den Deaktivierungshaken zu diesem Zweck zu verwenden“.

Die guten Nachrichten? Es gibt Möglichkeiten, ein Plugin zu bereinigen und richtig zu entfernen. Links zu den folgenden ausführlichen Tutorials findest du auf unserer Webseite unter: <https://kinsta.com/de/lernen/wordpress-beschleunigen/>.

- So deinstallierst du ein WordPress-Plugin (auf die richtige Weise)
- Manuelle Bereinigung von Tabellen, die hinterlassen wurden

KAPITEL 07:

Die optimalen WordPress Einstellungen



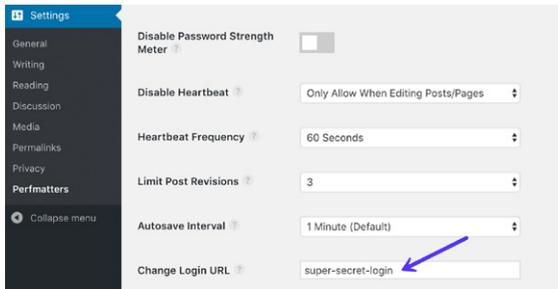
Nun zu den optimalen WordPress-Einstellungen. Hier sind ein paar Änderungen, die du vornehmen kannst, um deine WordPress-Seite zu beschleunigen. Viele davon sind sehr subtile Veränderungen, aber alles hilft!

Ändere deine WordPress Login URL

Standardmäßig ist die Anmelde-URL deiner WordPress-Seite `domain.com/wp-admin/`. Eines der Probleme dabei ist, dass alle Bots, Hacker und Skripte da draußen das auch wissen. Durch das Ändern der URL kannst du dich weniger zu einem potentiellen Angriffsziel machen, dich besser vor Brute-Force-Angriffen schützen und die Bandbreite der Bots verringern, die diese URL wiederholt treffen.

Das Ändern der Anmelde-URL von WordPress kann auch dazu beitragen, häufige Fehler wie „429 Too Many Requests“ zu vermeiden. Das ist keine Lösung, es ist nur ein kleiner Trick, der helfen kann, dich zu schützen und die Belastung auf der Seite zu verringern.

Um deine WordPress Login-URL zu ändern, empfehlen wir dir, eines der folgenden Plugins zu verwenden:



Ändere die WordPress Login URL mit Perfmatters

- **WPS Hide Login** (kostenlos)
- **Perfmatters** (Premium, beinhaltet aber auch andere Einstellungen zur Leistungsoptimierung; entwickelt von einem Teammitglied bei Kinsta)

Plugin und Theme Updates deaktivieren oder optimieren

Langsame WordPress Admin-Dashboards können vom Netzwerk, dem Standort des Rechenzentrums und sogar von PHP-Versionen beeinflusst werden. Aber ein weiterer Faktor, über den nicht viele Leute sprechen, ist der WordPress Update Checker, der im Hintergrund läuft. Das ist ein Fall, in dem viele WordPress Plugins und Themes dir schaden könnten. WeFoster hat einen großartigen Blog-Post zu diesem Thema, in dem sie den Begriff „Third Party Plugin Update Check Syndrome“ oder TPPUCS einführen.

Im Wesentlichen besteht das Problem darin, dass der eingebaute WordPress Update Checker hinter den Kulissen eine externe GET-Anfrage stellt (<https://third-party-plugin/update-check.php>). Manchmal kann das periodisch oder sehr häufig erfolgen. Wenn es häufiger vorkommt, könnte das dazu führen, dass dein Admin-Dashboard stark verlangsamt.

Das Problem hängt eher damit zusammen, wie der Update-Checker in WordPress aufgebaut ist. Wenn du unter langsamen Ladezeiten des WordPress-Administrations-Dashboards leidest, solltest du es vielleicht versuchen. Die Lösung besteht darin, automatische Updates zu deaktivieren. Warnung: Das ist nur dann der Fall, wenn du beabsichtigst, manuell nach Updates zu suchen. Viele Updates beinhalten Sicherheits- und Bugfixes.

Um Updates zu deaktivieren, empfehlen wir die Verwendung eines der folgenden Plugins:

- **Deactivate all WordPress Updates:** völlig kostenlos und ohne Einstellungen; macht das, was es sagt, gut.
- **Simple Update-Manager:** bietet mehr Kontrolle über selektive Updates. Die Basisversion ist kostenlos.

Du könntest dir ganz einfach eine Kalendererinnerung einrichten, um das Plugin einmal pro Woche zu deaktivieren, nach Updates zu suchen und es dann wieder zu aktivieren.

Pingbacks deaktivieren

Ein Pingback ist ein automatisierter Kommentar, der erstellt wird, wenn ein anderer Blog auf dich verlinkt. Es kann auch Self-Pingbacks geben, die erstellt werden, wenn du auf einen Artikel in deinem eigenen Blog verlinkst.

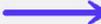
Wir empfehlen, diese einfach zu deaktivieren, da sie wertlose Abfragen und zusätzlichen Spam auf deiner Webseite erzeugen. Denke daran, je weniger Aufrufe deine WordPress-Seite machen muss, desto besser, besonders auf stark frequentierten Seiten. Ganz zu schweigen von der Tatsache, dass ein Pingback auf der eigenen Webseite einfach geradezu ärgerlich ist. Führe die folgenden Schritte aus, um Pingbacks zu deaktivieren.

Schritt 1 – Pingbacks von anderen Blogs deaktivieren

Klicke in deinem WordPress Dashboard auf „Einstellungen / Diskussion“. Deaktiviere im Bereich Diskussionseinstellungen die Option „Link-Benachrichtigungen aus anderen Blogs (Pingbacks und Trackbacks) über neue Artikel zulassen“.

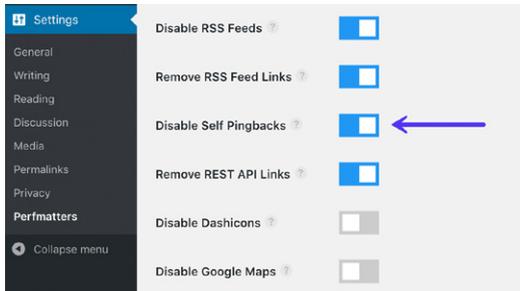
Discussion Settings

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.)

Schritt 2 – Deaktiviere Self-Pingbacks



Wenn es darum geht, Self-Pingbacks zu deaktivieren, hast du verschiedene Möglichkeiten. Du kannst das kostenlose No Self Pings Plugin verwenden. Oder du kannst ein Premium-Plugin wie Perfmatters verwenden.

Deaktiviere Self-Pingbacks mit Perfmatters

Alternativ kannst Du auch Self-Pingbacks deaktivieren, indem du den folgenden Code zur Datei `functions.php` deines WordPress-Themes hinzufügst. Warnung: das Bearbeiten der Quelle eines WordPress-Themes könnte deine Seite beschädigen, wenn sie nicht korrekt ausgeführt wird. Tipp: Du kannst mit dem kostenlosen Code Snippets Plugin einfach solche PHP-Snippets hinzufügen. Das bedeutet, dass du dein Theme nicht bearbeiten musst.

```
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' );
```

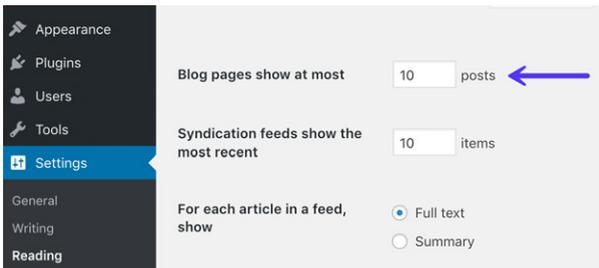
Limitiere die Anzahl der Posts in deinem Blog Feed

Egal, ob dein Blog-Feed als deine Homepage eingestellt ist oder eine andere Seite deiner Webseite, du brauchst keine 50 Miniaturansichten, die alle gleichzeitig geladen werden. Für diejenigen, die hochfrequentierte Blogs betreiben, ist die Homepage die wichtigste Seite einer Webseite, und du willst, dass sie schnell geladen wird. Je weniger Anfragen und Medien, desto besser in Bezug auf die Leistung!

Auch genau deshalb wurde der Seitennummerierung erfunden (wie unten gezeigt). Es ist das, was du am Ende von Blog-Feeds siehst, mit denen du zur nächsten Seite navigieren kannst. Typischerweise sind das Zahlen, oder sie verwenden möglicherweise „nächste/vorherige“ Beiträge. Dein WordPress-Thema wird höchstwahrscheinlich bereits eine benutzerdefinierte Seitennummerierung eingebaut haben.



WordPress setzt standardmäßig die Grenze für neue WordPress-Installationen auf 10, aber wir haben gesehen, dass sich das so oft geändert hat, dass wir aufgehört haben mitzuzählen. Achte also darauf, dass du noch einmal überprüfst, welchen Wert du verwendest. Wir empfehlen einen Wert zwischen 8 und 12. Falls du neugierig bist, wir verwenden 12 auf unserer Kinsta-Blog-Homepage.



Du findest diese Option in deinem WordPress Admin Dashboard unter „Einstellungen / Lesen“. Anschließend kannst du den Wert für „Blogseiten maximal anzeigen“ ändern.

KAPITEL 08:

Warum ist Caching so wichtig



Caching ist bei weitem eine der wichtigsten und einfachsten Möglichkeiten, WordPress zu beschleunigen! Aber bevor wir dir zeigen, wie man Caching verwendet, ist es wichtig, zuerst zu verstehen, wie es funktioniert und welche verschiedenen Arten von Caching es gibt.

Was ist Caching?

Kurz gesagt, jede Webseite, die auf deiner WordPress-Seite besucht wird, erfordert eine Anfrage an den Server, die Verarbeitung durch diesen Server (einschließlich Datenbankabfragen) und dann ein Endergebnis, das vom Server an den Browser des Benutzers gesendet wird. Das Ergebnis ist deine Webseite, komplett mit allen Dateien und Elementen, die sie so aussehen lassen, wie sie eben aussieht.

Zum Beispiel kannst du einen Header, Bilder, ein Menü und einen Blog haben. Da der Server alle diese Anfragen bearbeiten muss, dauert es einige Zeit, bis die gesamte Webseite dem Benutzer zugestellt wird – insbesondere bei klobigen oder größeren Webseiten.

Hier kommt ein WordPress Caching-Plugin ins Spiel! Caching weist den Server an, je nach Konfiguration einige Dateien auf der Festplatte oder im RAM zu speichern. Daher kann

es sich an den gleichen Inhalt erinnern und ihn duplizieren, den es in der Vergangenheit bereitgestellt hat. Im Grunde genommen reduziert es den Arbeitsaufwand für die Generierung eines Seitenaufrufs. **Dadurch werden deine Webseiten viel schneller geladen, direkt aus dem Cache.**

Einige andere Vorteile des Caching sind:

- **Dein Server verbraucht weniger Ressourcen** – das bringt folglich Geschwindigkeit mit sich, da weniger Ressourcen für eine schnellere Webseite sorgen. Es belastet aber auch den Server weniger. Das ist sehr wichtig, wenn es um hochdynamische Webseiten geht, wie z.B. Mitgliederseiten und um festzustellen, was man aus dem Cache bedienen kann und was nicht.
- **Du wirst sehen, dass die TTFB niedriger ist** – Caching ist eine der einfachsten Möglichkeiten, deine TTFB zu senken. Tatsächlich reduziert das Caching in unseren Tests typischerweise TTFB um bis zu 90%!

Cachingarten

Wenn es um die Art des Caching geht, gibt es zwei verschiedene Ansätze, die häufig verwendet werden:

1. Caching auf Serverebene
2. Caching mit einem Plugin

1. Caching auf Serverebene

Das Caching auf Serverebene ist bei weitem einer der einfachsten Ansätze für den Endanwender. Das bedeutet, dass der WordPress Hosting-Provider das für dich übernimmt. Bei Kinsta verwenden wir die folgenden **vier Arten von Cache**, die alle automatisch auf Software- oder Serverebene ausgeführt werden:

- Bytecode caching

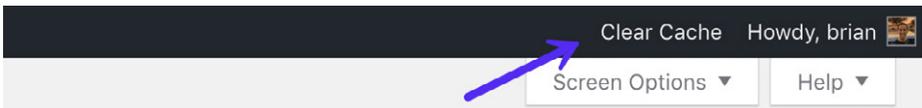
- Object caching
- Page caching
- CDN caching

Das bedeutet, dass du dir keine Sorgen machen musst, dich mit komplizierten und verwirrenden Caching-Plugins zu beschäftigen. Du kannst aufhören, nach den „besten Caching-Plugins von 2019“ zu suchen und dich auf produktivere Aufgaben konzentrieren.

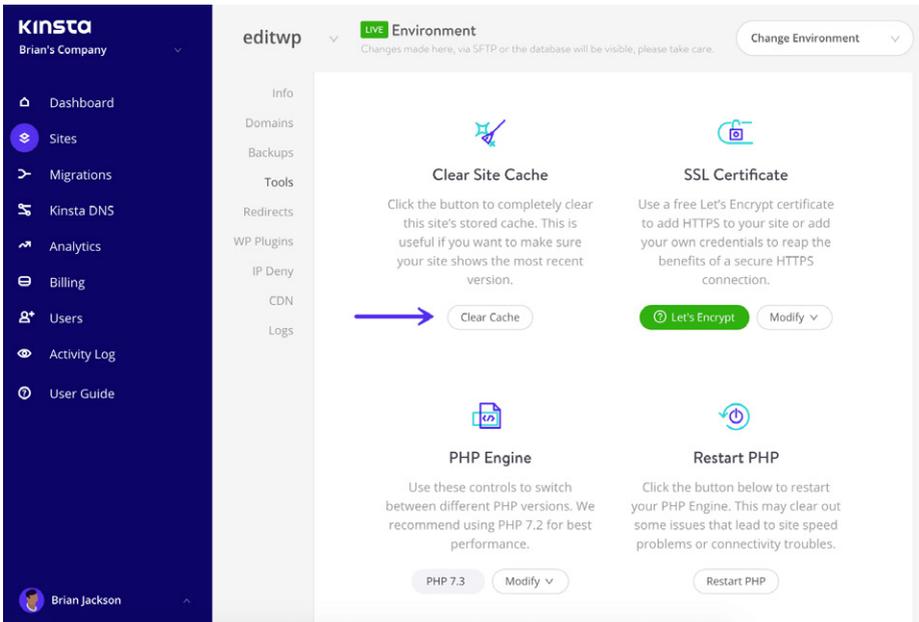
Der Seiten-Cache ist so konfiguriert, dass er mit Standard WordPress sofort nach dem Einrichten funktioniert. Du musst gar nichts tun! Starte einfach deine WordPress-Seite und der Seiten-Cache beginnt.

Wir haben auch Caching-Regeln für E-Commerce-Seiten wie WooCommerce und Easy Digital Downloads. Standardmäßig werden bestimmte Seiten, die niemals zwischengespeichert werden sollten, wie z.B. Warenkorb, mein Konto und Auschecken, vom Caching ausgeschlossen. Benutzer umgehen automatisch den Cache, wenn das `woocommerce_items_in_cart` Cookie oder `edd_items_in_cart` erkannt wird, um einen reibungslosen und synchronen Checkout-Prozess zu gewährleisten.

Du kannst den Cache deiner WordPress-Seite jederzeit über die Admin-Symboleiste löschen.



Es ist auch in unserem MyKinsta Dashboard integriert. Klick einfach bei Werkzeuge auf „Cache löschen“.



2. Caching mit einem Plugin

Wenn dein Hosting-Provider keinen Cache zur Verfügung stellt, kannst Du ein WordPress-Caching-Plugin von Drittanbietern verwenden. Aufgrund unserer Erfahrung empfehlen wir eine der folgenden Optionen:

- **WP Rocket** (Premium, erhältlich bei wp-rocket.me)
- **Cache Enabler** (kostenlos, erhältlich bei WordPress.org)
- **W3 Total Cache** (kostenlos, erhältlich bei WordPress.org)

Du kannst auch einige weitere Optionen in unserem ausführlichen Beitrag über WordPress Caching-Plugins nachlesen.

Wir unterstützen auch WP Rocket in Kinsta! Wir erlauben in der Regel keine Caching-Plugins in unserer Umgebung, da sie im Konflikt mit unserer integrierten Caching-Lösung

stehen. Ab WP Rocket 3.0 wird die Page-Caching-Funktionalität jedoch automatisch deaktiviert, wenn sie auf Kinsta-Servern ausgeführt wird.

Dies ermöglicht es Kinsta-Kunden, unser schnelles Caching auf Serverebene zu verwenden, aber dennoch die fantastischen Optimierungsfunktionen von WP Rocket zu nutzen.

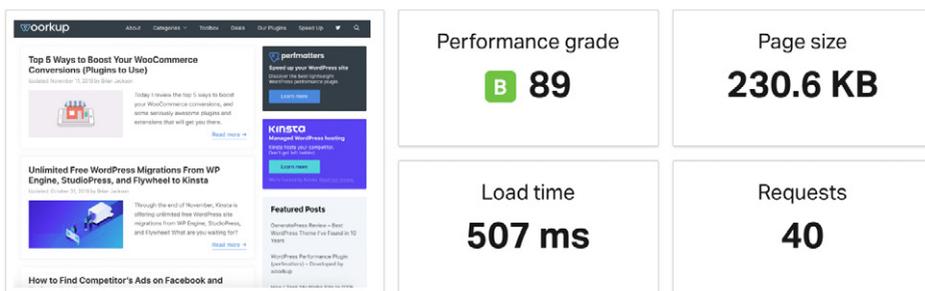
Kein Caching vs. Caching

Wie viel hilft das Caching? Der Beweis liegt im Probieren.

Wir haben einige Geschwindigkeitstests mit dem Caching auf Serverebene von Kinsta durchgeführt, damit du sehen kannst, welchen Unterschied es macht, sowohl in Bezug auf die Gesamtgeschwindigkeit als auch auf TTFB.

Kein Caching

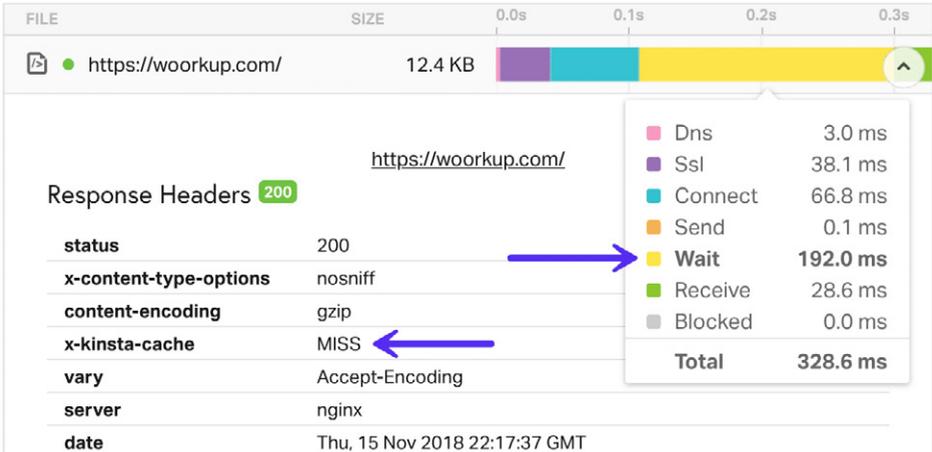
Wir haben zuerst fünf Tests auf Pingdom ohne aktiviertes Caching durchgeführt und den Durchschnitt genommen.



Kein Caching TTFB

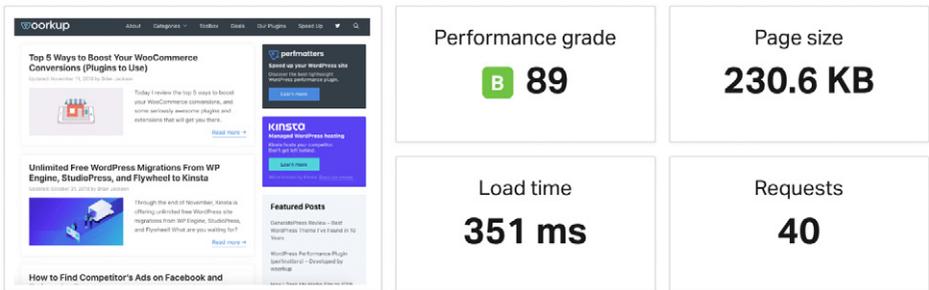
Es ist auch wichtig, den Unterschied in TTFB mit und ohne Caching zu beachten. TTFB

in Pingdom wird durch die gelbe „wartende“ Leiste dargestellt. Wie man sieht, beträgt der TTFB ohne Caching 192 ms. Wie du sehen kannst, wird die Seite nicht aus dem Cache geladen, da der `x-kinsta-cache` Header ein `MISS` anzeigt.



Mit aktiviertem Caching

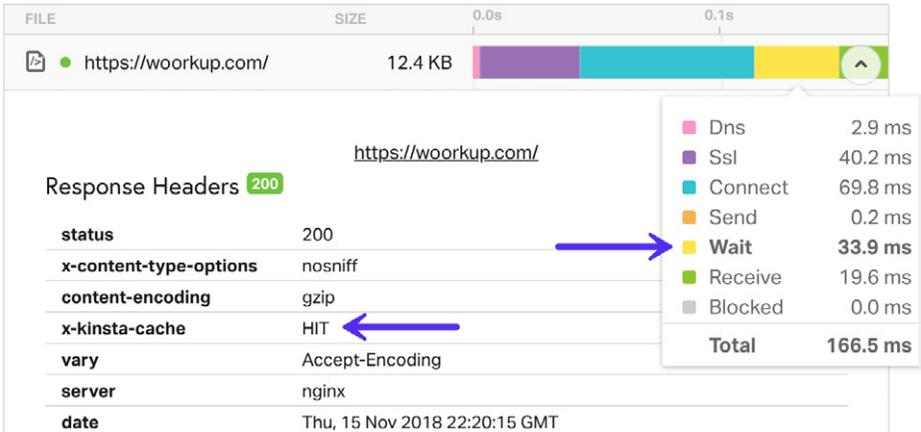
Dann haben wir fünf Tests mit Caching auf Server-Ebene aktiviert auf Pingdom durchgeführt und den Durchschnitt genommen.



Das Caching auf Server-Ebene hat unsere Seitenladezeit um 33,77% verkürzt! Und das ohne zusätzlichen Arbeitsaufwand. Diese von uns getestete Webseite ist schon ziemlich optimiert, so dass größere weniger optimierte Webseiten noch größere Unterschiede aufweisen werden.

TTFB mit aktiviertem Caching

Wenn wir uns nun die TTFB mit aktiviertem Caching ansehen, können wir sehen, dass sie unter 35 ms liegt. Man erkennt, dass es aus dem Cache bedient wird, da der `x-kinsta-cache` Header einen `HIT` anzeigt.



Der CDN-Cache ist ebenso wichtig wie der Cache von deinem WordPress-Host. Wir werden weiter unten mehr über CDNs erfahren.

*WordPress Caching kann deine
Seitenladezeiten um über 33% verkürzen!*

Probleme mit dem Caching und Mitgliederseiten

Mitgliederseiten enthalten viele **nicht cachebare Inhalte** und Seiten, die sich ständig ändern. Dinge wie die Login-Seite für Community-Mitglieder (die je nach Größe der Webseite ständig angegriffen werden können), Checkout-Seiten für digitale Waren oder Kurse und Diskussionsforen sind häufige Ursachen und Schwachstellen, da diese in der Regel nicht zwischengespeichert werden können.

Doch damit ist es noch nicht zu Ende. Auf Standard WordPress Seiten wird das WordPress Dashboard auch nicht für „**angemeldete**“ **Benutzer** zwischengespeichert. Das ist in Ordnung, wenn du nur ein paar Autoren und Administratoren hast, aber wenn plötzlich Tausende von Mitglieder das Dashboard benutzen, verursacht das sofort Performance-Probleme, da nichts davon aus dem Cache auf dem Server laufen kann. Das bedeutet, dass du die Kraft und Architektur hinter den Kulissen brauchst, um sie zu unterstützen. Shared Hosting Provider werden unter diesen Umständen in der Regel lahmgelegt.

Objekt Caching für hoch dynamische Webseiten

Wenn es um WordPress-Mitgliederseiten geht, sind deine üblichen Caching-Setups in der Regel nicht ausreichend, da sie nicht immer alle Vorteile nutzen. **Hier kommt das Objekt-Caching ins Spiel.**

Der Objekt-Cache speichert die Ergebnisse von Datenbankabfragen, so dass beim nächsten Mal, wenn ein bestimmtes Bit an Daten benötigt wird, diese aus dem Cache geliefert werden können, ohne die Datenbank abzufragen. Dies beschleunigt die PHP-Ausführungszeiten und entlastet deine Datenbank. Das wird bei Mitgliederseiten extrem wichtig! Mit WordPress kannst du das Objekt-Caching auf verschiedene Arten implementieren:

1. Eine Caching-Lösung eines Drittanbieters wie beispielsweise W3 Total Cache
2. **Redis (empfohlen)**
3. Memcached

Wir bieten Redis als Add-on bei Kinsta an, damit du den vollen Nutzen aus dem persistenten Objekt-Caching für deine Mitgliedsseiten ziehen kannst.

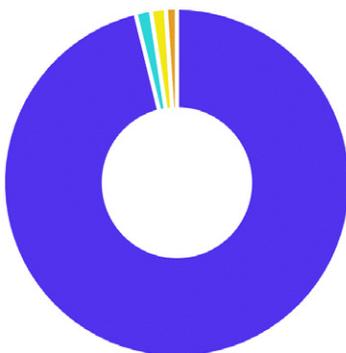
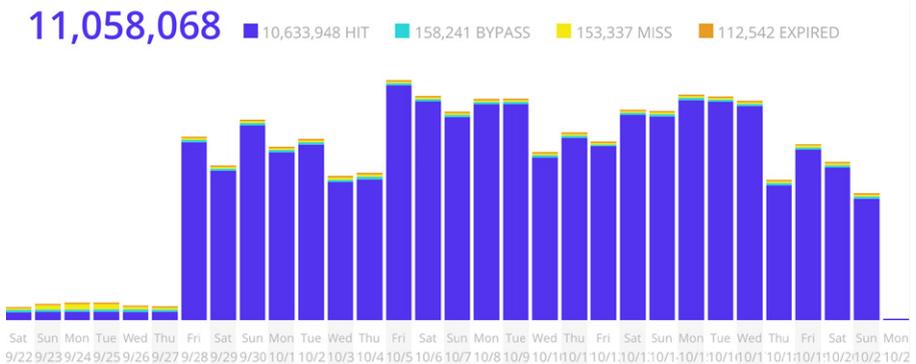
Cache analysieren

Erinnerst du dich an den `x-kinsta-cache` Header, den wir oben erwähnt haben?

Abhängig von deinem Hosting-Provider oder deiner Caching-Lösung kann der Header etwas anders heißen. Jedes Mal, wenn eine Anfrage von deiner WordPress-Seite gestellt wird, hat dieser Header einen Wert, wie beispielsweise HIT, BYPASS, MISS und EXPIRED. Auf diese Weise kannst du sehen, wie der Cache funktioniert.

Die Erhöhung der Cache-Trefferrate deiner WordPress-Seite ist wichtig, da du möchtest, dass so viel von deiner Seite aus dem Cache bedient wird wie möglich. Bei Kinsta kannst du die Daten in unserem MyKinsta Analysetool und den Kinsta-Cache-Protokollen analysieren, um festzustellen, ob es Cache BYPASSing GET-Anfragen gibt, die zwischengespeichert oder POST-Anforderungen, die eliminiert werden könnten.

Der Cache component stack (wie unten gezeigt) zeigt den Status jeder Anforderung an, unabhängig davon, ob es sich um einen HIT, BYPASS, MISS oder EXPIRED handelt. Du kannst die Daten nach den letzten 24 Stunden, 7 Tagen oder 30 Tagen filtern.



Total
11,058,068

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

Das Cache Component

Chart gibt dir einen Überblick über dein Caching-Verhältnis. Je mehr Anfragen du aus dem Cache stellst, desto besser. Wie du im folgenden Beispiel sehen kannst, liegt diese WordPress-

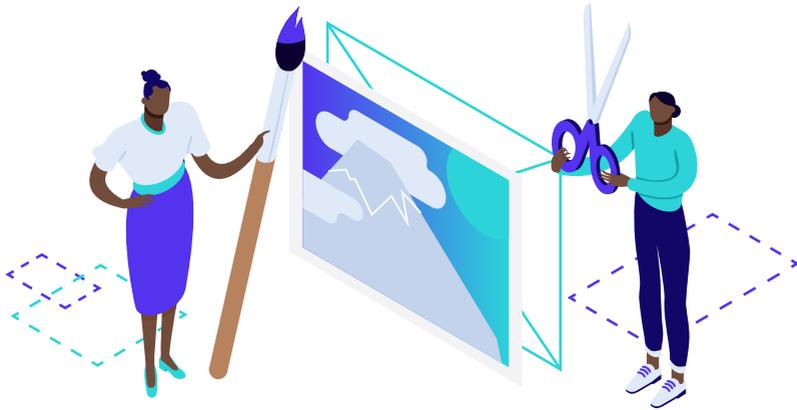
Seite bei einem HIT-Cache-Verhältnis von 96,2%, was sehr gut ist!

Im Abschnitt Bypasses des oberen Cache kannst du sehen, welche Requests nicht vom Cache bedient werden. Typischerweise können das CRON-Jobs, Admin-Ajax-Anfragen, E-Commerce-Checkout-Seiten, Query-Strings und UTM-Parameter sein etc.

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

KAPITEL 09:

Die Bildoptimierung ist ein Muss

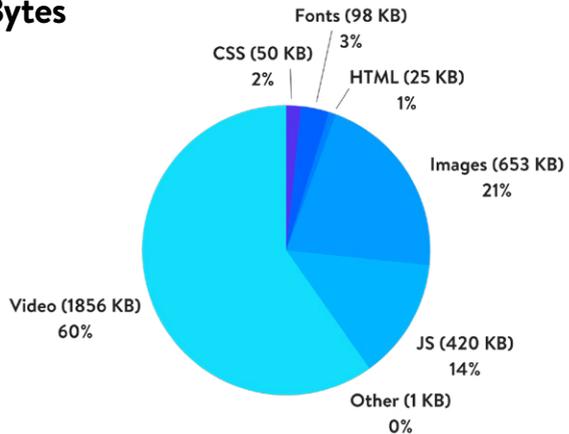


Die Bildoptimierung ist eine weitere einfache Sache, die man unternehmen kann, um einen erheblichen Einfluss auf die Ladezeiten der gesamten Seite zu haben. Das ist nicht optional; jede Seite sollte das tun!

Große Bilder verlangsamen deine Webseiten, was zu einem nicht optimalen Benutzererlebnis führt. Die Optimierung von Bildern ist der Prozess der Verringerung der Dateigröße, entweder mit einem Plugin oder einem Skript, was wiederum die Ladezeit der Seite beschleunigt. Verlorene und verlustfreie Kompression sind zwei gängige Methoden.

Laut HTTP Archive **machen Bilder ab November 2018 durchschnittlich 21% der Gesamtgröße einer Webseite aus.** Nach Videos, die viel schwieriger zu optimieren sind, sind Bilder bei weitem der erste Anlaufpunkt! Es ist wichtiger als JavaScript, CSS und Fonts. Und ironischerweise ist ein guter Bildoptimierungs-Workflow eines der am einfachsten zu implementierenden Dinge, aber viele Webseitenbetreiber übersehen das.

Durchschnittliche Bytes pro Webseite (KB)



Bereits im Dezember 2017 machten Bilder durchschnittlich 54% der Gesamtgröße einer Seite aus. Es scheint also, dass das Web als Ganzes bei der Bildoptimierung immer besser wird! Aber 21% ist immer noch eine Zahl, die nicht ignoriert werden darf. Wenn du keinen Videoinhalt auf deiner Webseite hast, sind Bilder wahrscheinlich immer noch dein größtes Laster in Bezug auf die Größe deiner Seite.

Bilder machen im Durchschnitt 21% der Gesamtgröße einer Webseite aus. Optimierte sie!

Finde die Balance (Dateigröße und Qualität)

Das Hauptziel der Formatierung deiner Bilder ist es, die **Balance zwischen der niedrigsten Dateigröße und der akzeptablen Qualität** zu finden. Es gibt mehr als eine Möglichkeit, fast alle dieser Optimierungen durchzuführen. Eine der grundlegendsten Möglichkeiten ist es, sie vor dem Hochladen auf WordPress zu komprimieren. Normalerweise kann das mit einem Tool wie Adobe Photoshop oder Affinity Photo erfolgen. Oder mit der neuen Squoosh-App von Google. Diese Aufgaben können aber auch automatisch über Plugins durchgeführt werden, auf die wir im Folgenden näher eingehen.

Die beiden wichtigsten Dinge, die zu berücksichtigen sind, sind **das Dateiformat und die Art der Kompression**, die du verwendest. Durch die Wahl der richtigen Kombination aus Dateiformat und Kompressionstyp kannst du deine Bildgröße um bis zu dem fünffachen reduzieren. Du musst mit jedem Bild oder Dateiformat experimentieren, um zu sehen, was am besten funktioniert.

Bevor Du anfängst, deine Bilder zu bearbeiten, stelle sicher, dass du den besten Dateityp ausgewählt hast. Es gibt mehrere Arten von Dateien, die man verwenden kann:

- **PNG** – erzeugt Bilder in höherer Qualität, hat aber auch eine größere Dateigröße; wurde als verlustfreies Bildformat erstellt, kann aber auch verlustbehaftet sein.
- **JPEG** – verwendet verlustbehaftete und verlustfreie Optimierung. Du kannst die Qualitätsstufe für ein ausgewogenes Verhältnis von Qualität und Dateigröße anpassen.

Im Idealfall solltest du JPEG (oder JPG) für Bilder mit viel Farbe und PNG für einfache Bilder verwenden.

Was ist mit GIFs? Animierte GIFs machen immer Spaß, aber sie beeinträchtigen die Webleistung. Viele GIFs sind über 1 MB groß. Wir empfehlen, diese für Social Media und Slack aufzubewahren. Wenn es einen gibt, ohne den du in deinem Blog-Post nicht leben lassen kannst, solltest du animierte GIFs immer komprimieren.

Kompressionsqualität vs. Größe

Hier ist ein Beispiel dafür, was passieren kann, wenn man ein Bild zu stark komprimiert. Das erste ist die Verwendung einer sehr niedrigen Kompressionsrate, was zu höchster Qualität (aber größerer Dateigröße) führt. Das zweite ist die Verwendung einer sehr hohen Kompressionsrate, was zu einer sehr geringen Bildqualität (aber einer geringeren Dateigröße) führt. Hinweis: Das unveränderte Originalbild beträgt 2,06 MB.

Wie du sehen kannst, ist das erste Bild oben 590 KB groß. Das ist ziemlich groß für ein Foto! Es ist im Allgemeinen am besten, wenn du das Gesamtgewicht einer Webseite unter 1 oder 2 MB halten kannst. 590 KB wären schon ein Viertel davon. Das zweite Bild sieht schrecklich aus, aber dann sind es nur noch 68 KB. Was du tun solltest, ist ein geeignetes Mittelmaß zwischen der Kompressionsrate (Qualität) und der Dateigröße zu finden.



Geringe Kompression (hohe Qualität) JPG – 590 KB



Hohe Kompression (geringe Qualität) JPG – 68 KB



Mittlere Kompression (gute Qualität) JPG – 151 KB

Also haben wir das Bild wieder mit einer mittleren Kompressionsrate aufgenommen, und wie du unten sehen kannst, sieht die Qualität jetzt gut aus, und die Dateigröße beträgt 151 KB, was für ein hochauflösendes Foto akzeptabel ist. Dies ist fast 4x kleiner als das Originalfoto bei geringer Kompression. Wir versuchen, die meisten unserer Bilder unter der 100 KB-Marke zu halten,

um die beste Leistung zu erzielen.

Verlustbehaftete vs. verlustfreie Optimierung

Bei der Komprimierung ist es wichtig sich ins Gedächtnis zu rufen, dass es zwei Arten von Kompression gibt, die man verwenden kann: verlustbehaftet und verlustfrei.

Bei der verlustbehafteten Komprimierung **werden einige der Daten in deinem Bild**

entfernt. Folglich bedeutet das, dass du eine Verschlechterung (Qualitätsminderung oder was man auch als “verpixelt” bezeichnen würde) sehen könntest. Du musst also vorsichtig dabei sein, wie sehr du dein Bild reduzierst. Nicht nur wegen der Qualität, sondern auch, weil man den Vorgang nicht rückgängig machen kann. Tatsache ist aber, dass einer der großen Vorteile und Gründe der Popularität der verlustbehafteten Komprimierung ist, dass man **die Dateigröße um einen beträchtlichen Betrag reduzieren** kann.



Die verlustfreie Kompression **verringert** im Gegensatz zur verlustbehafteten Komprimierung **nicht die Bildqualität.** Wie ist das möglich? Das geschieht in der Regel durch das Entfernen unnötiger Metadaten (automatisch generierte Daten, die von dem Gerät erzeugt werden, das das Bild erfasst). Der größte Nachteil dieser Methode ist jedoch, dass es **keine signifikante Verringerung der Dateigröße bewirken wird.** Mit anderen Worten, es wird im Laufe der Zeit viel Speicherplatz einnehmen.

Du wirst mit dem experimentieren wollen, was am besten für dich funktioniert. Aber für die Mehrheit der Benutzer **empfehlen wir die verlustbehaftete Komprimierung,** da man ein Bild leicht weit über 70% (manchmal sogar über 90%!) ohne großen Qualitätsverlust komprimieren kann. Multipliziere das mit 15 Bildern auf einer Seite, und es wird eine wichtige Rolle bei der Reduzierung der Ladezeit deiner Webseite spielen.

Plugins für die Bildkomprimierung

Die großartige Nachricht ist, dass es einige erstaunliche WordPress Bild Kompressions Plugins gibt, mit denen du den gesamten Prozess automatisieren kannst. Hier sind einige Plugins, die wir empfehlen:

- **Imagify** (verlustbehaftet und verlustfrei – optimiert Bilder extern)
- **WP Smush** (verlustbehaftet und verlustfrei – optimiert Bilder extern)
- **EWWW Cloud** (verlustbehaftet und verlustfrei – optimiert Bilder extern)
- **ShortPixel** (verlustbehaftet und verlustfrei – optimiert Bilder extern)

Das Wichtigste bei der Auswahl eines Plugins zur Bildoptimierung ist, **eines zu verwenden, das Bilder extern auf deren Servern komprimiert und optimiert**. Das wiederum reduziert die Belastung deiner Webseite. Alle der oben genannten Plugins gehen diesen Weg.

Falls du neugierig bist, bei Kinsta verwenden wir das Imagify-Plugin. Es komprimiert Bilder automatisch, wenn wir sie in die WordPress-Medienbibliothek hochladen. Also müssen wir uns um nichts Sorgen machen. Im Laufe der Zeit kannst du ein Gefühl dafür bekommen, welche Bildkompressionsstufe du verwenden möchtest. Es bietet Normal, Aggressiv und Ultra an.

Wir verwenden den aggressiven Modus bei Kinsta und sehen typischerweise **60-70% Einsparungen** je nach Bild. Hinweis: Wir verwenden viel mehr PNGs als JPEGs, da die meisten unserer Bilder Symbole und Illustrationen sind und keine Fotos.

The screenshot shows two entries in the WordPress media library. Each entry includes a checkbox, a thumbnail, the image name, author, title, date, and optimization statistics. A blue arrow points to the 'VIEW DETAILS' button in each entry.

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

Wie viel schneller wird Deine WordPress-Seite sein, wenn du die Bildkompression verwendest? Es hängt alles von den Größen der Originalbilder ab und wie stark sie nach der Komprimierung reduziert werden. Wir haben jedoch einige Geschwindigkeitstests

durchgeführt und festgestellt, dass eine qualitativ hochwertige Bildkompressionslösung die **Seitenladezeiten um über 80% verkürzen kann!**

Le Lazy Loading

Wenn du viele Bilder hast, kannst du sie langsam laden lassen. Das ist eine Optimierungstechnik, die sichtbare Inhalte lädt, aber das Herunterladen und Rendern von Inhalten, die unter dem Seitenumbruch erscheinen, verzögert.

So funktioniert Lazy Loading:

- Der Browser erstellt das DOM der Webseite, ohne Bilder herunterzuladen und Videos vorzuladen.
- JavaScript wird verwendet, um zu bestimmen, welche Bilder heruntergeladen und welche Videos vorgeladen werden sollen, basierend auf dem Inhalt, der beim Laden der Seite zunächst sichtbar ist. Diese Bilder und Videos werden heruntergeladen und gegebenenfalls vorinstalliert.
- Das Herunterladen und Rendern von zusätzlichen Videos verzögert sich, bis ein Seitenbesucher die Seite nach unten scrollt und zusätzliche Inhalte angezeigt werden.

Das Endergebnis ist, dass Bilder nicht heruntergeladen und Videos geladen werden, bis sie tatsächlich benötigt werden. Dies kann eine deutliche Leistungssteigerung für Webseiten mit vielen hochauflösenden Bildern und eingebetteten Videos bewirken.

Glücklicherweise gibt es viele großartige kostenlose WordPress-Plugins, die das Aktivieren des lazy Loadings ganz einfach machen (alle sind auf WordPress.org verfügbar):

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

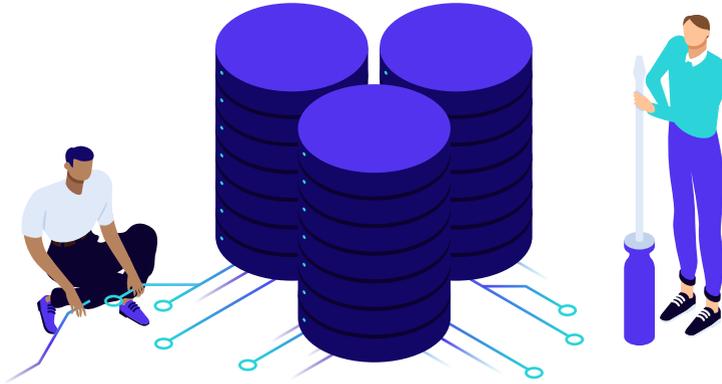
Zusätzliche Tipps zur Bildoptimierung

Hier sind ein paar letzte Tipps zur Bildoptimierung, mit denen du durchstarten kannst:

- Die Tage des Hochladens von Bildern, die nur auf die Breite der Spalte oder des DIV zugeschnitten sind, sind vorbei. Responsive Bilder funktionieren in WordPress (seit Version 4.4) sofort und zeigen mobilen Benutzern automatisch kleinere Bildgrößen an.
- SVGs können eine weitere großartige Alternative zur Verwendung von Bildern sein. Alle handgezeichneten Illustrationen, die du auf der Kinsta Webseite siehst, sind SVGs (Vektoren). SVGs sind in der Regel viel kleiner in der Dateigröße, wenn auch nicht immer. Schau dir unser Tutorial an, wie du SVGs auf deiner WordPress-Seite verwenden kannst.
- Benutze Symbolfonts, anstatt Text in Bildern zu platzieren – sie sehen besser aus, wenn sie skaliert werden und nehmen weniger Platz ein. Und wenn du einen Schriftgenerator verwendest, kannst du ihn noch weiter optimieren. Sieh dir an, wie wir die Größe unserer Icon-Fonts-Datei mit Hilfe eines Schriftgenerators **um satte 97,59% reduziert haben**.

KAPITEL 10:

Die Datenbank optimieren



Als nächstes folgen einige Tipps, wie man seine WordPress-Datenbank verfeinern kann. Genau wie ein Auto muss deine Datenbank gepflegt werden, da sie mit der Zeit aufgebläht wird.

Mitgliederseiten machen es besonders schwierig, da sie in der Regel **komplexere Abfragen erzeugen**, was wiederum zusätzliche Latenzzeiten beim Abrufen der Informationen aus der MySQL-Datenbank mit sich bringt. Das ist häufig auf die vielen unterschiedlichen Prozesse und die großen Mengen an Daten zurückzuführen, die bei diesem Seitentyp üblich sind. Das kann auch durch Webseiten verursacht werden, die stark auf Suchanfragen zur Navigation angewiesen sind oder `WP_Query` verwenden.

Nicht zu vergessen, sie haben auch eine große Anzahl gleichzeitiger Benutzer, die ständige die Datenbankabfragen erzeugen.

Verwende den InnoDB MySQL Storage Engine

Viele ältere Webseiten verwenden noch immer die MyISAM-Speichermaschine in ihrer Datenbank. In den letzten Jahren hat sich InnoDB als leistungsfähiger und zuverlässiger erwiesen.

*InnoDB ist wie synthetisches Öl, während MyISAM
sich mit regelmäßigen Abständen begnügt.*

Hier sind einige Vorteile von InnoDB gegenüber MyISAM:

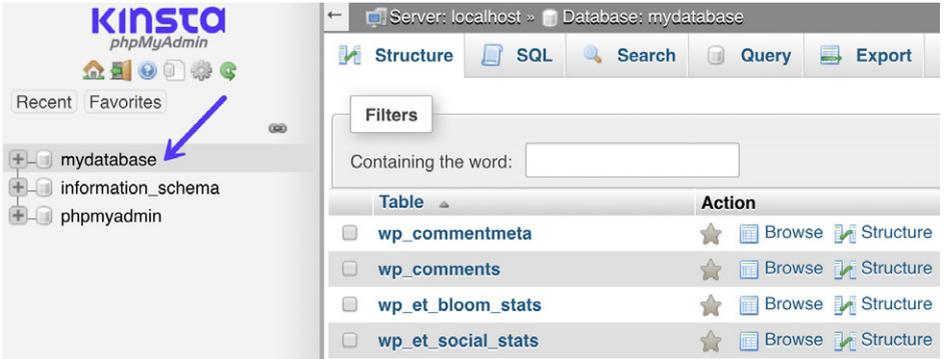
- InnoDB hat eine **Sperre auf Zeilenebene**. MyISAM hat nur eine vollständige Sperre auf Tabellenebene. Dies ermöglicht eine schnellere Bearbeitung der Anfragen.
- InnoDB hat die so genannte referentielle Integrität, die die **Unterstützung von Fremdschlüsseln** (RDBMS) und Beziehungsbeschränkungen beinhaltet, MyISAM nicht (DMBS).
- InnoDB unterstützt **Transaktionen**, d.h. man kann Commit und Rollback durchführen. MyISAM nicht.
- InnoDB ist zuverlässiger, da es **Transaktionsprotokolle** für die automatische Wiederherstellung verwendet. MyISAM nicht.

Jetzt fragst du dich vielleicht, ob du InnoDB oder MyISAM verwendest? Wenn du auf einer ziemlich neuen WordPress-Seite arbeitest, besteht die Chance, dass du bereits den InnoDB MySQL Storage Engine verwendest. Aber mit älteren WordPress-Seiten solltest du vielleicht einen kurzen Check machen. Einige Seiten haben vielleicht sogar MyISAM- und InnoDB-Tabellen gemischt und abgeglichen, in denen du Verbesserungen erzielen kannst, indem du einfach alles konvertierst.

Befolge diese einfachen Schritte unten, um zu überprüfen.

Schritt 1

Melde dich bei phpMyAdmin an und klicke auf deine MySQL-Datenbank.

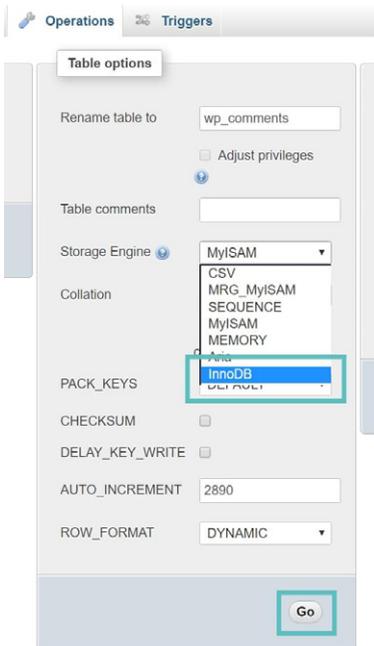


Schritt 2

Führe einen Schnellscan oder eine Sortierung der Spalte „Typ“ durch, und du kannst sehen, welche Speicher-Engine Typen deine Tabellen verwenden. In diesem Beispiel unten ist ersichtlich, dass zwei der Tabellen noch MyISAM verwenden.

Table	Engine	Charset	Collation	Size
wp_commentmeta	InnoDB	utf8	unicode_ci	64 KiB
wp_comments	InnoDB	utf8	unicode_ci	48 KiB
wp_et_bloom_stats	InnoDB	utf8	unicode_ci	64 KiB
wp_et_social_stats	InnoDB	utf8	unicode_ci	80 KiB
wp_et_bloom_stats	InnoDB	utf8	unicode_ci	48 KiB
wp_et_social_stats	InnoDB	utf8	unicode_ci	32 KiB
wp_et_bloom_stats	MyISAM	utf8mb4	unicode_ci	80 KiB
wp_et_social_stats	MyISAM	utf8mb4	unicode_ci	208 KiB
wp_et_bloom_stats	InnoDB	utf8mb4	unicode_ci	48 KiB
wp_et_social_stats	InnoDB	utf8mb4	unicode_ci	16 KiB

Wenn du welche gefunden hast, dann ist es wahrscheinlich an der Zeit, sie nach InnoDB zu verschieben. Wir empfehlen immer, sich an den eigenen Host zu wenden und ihn zu fragen, ob er dies für einen tun kann. Bei Kinsta werden die Datenbanktabellen jedes Kunden von unserem Migrationsteam automatisch in InnoDB umgewandelt.

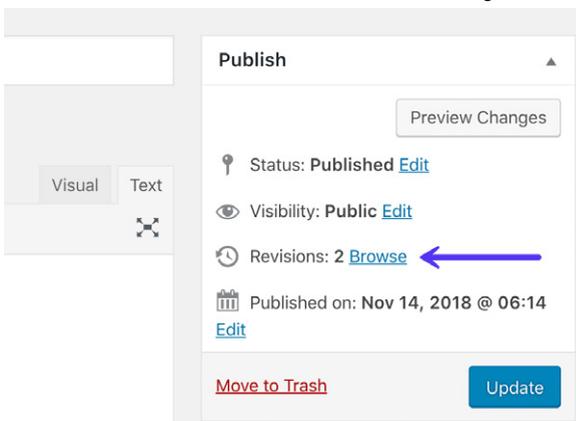


Du kannst sie aber auch manuell mit phpMyAdmin konvertieren. Klicke einfach auf die myISAM-Tabelle, klicke auf die Registerkarte “Operationen” und ändere die Speicher-Engine.

Wenn du ein Entwickler bist, wäre eine weitere Alternative, diese mit WP-CLI zu konvertieren.

Entferne und limitiere Seiten und Post Revisionen

Wann immer du eine Seite oder einen Beitrag in WordPress speicherst, erstellt es eine



sogenannte Revision. Das geschieht sowohl bei Entwürfen als auch bei bereits veröffentlichten Beiträgen, die aktualisiert werden. Revisionen können hilfreich sein, wenn du zu einer früheren Version deines Inhalts zurückkehren musst.

Änderungen können jedoch auch die Leistung deiner WordPress-Seite beeinträchtigen. Auf großen Webseiten kann das sehr schnell zu Tausenden von Zeilen in deiner Datenbank führen, die nicht unbedingt benötigt werden. Und je mehr Zeilen du hast, desto größer ist deine Datenbank, was Speicherplatz in Anspruch nimmt. Obwohl Indizes gerade für diesen Zweck erstellt wurden, können sie WordPress-Seiten trotzdem lähmen. Es gibt jedoch ein paar Dinge, die du tun kannst.

1. Entferne alte Revisionen

Wenn du eine ältere WordPress-Seite mit vielen Seiten und Beiträgen hast, könnte es an der Zeit sein, eine schnelle Bereinigung durchzuführen und diese alten Revisionen zu löschen. Du kannst dies mit MySQL tun, aber mit all den schlechten Codeausschnitten, die im Web herumfliegen, empfehlen wir ein Backup deiner Webseite und die Verwendung eines kostenlosen Plugins wie WP-Sweep.

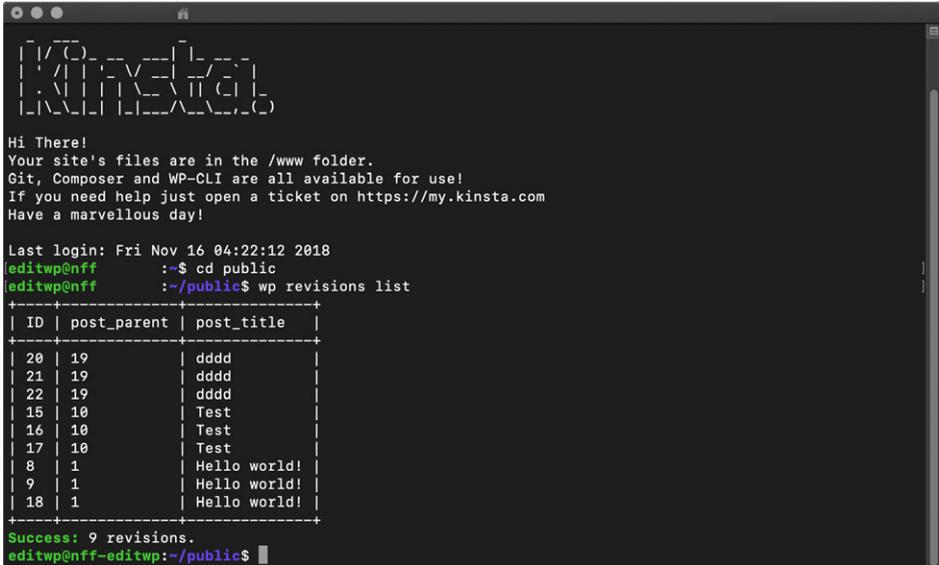
Ein weiteres unserer Lieblings-Plugins, WP Rocket, verfügt ebenfalls über eine Datenbankoptimierungsfunktion, um Revisionen zu entfernen.

The screenshot shows the WP Rocket interface with the 'Database' section active. The sidebar on the left contains navigation links for Dashboard, Cache, File Optimization, Media, Preload, Advanced Rules, Database, and CDN. The main content area is titled 'Database' and includes a 'Post Cleanup' section with a warning: 'Post revisions and drafts will be permanently deleted. Do not use this option if you need to retain revisions or drafts.' Below this, there are three items: 'Revisions' (9 revisions in your database), 'Auto Drafts' (1 draft in your database), and 'Trashed Posts' (0 trashed posts in your database). A blue arrow points to the 'Revisions' item. There is also a 'Comments Cleanup' section with a warning: 'Spam and trashed comments will be permanently deleted.' and one item: 'Spam Comments' (0 spam comments in your database). A 'Show Sidebar' toggle is in the top right, and a 'NEED HELP?' link is in the top right of the main content area.

Falls Du mit WP-CLI vertraut bist, gibt es ein paar Befehle, die du dafür verwenden kannst.

Melde dich über SSH bei deinem Server an und führe den folgenden Befehl aus, um die Anzahl der Revisionen, die sich derzeit in der Datenbank befinden, abzurufen und zu sehen.

```
wp revisions list
```



```

  _--_
  | / ( ) |
  | / | ' \ | |
  | . \ | | \ |
  | \ \ \ | | \ \ \ |
  | \ \ \ | | \ \ \ |

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

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

Wenn du einen Fehler erhältst, musst du möglicherweise zuerst das Paket wp-revisions-cli Paket mit dem folgenden Befehl installieren:

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

Du kannst dann den folgenden Befehl ausführen, um die Revisionen zu bereinigen:

```
wp revisions clean
```

2. Revisionen limitieren

Eine weitere gute Strategie, die wir bei Kinsta verfolgen, ist die Begrenzung der Anzahl der Revisionen, die pro Beitrag oder Seite gespeichert werden können.

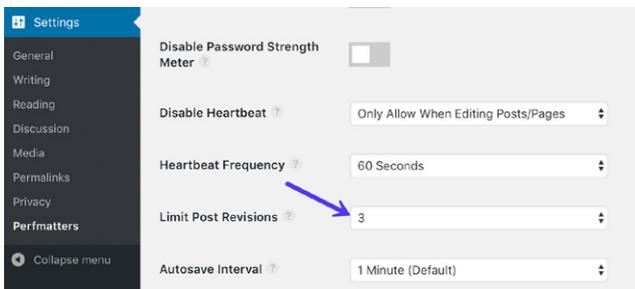
Selbst wenn man diese etwa auf zehn setzt, werden Revisionen nicht außer Kontrolle geraten, besonders, wenn man viele Updates durchführt.

Um Revisionen zu begrenzen, kannst du den folgenden Code zu deiner `wp-config.php`-Datei hinzufügen. Der untenstehende Code muss über dem ‚`ABSPATH`‘ eingefügt werden, sonst funktioniert er nicht. Du kannst die Nummer auf beliebig viele Revisionen ändern, die du in Deiner Datenbank speichern möchtest.

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

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

Line 28, Column 1



Oder du benutzt ein Plugin wie Perfmatters um das Limit einzustellen.

Limitiere die Revisionen mit Perfmatters

3. Revisionen deaktivieren

Und zu guter Letzt kannst du auch Revisionen auf deiner Seite komplett deaktivieren. In diesem Fall empfehlen wir dir dringend, der ersten Option oben zu folgen, um Revisionen zu löschen und sie anschließend zu deaktivieren. Auf diese Weise ist deine Datenbank völlig frei von allen alten Revisionen und es werden in Zukunft keine neuen mehr hinzugefügt.

Um Revisionen zu deaktivieren, kannst du den folgenden Code zu deiner `wp-config.php`-Datei hinzufügen. Der untenstehende Code muss über dem `'ABSPATH'` eingefügt werden, sonst funktioniert er nicht.

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

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

Line 33, Column 34

The screenshot shows the WordPress Settings interface with the 'Perfmatters' section expanded. The 'Limit Post Revisions' dropdown menu is set to 'Disable Post Revisions'. A blue arrow points to this dropdown. Other visible settings include 'Disable Password Strength Meter' (disabled), 'Disable Heartbeat' (set to 'Only Allow When Editing Posts/Pages'), 'Heartbeat Frequency' (60 Seconds), and 'Autosave Interval' (1 Minute (Default)).

Deaktiviere Revisionen mit Perfmatters

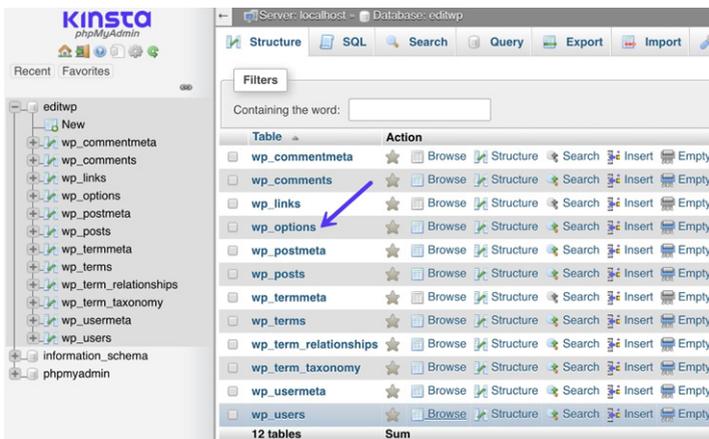
Oder du benutzt ein Plugin wie Perfmatters um Revisionen komplett zu deaktivieren.

Bereinige deine wp_options Tabelle und automatisch geladene Daten

Die Tabelle `wp_options` wird oft übersehen, wenn es um die allgemeine WordPress- und Datenbankleistung geht. Besonders auf älteren und großen Webseiten kann das leicht der Grund für langsame Suchanfragen auf deiner Webseite sein aufgrund von automatisch geladenen Daten, die von Plugins und Themes von Drittanbietern zurückgelassen werden. Vertraue uns, wir sehen das jeden Tag!

Die Tabelle `wp_options` enthält alle möglichen Daten für deine WordPress-Seite, wie z.B.:

- Site-URL, Home-URL, Admin-E-Mail, Standardkategorie, Beiträge pro Seite, Zeitformat, etc.
- Einstellungen für Plugins, Themes, Widgets etc.
- Temporär zwischengespeicherte Daten



Diese Tabelle enthält folgende Felder (columns):

- `option_id`
- `option_name`
- `option_value`
- **Autoload** (auf das fokussieren wir uns dann wenn es um Performance geht)

+ 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

Eines der wichtigsten Dinge, die man über die Tabelle `wp_options` wissen sollte, ist das Feld **autoload**. Dieses enthält einen Ja- oder Nein-Wert (Flag). Das steuert im Wesentlichen, ob es von der Funktion `wp_load_alloptions()` geladen wird oder nicht. Automatisch geladene Daten sind **Daten, die auf jeder Seite deiner WordPress-Seiten abgerufen werden**. Genauso wie wir dir gezeigt haben, wie du bestimmte Skripte vom standortübergreifenden Laden ausschließen kannst, gilt hier derselbe Ansatz. Das `autoload`-Attribut ist für Entwickler standardmäßig auf „yes“ gesetzt, aber nicht jedes Plugin sollte theoretisch seine Daten auf jeder Seite laden.

Das Problem, auf das WordPress-Seiten stoßen können, ist, wenn sich eine große Menge an automatisch geladenen Daten in der Tabelle `wp_options` befindet. Das ist in der Regel eine Konsequenz aus dem Folgenden:

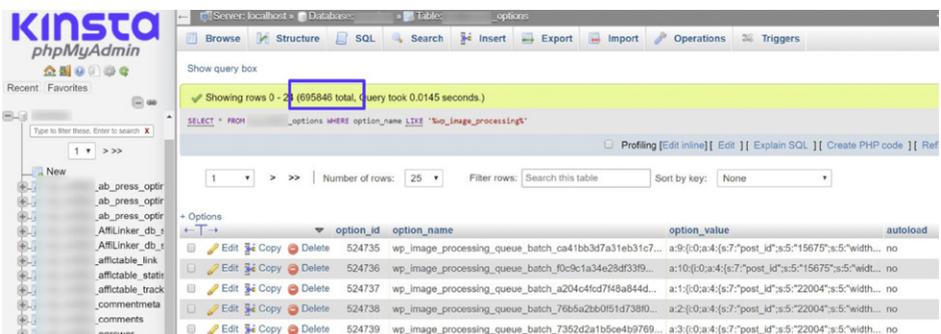
- Die Daten werden von einem Plugin automatisch geladen, auch wenn sie eigentlich auf „nein“ gesetzt werden könnten. Ein gutes Beispiel dafür ist ein Kontaktformular-Plugin. Muss es diese Daten auf jeder Seite laden oder würde die Kontaktseite reichen?
- Plugins oder Themes wurden von der WordPress-Seite entfernt, aber ihre Optionen bleiben in der Tabelle `wp_options` zurück. Das kann dazu führen, dass bei jeder Anforderung unnötige, automatisch geladene Daten abgefragt werden.
- Plugin- und Theme-Entwickler laden Daten in die Tabelle `wp_options` anstatt ihre eigenen Tabellen zu verwenden. Es gibt Argumente dafür und dagegen, da einige Entwickler Plugins bevorzugen, die keine zusätzlichen Tabellen erstellen. Die Tabelle `wp_options` ist jedoch auch nicht für Tausende von Zeilen ausgelegt.

Wie viel sind zu viele Daten? Das kann natürlich variieren, aber idealerweise möchtest du, dass diese zwischen 300 KB und 1 MB liegt. Sobald sie anfangen, sich dem Bereich von 3-5 MB oder mehr zu nähern, gibt es höchstwahrscheinlich Dinge, die optimiert oder entfernt werden können, wenn sie automatisch geladen werden. Und alles, was über 10 MB hinausgeht, sollte sofort in Angriff genommen werden. Das bedeutet nicht immer, dass es ein Problem verursachen wird, aber die Möglichkeit besteht.

Wann hast du das letzte Mal deine wp_options-Tabelle aufgeräumt? Ja.... das haben wir uns gedacht. Auf geht's!

Bereinigung von Transienten

Sofern du keinen Objekt-Cache verwendest, speichert WordPress transiente Datensätze in der Tabelle `wp_options`. Typischerweise erhalten diese eine Verfallszeit und sollten mit der Zeit verschwinden. Das ist jedoch nicht immer der Fall. Wir haben einige Datenbanken gesehen, in denen es Tausende von alten transienten Datensätzen gibt. Tatsächlich haben wir uns an einem Standort mit einigen korrupten transienten Datensätzen beschäftigt, **bei denen über 695.000 Zeilen in der Tabelle `wp_options` erzeugt wurden.**



The screenshot shows the phpMyAdmin interface with the following details:

- Database: wp_options
- Query: `SELECT * FROM wp_options WHERE option_name LIKE 'wp_image_processing*'`
- Result: Showing rows 0 - 25 (695846 total, query took 0.0145 seconds)
- Table structure:

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_f09c1a34e23df339...	a:10:{:0:a:4:{s:7:"post_id";s:5:"15675";s:5:"width...	no
524737	wp_image_processing_queue_batch_a204c4fd7f48a844d...	a:1:{:0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...	no
524738	wp_image_processing_queue_batch_76b5a2bb0151d7380...	a:2:{:0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...	no
524739	wp_image_processing_queue_batch_7352d2a1b5ee4b9769...	a:3:{:0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...	no

Es ist auch wichtig zu beachten, dass Transienten standardmäßig nicht automatisch geladen werden. Du könntest eine Abfrage wie die folgende verwenden, um zu sehen, ob es automatisch geladene transiente Daten gibt.

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

Eine bessere und sicherere Option wäre die Verwendung eines kostenlosen Plugins wie Transient Cleaner oder Delete Expired Transients, das nur die abgelaufenen Transienten aus der Tabelle `wp_options` bereinigen kann. Es scheint jedoch, dass es jetzt eine Funktion in WordPress gibt, die in 4.9 hinzugefügt wurde, dass sich um abgelaufene Transienten kümmert. Somit wird das hoffentlich zukünftig automatisch auf deiner Seite geschehen.

WP Rocket bietet auch die Möglichkeit, Transienten in ihren Datenbankoptimierungsoptionen zu bereinigen.

The screenshot shows the WP Rocket interface with a sidebar on the left containing menu items: ADVANCED RULES, DATABASE, CDN, HEARTBEAT, ADD-ONS, IMAGE OPTIMIZATION, and TOOLS. The main content area is titled 'Comments Cleanup' and includes a warning: 'Spam and trashed comments will be permanently deleted.' Below this are two checkboxes: 'Spam Comments' (0 spam comments) and 'Trashed Comments' (0 trashed comments). A blue arrow points to the 'Transients Cleanup' section, which has a warning: 'Transients are temporary options; they are safe to remove. They will be automatically regenerated as your plugins require them.' This section contains two checkboxes: 'Expired transients' (2 expired transients) and 'All transients' (24 transients).

WordPress Sitzungen bereinigen

Ein weiteres häufiges Problem, das wir gesehen haben ist, dass Cronjobs manchmal nicht synchronisiert oder nicht richtig ausgelöst werden, und Sitzungen daher nicht bereinigt werden. Es kann passieren, dass du Tonnen von `_wp_session_` Zeilen in deiner Datenbank ansammelst. In diesem Beispiel unten wurde die betreffende Webseite mit

über 3 Millionen Zeilen in der Tabelle `wp_options` abgeschlossen. Und die Tabelle war auf über 600 MB angewachsen.

<input type="checkbox"/>	<code>wp_links</code>	0	InnoDB	utf8mb4_unicode_ci	32 KiB
<input type="checkbox"/>	<code>wp_options</code>	~3,419,339	InnoDB	utf8mb4_unicode_ci	620.9 MiB
<input type="checkbox"/>	<code>wp_postmeta</code>	12,986	InnoDB	utf8mb4_unicode_ci	9.1 MiB
<input type="checkbox"/>	<code>wp_posts</code>	2,367	InnoDB	utf8mb4_unicode_ci	4 MiB

Du könntest eine Abfrage wie die untenstehende verwenden, um zu sehen, ob du auf dieses Problem gestoßen bist:

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

+ Options

	option_id	option_name	option_value	autoload
<input type="checkbox"/>	3301835	<code>_wp_session_254a6d16b4400d5aabf3c48b41b72d2b</code>	<code>a:1:{s:10:"edd_errors";s:0:""}</code>	no
<input type="checkbox"/>	3301836	<code>_wp_session_expires_254a6d16b4400d5aabf3c48b41b72d2b</code>	1520801306	no
<input type="checkbox"/>	3301837	<code>_wp_session_747c86ae52149d1068a60f240249f</code>	<code>a:1:{s:10:"edd_errors";s:0:""}</code>	no
<input type="checkbox"/>	3301838	<code>_wp_session_expires_747c86ae52149d1068a60f240249f</code>	1520801306	no
<input type="checkbox"/>	3301839	<code>_wp_session_17aa53659273873116d83993bae7cfd8</code>	<code>a:1:{s:10:"edd_errors";s:0:""}</code>	no
<input type="checkbox"/>	3301840	<code>_wp_session_expires_17aa53659273873116d83993bae7cfd8</code>	1520801306	no
<input type="checkbox"/>	3301841	<code>_wp_session_f3f29f822eec97951504e4d66afcd081</code>	<code>a:1:{s:10:"edd_errors";s:0:""}</code>	no
<input type="checkbox"/>	3301842	<code>_wp_session_expires_f3f29f822eec97951504e4d66afcd081</code>	1520801306	no
<input type="checkbox"/>	3301843	<code>_wp_session_2d4be84abb1e786ee229506e1c26e60e</code>	<code>a:1:{s:10:"edd_errors";s:0:""}</code>	no

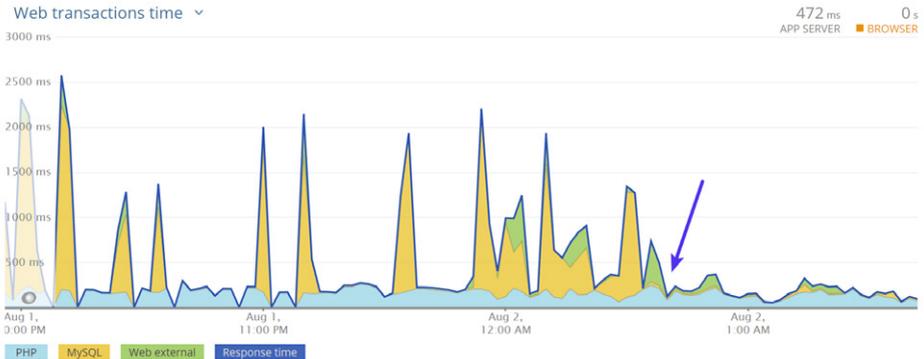
In den meisten Fällen kannst du diese dann mit dem folgenden Befehl sicher löschen (wie es ein Cron-Job getan haben sollte):

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

Nach der Bereinigung aller übrig gebliebenen `_wp_session_` rows Zeilen hatte die Tabelle weniger als 1.000 Zeilen und wurde auf 11 MB reduziert.

<input type="checkbox"/>	<code>wp_links</code>	0	InnoDB	utf8mb4_unicode_ci	32 KiB
<input type="checkbox"/>	<code>wp_options</code>	672	InnoDB	utf8mb4_unicode_ci	11.5 MiB
<input type="checkbox"/>	<code>wp_postmeta</code>	12,986	InnoDB	utf8mb4_unicode_ci	9.1 MiB
<input type="checkbox"/>	<code>wp_posts</code>	2,367	InnoDB	utf8mb4_unicode_ci	4 MiB

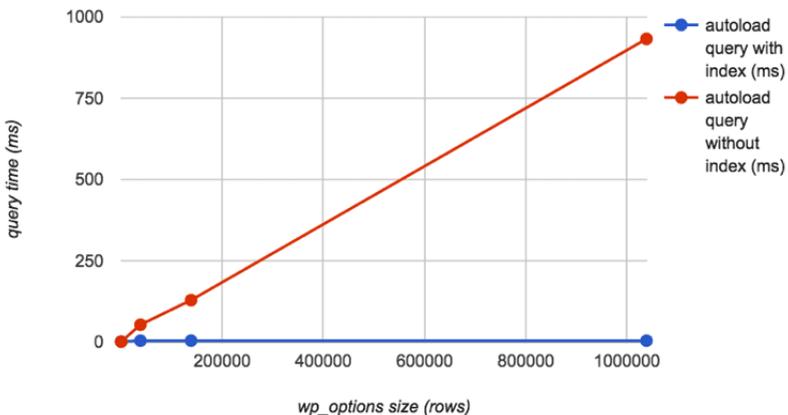
Es wurden auch die Spikes behoben, die die Seite in MySQL erhielt.



Füge einen Autload zum Index hinzu

Wenn das Bereinigen deiner `wp_options`-Tabelle nicht ausreicht, kannst du versuchen, dem Feld Autoload einen „Index“ hinzuzufügen. Das kann im Wesentlichen dazu beitragen, dass die Suche effizienter wird. Das großartige Team von 10up führte einige Tests an einer `wp_options`-Tabelle mit einer typischen Anzahl von autoload Datensätzen durch, um zu zeigen, wie das Hinzufügen eines Autoload-Index zu `wp_options`-Abfragen die Leistung steigern kann.

wp_options Größe vs query Zeit



Bildquelle: 10up

Nutze Redis als einen Persistenten Objekt Cache für WordPress

Redis ist ein Open-Source-Speicher für Datenstrukturen im Speicher. Im Rahmen von WordPress kann Redis verwendet werden, um die vom nativen Objekt-Cache von WordPress erzeugten Werte dauerhaft zu speichern, so dass zwischengespeicherte Objekte zwischen den Seitenladevorgängen wiederverwendet werden können.

Die Verwendung eines persistenten Objekt-Cache wie Redis ermöglicht die **Wiederverwendung von zwischengespeicherten Objekten**, anstatt dass die MySQL-Datenbank ein zweites Mal für dasselbe Objekt abgefragt werden muss. Das Ergebnis ist, dass Redis die Last auf die MySQL-Datenbank einer Webseite reduzieren, gleichzeitig die Antwortzeit der Webseite verkürzen und die Fähigkeit der Seite erhöhen kann, zu skalieren und zusätzlichen Traffic zu verarbeiten.



Hochdynamische Webseiten (WooCommerce, Mitgliederseiten, Foren, Diskussionsforen, Blogs mit extrem aktiven Kommentarsystemen), die das Page Caching nicht sinnvoll nutzen können, sind potenzielle Kandidaten für eine persistente Objekt-Caching Option wie Redis.

Wenn du Kinsta-Kunde bist, bieten wir ein Redis-Add-on an.

Nutze Elasticsearch um die WordPress Suchfunktion zu beschleunigen

Elasticsearch ist eine Open-Source-Volltextsuchmaschine. Es wird verwendet, um Daten zu indizieren und diese unglaublich schnell zu durchsuchen.

Im Rahmen von WordPress kann mit Elasticsearch die **Abfrage der WordPress-Datenbank beschleunigt werden**. Das geschieht, indem man einen Index des Inhalts der Datenbank deiner Webseite erstellt und dann Elasticsearch verwendet, um diesen Index viel schneller

zu durchsuchen, als eine MySQL-Abfrage in der Lage ist, die gleiche Suche durchzuführen.



Wenn du die Zeit und die Fähigkeit hast, kann Elasticsearch von einem sehr erfahrenen WordPress- und Elasticsearch-Entwickler in eine WordPress-Seite integriert werden. Wenn deine Webseite relativ standardmäßig WP_Query verwendet, kann Elasticsearch auch durch die Installation von ElasticPress integriert werden, einem kostenlosen WordPress-Plugin von 10Up, das auf WordPress.org verfügbar ist und automatisch in das WP_Query-Objekt integriert wird, um Abfrageergebnisse mit Elasticsearch und nicht mit MySQL zu generieren.

Jede Webseite, die WP_Query intensiv nutzt, kann von Elasticsearch profitieren. Beispiele für Webseiten, die von Elasticsearch profitieren können:

- Seiten, bei denen die Suche das primäre Navigationsmittel ist.
- WooCommerce-Seiten mit einer großen Anzahl von Bestellungen, bei denen die Administratoren der Webseite in der Lage sein müssen, die Liste der Bestellungen regelmäßig zu durchsuchen.
- Jede Webseite mit einer großen Anzahl von Beiträgen, bei der MySQL-Abfragen zu inakzeptabel langsamen Ergebnissen führen.

Wie bei Redis haben wir auch bei Elasticsearch ein Add-on.

Deaktiviere nicht essentielle Merkmale die Datenbanksensitiv sind

Das mag ein wenig offensichtlich erscheinen, aber es kann einen großen Unterschied machen, wenn du weniger wichtige Plugins und Themenfunktionen deaktivierst, die datenbankintensiv sind.

- Beliebte und/oder verwandte Post-Widgets und Plugins sind schrecklich. Sie haben typischerweise umfangreiche, seitenweiten Abfragen.

- Bildoptimierungs-Plugins, die Bilder mit deinem Server komprimieren; Du solltest immer ein Bildoptimierungs-Plugin verwenden, das Bilder extern optimiert.

Wenn du den Kinsta-Blog besuchst und bis zum Ende eines Beitrags scrollst, wirst du feststellen, dass wir sogenannte „handverlesene“ Artikel haben. Diese werden von uns manuell ausgewählt und dem Beitrag zugeordnet. Das reduziert die Abfrage auf fast nichts und beeinträchtigt nicht die Leistung deiner gesamten Webseite. Nimmt es mehr Arbeit in Anspruch? Ja, aber es ist hilfreich, da du wählen kannst, was die Leser sehen sollen.

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

Wie haben wir das geschafft? Wir haben das erstaunliche Advanced Custom Fields Plugin verwendet und diese Felder dann unserem Blog-Post-Typ zugeordnet. Das ermöglicht es uns, jeden unserer Blog-Posts (wie unten gezeigt) zu durchsuchen und mit verwandten Inhalten zu versehen.

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

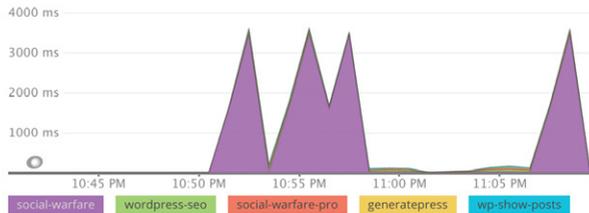
Wir empfehlen auch, sich von Plugins fernzuhalten, die einen View/Post-Counter deiner Webseite hinzufügen, es sei denn, du benötigst ihn unbedingt. Vermeide beispielsweise Dinge wie „792 Beiträge“ neben dem Avatar eines Benutzers in Forumsbeiträgen oder „5.243 Ansichten“, wenn du Forumsbeiträge auflistest. Wenn du eine lange Diskussion hast, werden diese Zähler eine enorme Belastung für deine Datenbank darstellen. Im Allgemeinen solltest du den Einsatz von Zählern minimieren und sie nur bei Bedarf verwenden.

Das gilt auch für viele Social Counters. Zum Beispiel, auf dieser Seite unten kannst du sehen, dass die Antwortzeit von dem beliebten Social Warfare Plugin 30x länger ist als das nächste Plugin darunter. Das Caching ist aktiviert, aber offensichtlich birgt dieses Plugin einen erheblichen Performanceverlust. Nachdem das Plugin auf der Webseite deaktiviert wurde, verbesserten sich die Ladezeiten sofort und die Reaktionsfähigkeit des WordPress Admin Dashboards wurde verbessert.

Sort by

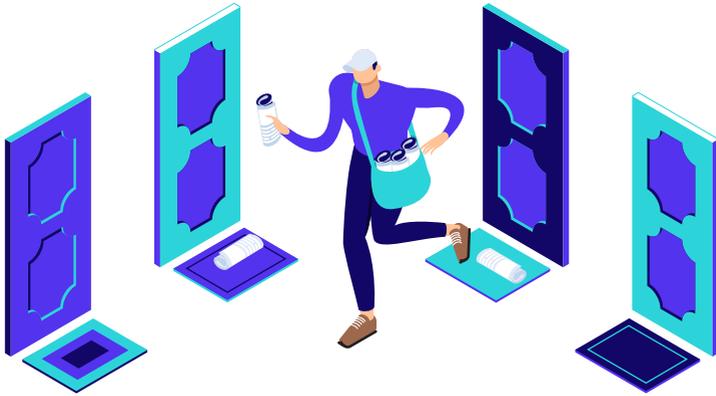
social-warfare	18.9 sec
wordpress-seo	624 ms
social-warfare-pro	449 ms
generatepress	425 ms
wp-show-posts	389 ms
aaawp	356 ms
gp-premium	229 ms
nogluten	160 ms
imagify	115 ms
wp-review	107 ms

Top 5 plugins and themes
by response time



KAPITEL 11:

Nutze ein Content Delivery Network (CDN)



CDN ist die Abkürzung für Content Delivery Network. Das ist ein Netzwerk von Servern (auch bekannt als POPs), die sich auf der ganzen Welt befinden. Sie wurden entwickelt, um Kopien der statischen (und manchmal auch dynamischen) Inhalte deiner WordPress-Seite wie Bilder, CSS, JavaScript und Videostreams bereitzustellen.

Erstens sind ein CDN und ein WordPress Host nicht zu verwechseln. Es handelt sich dabei um völlig getrennte Dienstleistungen. Ein CDN ist kein Ersatz für deinen Hosting-Provider, sondern eine zusätzliche Möglichkeit, die Geschwindigkeit deiner Webseite zu erhöhen. Während unser Hosting hier in Kinsta rasant schnell ist, kann ein CDN deine Webseite noch schneller machen.

Wie funktioniert ein CDN

Wie funktioniert ein CDN genau? Nun, wenn du zum Beispiel deine Webseite bei Kinsta hosten möchtest, musst du einen physischen Rechenzentrumsstandort wählen, wie z.B. die USA, Europa, Asien-Pazifik oder Südamerika.

Angenommen, du entscheidest dich für Zentralamerika. Das bedeutet, dass sich deine Webseite physisch auf einem „Host-Server“ in Council Bluffs, Iowa, befindet. Wenn Menschen in Europa deine Webseite besuchen, wird es länger dauern, bis sie geladen

ist, als wenn jemand sie von Dallas, TX, besucht.

Warum? Weil die Daten eine weitere Strecke zurücklegen müssen. Das ist die sogenannte Latenzzeit. Latenzzeit bezieht sich auf die Zeit und/oder Verzögerung, die bei der Übertragung von Daten über ein Netzwerk anfällt. Je weiter der Abstand, desto größer die Latenzzeit.

Verschiedene CDN Arten

Es gibt zwei verschiedene Arten von Content Delivery Networks:

1. Traditioneller Pull CDN
2. Reverse Proxy CDN

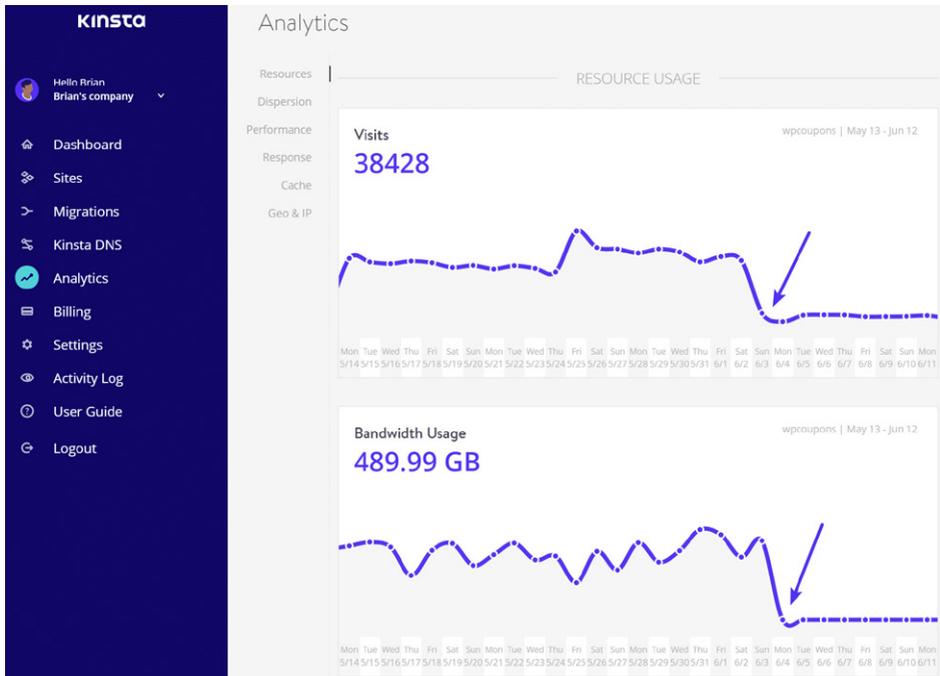
Traditionelle Pull-CDNs speichern eine Kopie all deiner Inhalte und Medien im Cache, aber eine Anfrage des Kunden wird immer noch direkt an deinen Hosting-Provider gerichtet. KeyCDN und CDN77 sind Beispiele für traditionelle CDNs.

Ein Reverse-Proxy-CDN ist etwas anders. Während es sich immer noch wie ein CDN verhält, fängt es alle eingehenden Anfragen ab und fungiert als Zwischenserver zwischen dem Client und dem Host. Cloudflare und Sucuri sind Beispiele für Reverse-Proxy-CDNs. Dies ist ein Grund, warum du dein DNS direkt auf diese Anbieter und nicht auf deinen Host verweisen musst.

Der Vorteil hier ist, dass sie als Zwischenserver fungieren und somit auch als Firewall dienen können, und etwa unerwünschten Traffic blockieren können, sodass dieser Deinen Host und somit Deine WordPress-Seite gar nicht erst erreicht. Ein Nachteil hingegen ist, dass sie im Vergleich zu einem herkömmlichen Pull-CDN mit zusätzlichem Leistungsaufwand verbunden sind. Aber mit zusätzlichen Leistungs- und Sicherheitsfunktionen kann das als vernachlässigbar angesehen werden.

Nachfolgend wird anhand eines Beispiels gezeigt, was nach der Aktivierung von Sucuri auf der Webseite eines Kunden passiert. Wie du sehen kannst, hat es einen enormen

Einfluss auf die Menge an schlechtem Traffic gehabt, der durchkommt. Letztendlich können diese Arten von Diensten dazu beitragen, dass du Hostingkosten einsparst.



CDN Speed Tests

Vorhin haben wir über die großen Vorteile des WordPress-Caching gesprochen. Nun, CDN-Caching ist auch super mächtig. Das liegt daran, dass CDNs in der Regel viel mehr Serverstandorte haben als Hosting-Provider. Das bedeutet, dass sie alle deine Assets (Bilder, JS, CSS) näher bei deinen Besuchern zwischenspeichern und mit rasanter Geschwindigkeit bereitstellen können.

Lass uns ein paar Kurztests durchführen, um zu sehen, wie viel schneller deine Seite mit einem CDN sein könnte.

Ohne CDN

Unsere Test-Webseite wird bei Kinsta gehostet und befindet sich physisch im Rechenzentrum in Iowa, USA. Wir haben zuerst fünf Geschwindigkeitstests in Pingdom durchgeführt (ohne aktiviertes CDN) und den Durchschnitt genommen. Wichtig: Wir nutzen den Standort Europa – Großbritannien – London in Pingdom, um die wahre Leistungsfähigkeit eines CDNs zu demonstrieren. **Die Gesamtladezeit betrug 1,03 s.**



Performance grade B 89	Page size 231.0 KB
Load time 1.03 s	Requests 40

Mit CDN

Wir aktivierten dann unser CDN und führten fünf weitere Geschwindigkeitstests in Pingdom durch. Unsere **Gesamtladezeit beträgt nun 585 ms** vom Testort Europa - Großbritannien - London Pingdom. Durch die Verwendung des CDN konnten wir also unsere **Seitenladezeiten um 43,2% verringern!** Das ist ein Wahnsinn.

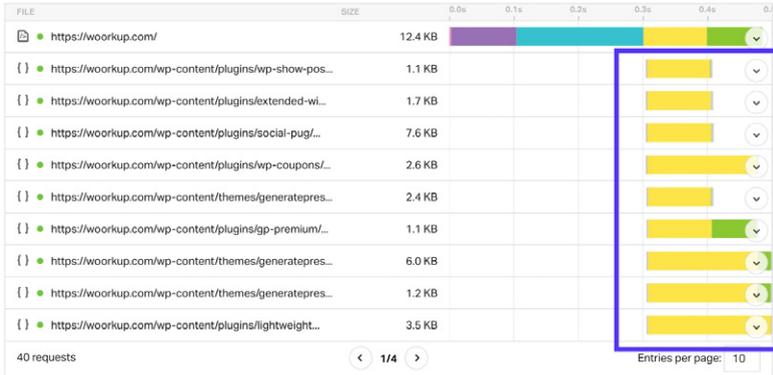


Performance grade B 89	Page size 230.6 KB
Load time 585 ms	Requests 40

Der Grund für diesen drastischen Unterschied ist, dass das CDN über ein Rechenzentrum in London verfügt. Das bedeutet, dass alle Assets an diesem Ort zwischengespeichert werden und mit minimaler Latenz bedient werden können.

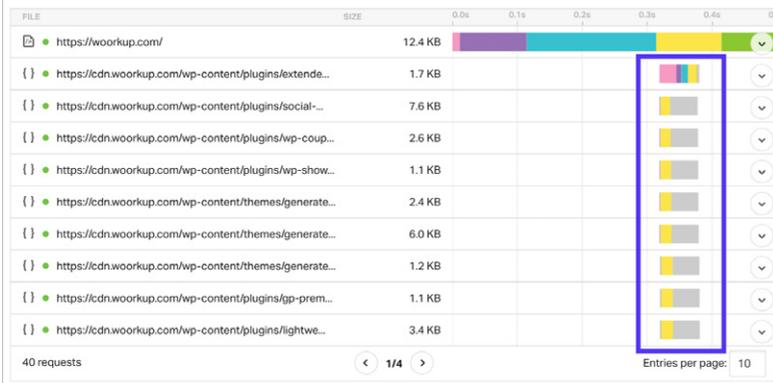
TTFB ohne CDN

Denk daran, dass der gelbe Balken in Pingdom für Wartezeit steht, also Zeit bis zum ersten Byte (TTFB). Bei unseren Geschwindigkeitstests ohne das laufende CDN lag der durchschnittliche TTFB auf Assets bei **etwa 98 ms**.



TTFB mit CDN

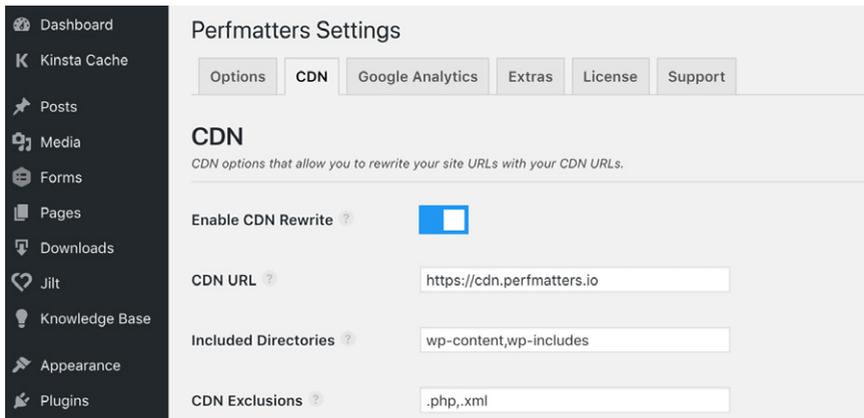
Sobald wir das CDN aktiviert haben, sank der durchschnittliche **TTFB auf durchschnittlich 15 ms**. Durch die Verwendung eines CDN sank unser durchschnittlicher **TTFB also um 84,69%**. Dies liegt vor allem daran, dass die Assets direkt aus dem Cache des CDN bedient wurden.



Ein CDN hat unsere Ladezeiten um 43,2% reduziert! Überprüfe, warum du ein CDN verwenden solltest.

Wie man ein CDN aktiviert

Wenn du ein traditionelles Pull-CDN verwendest, kannst du kostenlose Plugins wie CDN Enabler, WP Rocket oder Perfmatters verwenden, um es in deine WordPress-Seite zu integrieren. Diese Plugins verknüpfen deine Assets automatisch mit dem CDN. Es ist keine zusätzliche Arbeit deinerseits erforderlich, um deine Inhalte auf das CDN zu bringen; das funktioniert alles von selbst! Reverse Proxy-CDNs benötigen in der Regel keine Plugins, obwohl sie manchmal über sie verfügen, um zusätzliche Funktionen zu aktivieren.



The screenshot shows the 'Perfmatters Settings' page in a WordPress dashboard. The left sidebar contains navigation links: Dashboard, Kinsta Cache, Posts, Media, Forms, Pages, Downloads, Jilt, Knowledge Base, Appearance, and Plugins. The main content area is titled 'Perfmatters Settings' and has tabs for 'Options', 'CDN', 'Google Analytics', 'Extras', 'License', and 'Support'. The 'CDN' tab is active, showing the following settings:

- Enable CDN Rewrite**: A toggle switch that is currently turned on (blue).
- CDN URL**: A text input field containing 'https://cdn.perfmatters.io'.
- Included Directories**: A text input field containing 'wp-content,wp-includes'.
- CDN Exclusions**: A text input field containing '.php,.xml'.

Aktiviere CDN in WordPress mit Perfmatters

Wie man das CDN bei Kinsta aktiviert

Hat dir die CDN-Geschwindigkeitsprüfung oben gefallen? Wir haben KeyCDN in diesen Tests verwendet. Die tolle Nachricht ist, dass unser Kinsta CDN auf KeyCDN basiert. Es handelt sich um ein HTTP/2- und IPv6-fähiges Content-Delivery-Netzwerk mit

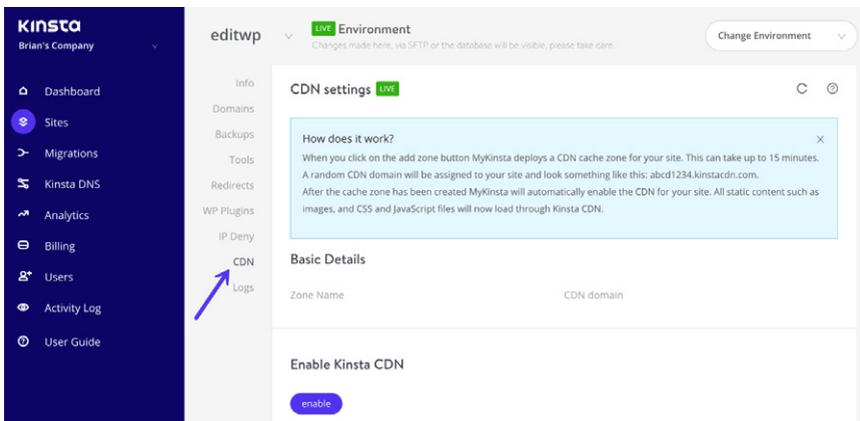
34 Standorten, das deine Assets und Medien rund um den Globus auf Hochtouren befördert. Zu den derzeit bedienten Regionen gehören Amerika, Südamerika, Europa, Afrika, Asien und Australien.



Wenn du ein Kinsta-Kunde bist, integrieren wir kostenlose CDN-Bandbreite in alle unsere Hosting-Angebote. Du kannst das Kinsta CDN in zwei einfachen Schritten aktivieren.

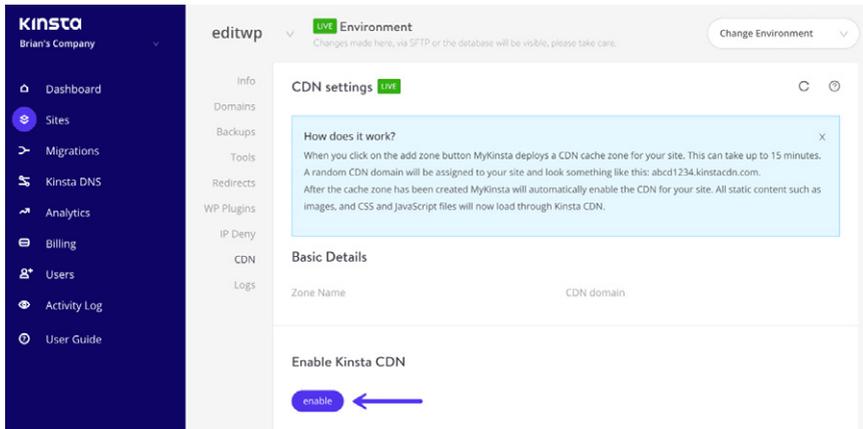
Schritt 1

Melde dich zunächst in deinem MyKinsta Dashboard an. Klicke dann auf deine Webseite und dann auf die Registerkarte CDN.



Schritt 2

Klicke dann auf „Aktivieren“. Nach einigen Minuten wird das CDN automatisch bereitgestellt, und deine Assets werden aus dem Cache rund um den Globus bereitgestellt. Das war's dann auch schon.



Zusätzliche CDN Optimierungen

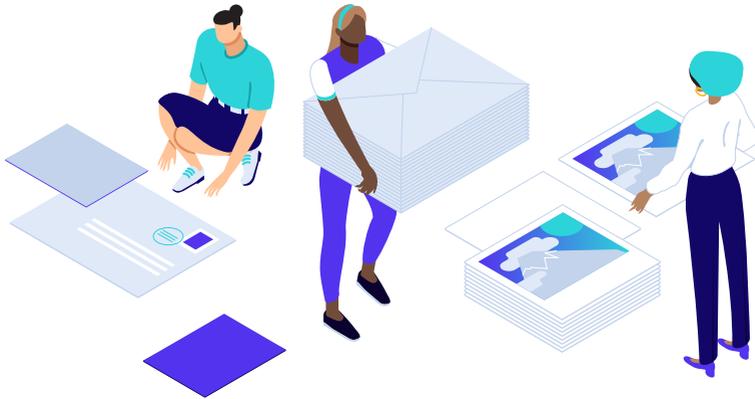
Hier sind ein paar zusätzliche CDN-Optimierungen, die du vielleicht ausprobieren oder in Betracht ziehen möchtest.

- Wenn man viele Kommentare hat, können Gravatare viele Anfragen generieren. Sie werden vom `secure.gravatar.com` geladen. In diesem Tutorial erfährst du, wie du stattdessen Gravatare aus deinem CDN laden kannst. Das tun wir auch auf der Kinsta-Webseite.
- Du kannst deine benutzerdefinierten Webfonts von deinem CDN oder sogar von Google-Fonts auf deinem CDN hosten. Schau dir unser ausführliches Tutorial über lokale Schriften an. Dieses findest du auf <https://kinsta.com/de/lernen/wordpress-beschleunigen>.

Achte darauf, dass du dein Favicon von deinem CDN lädst. Auch wenn es klein ist, jede Anfrage zählt!

KAPITEL 12:

Medien und Emails auslagern falls nötig



Alles, was eine Anfrage generiert, hat auf die eine oder andere Weise Auswirkungen auf die Leistung deiner Webseite. Für Webseiten mit Hunderttausenden von Dateien oder großen Medien kann es sinnvoll sein, das vollständig auszulagern. Das Auslagern unterscheidet sich von der Bereitstellung über ein CDN. Bei einem CDN befinden sich die Originaldaten noch auf deinem Host, das CDN hat einfach mehrere Kopien davon.

Wenn das Caching für deine CDN-Assets abläuft, fragt es deinen Host nach den neuesten Kopien der Dateien. Aber da sie auf so viele POPs setzen, müssen Dateien häufig erneut in den Cache geladen werden, weil dieser in verschiedenen Regionen ausläuft. Das könnte eine Menge zusätzlicher Abfragen bedeuten.

Wenn du Medien oder Dateien auslagerst, bedeutet das, dass du den ursprünglichen physischen Speicherort von ihnen tatsächlich von deinem Hosting-Provider verlagerst. Während es also den Anschein haben mag, dass die Dateien von deiner Webseite aus bereitgestellt werden, befinden sie sich in Wirklichkeit an einem anderen Ort. Neben der Reduzierung zusätzlicher Anfragen an den Host ist der wichtigste Grund natürlich, auch Speicherplatz zu sparen.

Medien zu Amazon S3 auslagern

Eine der beliebtesten Auslagerungslösungen ist Amazon S3.

Amazon S3 ist eine Speicherlösung und eines vieler Amazon Web Services Produkte. Typischerweise wird das für große

Webseiten verwendet, die entweder zusätzliche Backups benötigen oder große Dateien bereitstellen (Downloads, Software, Videos, Spiele, Audiodateien, PDFs, etc.). Amazon hat sich als sehr zuverlässig erwiesen und kann aufgrund seiner massiven Infrastruktur sehr niedrige Speicherkosten anbieten. Zu den Kunden von S3 gehören unter anderem Netflix, Airbnb, SmugMug, Nasdaq etc.



Da sie sich ausschließlich mit Massenspeicher befassen, können sie fast garantieren, dass die Preise billiger sind als bei deinem WordPress Host. Das Offloading von Medien auf AWS kann eine sehr gute Möglichkeit sein, Geld zu sparen und ist für das erste Jahr kostenlos (bis zu 5 GB Speicherplatz). Da die Anfragen für deine Medien direkt von Amazon bedient werden, wird deine WordPress-Seite weniger belastet, was zu schnelleren Ladezeiten führt.

Wir empfehlen die Verwendung des WP Offload Media Lite for Amazon S3 Plugins (verfügbar auf WordPress.org). Das Plugin kopiert Dateien von deiner WordPress-Seite nach Amazon S3 automatisch, wenn sie in die Medienbibliothek hochgeladen werden.

Medien in den Google Cloud Speicher auslagern



Eine weitere beliebte Offloading-Lösung ist Google Cloud Storage. Da Kinsta von der Google Cloud Plattform betrieben wird, sind wir große Fans der Technologie und Infrastruktur. Aufgrund der massiven Infrastruktur von Google und der Tatsache, dass sie sich mit Massenspeichern befassen, können sie sehr niedrige Speicherkosten anbieten. Zu den Kunden gehören unter anderem Spotify, Vimeo, Coca-Cola, Philips, Evernote und Motorola.

Wir empfehlen dazu das kostenlose WP-Stateless Plugin (verfügbar auf WordPress.org).

Das Plugin kopiert Dateien von deiner WordPress-Seite automatisch in den Google Cloud Storage, wenn sie in die Medienbibliothek hochgeladen werden. Es bietet dann verschiedene Modi, wie z.B. das Sichern einer Kopie deiner Medien oder sogar das Löschen der Medien aus WordPress und das Bereitstellen aus dem Google Cloud Storage.

Transaktions- und Marketingemails auslagern

Ganz gleich, ob du davon überzeugt bist oder nicht, E-Mails haben Auswirkungen auf deine Server- und Serverressourcen. Bei einigen Hosts, insbesondere bei Shared Hosts, könnte ein Missbrauch sogar zur Suspendierung führen. Das wird insbesondere bei denen, die versuchen, Massen-E-Mails zu versenden, zu einem Problem. Das ist der Grund, warum es Drittanbieter für transaktionale E-Mail-Provider gibt und warum viele Hosting-Provider die E-Mail-Zustellung auf Standardports insgesamt blockieren. Wir raten davon ab, deinen Hosting-Provider für E-Mails zu verwenden.

Wenn du Newsletter oder Massen-E-Mails versendest, empfehlen wir immer die folgenden Alternativen, um die besten Ergebnisse zu erzielen:

- Verwende eine **professionelle E-Mail-Marketing-Software von Drittanbietern**, die nicht Teil von WordPress ist.
- Verwende einen **transaktionalen E-Mail-Dienstleister** (HTTP-API oder SMTP) zusammen mit WordPress.

Weitere Vorteile der Nutzung eines Drittanbieter-Service sind:

- Bessere Zustellbarkeit von E-Mails. Lass die E-Mail-Anbieter das tun, was sie am besten können!
- Bessere Chancen, schwarze Listen zu vermeiden.
- Es ist möglicherweise nicht immer möglich, DMARC-Einträge mit deinem Hosting-Provider einzurichten.

Email Marketing Tools

Einige Beispiele für Marketing-E-Mails sind Newsletter, Produkt- und Feature-Ankündigungen, Verkäufe, Event-Einladungen, Onboard-Erinnerungen etc. Hier sind ein paar E-Mail-Marketing-Tools, die wir empfehlen:

- **MailChimp** – Wir verwenden MailChimp bei Kinsta.
- **MailerLite**
- **Drip**

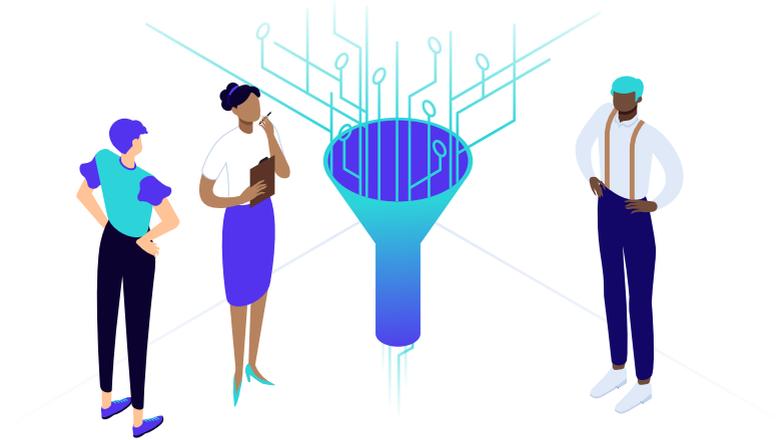
Emaldienste für Transaktionsmails

Einige Beispiele für transaktionale E-Mails sind Kaufbelege von WooCommerce oder EDD, Benachrichtigungen zur Kontoerstellung, Versandbenachrichtigungen, App-Fehlermeldungen, Passwort-Rücksetzungen usw. Wenn du ein Kinsta-Kunde bist, verlassen wir uns auf einen externen SMTP-Anbieter, um eine hohe Lieferfähigkeit zu gewährleisten. Aber abhängig von deinem Volumen empfehlen wir immer, dieses außerhalb zu verschieben. Hier sind ein paar Transaktions-E-Mail-Dienste, die wir empfehlen:

- **SendGrid** – Wir verwenden SendGrid bei Kinsta.
- **Mailgun**
- **SparkPost**

KAPITEL 13:

Wie man Engpässe und langsame Plugins findet

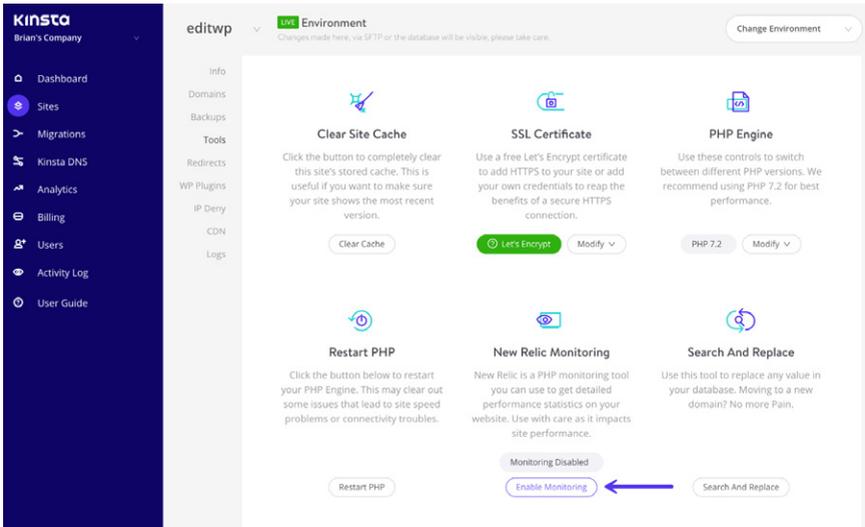


Nun werden wir dir einige Tipps offenbaren, wie man Engpässe auf deiner WordPress-Seite findet und was du dagegen tun kannst.

Verwende New Relic um langsame Plugins und Datenbankabfragen zu identifizieren

Es gibt einige großartige Tools auf dem Markt, die dir helfen können, langsame Datenbankabfragen und Plugins zu lokalisieren und zu identifizieren, die viel Zeit in Anspruch nehmen. Wir sind große Fans von New Relic bei Kinsta und nutzen es täglich. New Relic ist ein PHP-Monitoring-Tool, mit dem du detaillierte Leistungsstatistiken auf deiner Webseite erhalten kannst.

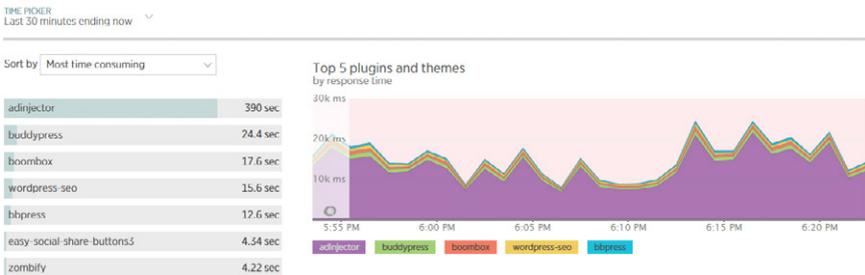
Wenn du ein Kinsta-Kunde bist, kannst du sogar deinen eigenen New Relic-Lizenzschlüssel in unserem MyKinsta-Dashboard hinzufügen.



Verwende New Relic jedoch mit Vorsicht, da es die Leistung der Webseite beeinträchtigt. Es fügt außerdem JavaScript zu deiner Webseite hinzu. Wir empfehlen, es zu aktivieren, wenn du Probleme bei der Performance beheben musst, und es danach zu deaktivieren.

Langsame Plugins finden

Wenn ein WordPress-Plugin eine allgemeine Verlangsamung verursacht, variieren die Symptome je nach Aktivität, die das Plugin ausführt. In vielen Fällen wirst du jedoch feststellen, dass ein langsames Plugin jede Seite einer WordPress-Webseite beeinflusst. Im Falle der Webseite, deren Daten du im Bild unten siehst, wurde allgemeine Langsamkeit auf jeder Frontend-Seite der Webseite festgestellt. Hier sieht man, was New Relic über die Leistung der Plugins auf der Webseite zu sagen hat.



Man erkennt sofort, dass das Adinjektor-Plugin mehr als 15-mal so viel Zeit verbraucht wie das nächst langsamere Plugin.

Beim Anblick solcher Daten kann es verlockend sein, das Plugin sofort als schlecht kodiert oder irgendwie unwirksam zu erachten. Obwohl das manchmal der Fall ist, muss das nicht zwingend sein. Fehlkonfiguration des Plugins, Langsamkeit der Datenbank oder externe Ressourcen, die nur langsam reagieren, können dazu führen, dass ein Plugin viel Zeit in Anspruch nimmt.

Wenn du also ein Plugin siehst, das langsam reagiert, ist es eine gute Idee, mehrere andere Screens in New Relic zu überprüfen, um zusätzliche Informationen zu sammeln. Die Transaktionen, Datenbanken und externen Ressourcen sollten alle überprüft werden, bevor entschieden wird, dass die Deaktivierung des Plugins der beste oder einzige Weg ist.

Allgemeine Verzögerungen durch überforderte Datenbanken

Eine schlecht optimierte Datenbank kann zu einer allgemeinen Langsamkeit auf einer WordPress-Seite führen. Weiter oben haben wir schon über zahlreiche Dinge berichtet, die man tun kann, um das Problem zu lösen. In New Relic wird diese datenbankbezogene Langsamkeit höchstwahrscheinlich an zwei Stellen auftreten:

- Zunächst siehst du in der Übersicht eine überdimensionale Menge an MySQL-Aktivität.
- Als nächstes werden eine oder mehrere Datenbanktabellen angezeigt, die viel Zeit in der Registerkarte Datenbanken verbrauchen.

Ausgehend vom Übersichtsscreen könnte eine Webseite mit einer belasteten Datenbank etwa so aussehen:

In diesem Fall weisen die Aufnahmen auf automatisch geladene Daten in der Tabelle `wp_options` hin. Vielleicht erinnerst du dich, wir haben das vorhin besprochen! Eine schnelle Analyse der Tabelle `wp_options` bestätigt, dass fast 250 MB Daten aus dieser Tabelle automatisch geladen werden, was diese Webseite zu einem offensichtlichen Kandidaten für die Wartung und Optimierung von Datenbanken macht.

Schau dir unser ausführliches Tutorial an, wie du mit New Relic, Performance-Probleme auf deiner WordPress-Seite beheben kannst.

Nutze das kostenlose Query Monitor Plugin

Du kannst auch das kostenlose Query Monitor WordPress Plugin verwenden. Mithilfe dieses Programms kann man langsame Datenbankabfragen, AJAX-Aufrufe, REST-API-Anfragen und vieles mehr identifizieren und prüfen. Darüber hinaus meldet das Plugin Webseite-Details wie Script Dependencies und Dependents, WordPress-Hooks, die während der Seitengenerierung ausgelöst wurden, Details zur Hosting-Umgebung, bedingte Abfrage-Tags, die von der aktuellen Seite erfüllt werden, und vieles mehr.

Query Monitor				
Page generation time	Peak memory usage	Database query time	Database queries	Object cache
0.0321 0.0% of 6s 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

Das Plugin wurde von John Blackburn entwickelt, einem WordPress Core Committer, der derzeit Entwickler bei Human Made ist und zuvor bei WordPress.com beschäftigt war – mit anderen Worten, jemand, der WordPress tiefgreifend kennt. Query Monitor wurde 2013 in das WordPress Plugin-Verzeichnis aufgenommen und verfügt derzeit über mehr als 10.000 aktive Installationen – eine beeindruckende Zahl für ein Entwicklungs-Plugin. Die Benutzerbewertung des Plugins mit fünf von fünf Sternen bestätigt seine Beliebtheit bei Entwicklern.

Nutze Staging-Umgebungen ohne die Produktions-Umgebung zu berühren

Wir wissen nicht, was wir ohne Staging-Umgebungen tun würden. Diese können von unschätzbarem Wert sein, wenn es um die Behebung von Performance-Problemen geht. Glücklicherweise stellt Kinsta Staging-Umgebungen mit nur einem Klick bereit. Wenn dein WordPress Host keine Staging-Umgebungen anbietet, kannst du auch ein Plugin wie WP Staging verwenden, obwohl das nicht so einfach ist.

The screenshot shows the Kinsta dashboard for 'Brian's Company'. The left sidebar contains navigation links: Dashboard, Sites, Migrations, Kinsta DNS, Analytics, Billing, Users, Activity Log, and User Guide. The main content area is titled 'editwp Environment' and includes a sub-header 'STAGING Environment' with a note: 'For testing and development, changes will not be visible on your live site.' Below this, it states 'You have no staging environment' and provides an explanation: 'Staging environments give you a playground to test new plugins, code modifications and other changes. When you create a staging site it will be a copy of your live site. You can then perform and test modifications. Once you're ready you can send all changes to your live site with the push of a button.' A prominent blue button labeled 'Create A Staging Environment' is visible. On the right, there is an illustration of a person at a laptop with a 'Change Environment' dropdown menu showing 'Live Environment' and 'Staging Environment' options. Blue arrows point from the text to the button and the dropdown menu.

Nachdem du eine Staging-Seite eingerichtet hast, kannst du als erstes alle deine Plugins deaktivieren. Da es sich hierbei um eine Kopie deiner Live-Seite handelt, musst du dir keine Sorgen machen, dass du etwas kaputt machst. Es ist bei weitem eine der einfachsten Möglichkeiten, Probleme einzugrenzen. Gehe einfach zu Plugins, wähle alle aus und klicke in den Bulk Actions auf „Deaktivieren“.

Dashboard

Kinsta Cache

Posts

WP Show Posts

Media

Forms

Pages

Comments **350**

W4 Post List

StCR

Coupons

TablePress

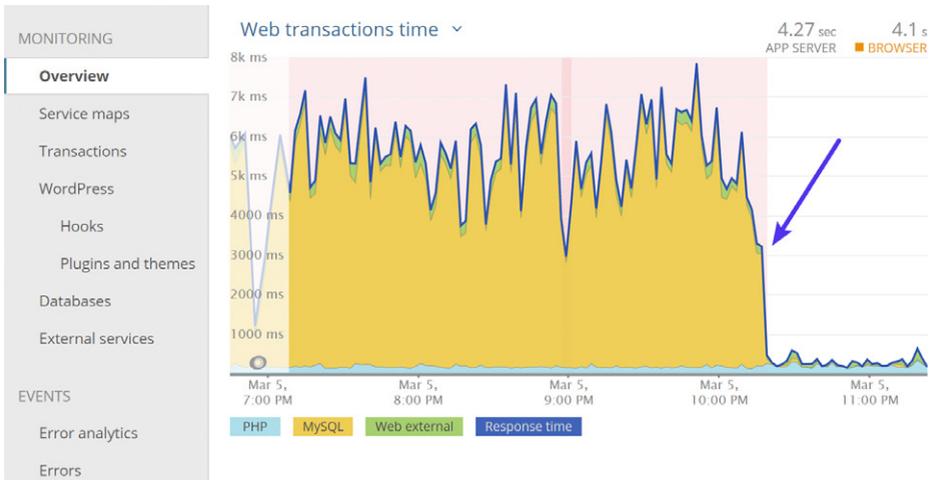
Plugins [Add New](#)

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

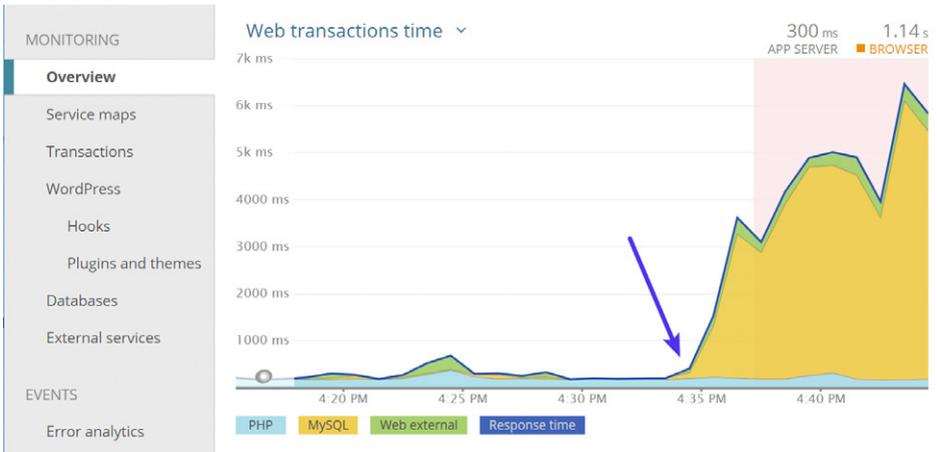
Bulk Actions [Apply](#)

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

Anschließend kannst du die Antwortzeiten im New Relic oder Query Monitor überwachen und sehen, was passiert. In diesem Beispiel unten fielen die Antwortzeiten sofort wieder auf den Normalzustand auf der Webseite zurück, so dass wir wussten, dass eines der Plugins die Ursache des Problems war. Du kannst sie dann nacheinander wieder aktivieren und den gleichen Prozess wiederholen, bis du den Übeltäter gefunden hast.



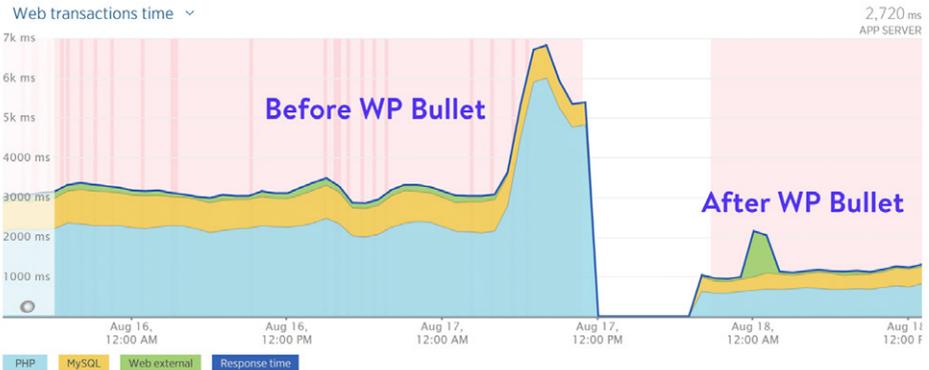
Hier ist ein Beispiel dafür, was passiert ist, als wir das Plugin aktiviert haben, das das Problem verursacht hat. Die Ladezeiten (Web-Transaktionszeiten) stiegen sofort wieder an.



Was ist zu tun, nachdem du das Plugin gefunden hast, das die Verzögerung verursacht?
Hier ist, was wir Dir raten:

1. Aktualisiere deine Plugins und Themes auf die neueste Version, wenn du es nicht schon getan hast.
2. Wende dich an den Entwickler des Plugins oder Theme und bitte ihn um Hilfe.
3. Finde ein alternatives Plugin, das die gleiche Funktion bietet.
4. Vielleicht verursacht deine PHP-Version ein Problem. Ändere dein PHP-Engine auf eine niedrigere Version und sieh nach, ob das Plugin oder das Theme dann funktioniert.

Du kannst auch einen WordPress-Entwickler beauftragen, um das Problem zu beheben. Wenn es um Geschwindigkeitsfragen geht, müssen wir Mike Andreason von WP Bullet an dieser Stelle kurz loben. Er ist ein hauptberuflicher Codeable Entwickler, der sich auf Performance-Optimierung spezialisiert hat und vielen Kunden hier bei Kinsta mit komplizierten Installationen geholfen hat, ihre Seite deutlich zu verbessern.



Überprüfe deine Fehlerprotokolle

Das Überprüfen von Fehlerprotokollen macht nie Spaß, kann aber viel über Performance-Probleme mit WordPress-Plugins aussagen. Als Kinsta-Kunde kannst du deine Fehlerprotokolle, Cache-Protokolle und Zugriffsprotokolle direkt im MyKinsta-Dashboard einsehen.

Kinsta
Brian's Company

editwp **LIVE Environment**
Changes made here, via SFTP or the database will be visible, please take care.

Change Environment

Info Log Viewer **LIVE**

Domains

Backups error.log ... Search

Tools

Redirects

WP Plugins

IP Deny

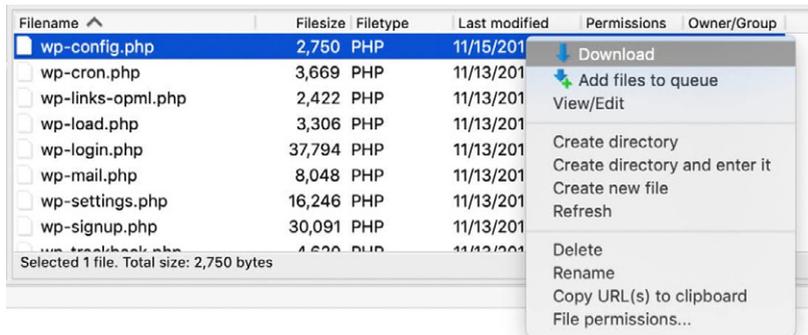
CDN

Logs

2018/11/16 18:08:15 [error] 607#607: *686 FastCGI sent in stderr: "PHP message: PHP Warning: trim() expects parameter"

Dashboard
Sites
Migrations
Kinsta DNS
Analytics
Billing
Users
Activity Log
User Guide

Du kannst Fehlerprotokolle auch aktivieren, indem du einen Code zu deiner `wp-config.php`-Datei hinzufügst. Zuerst wirst du dich mit deiner Webseite über SFTP verbinden wollen. Lade dann deine `wp-config.php` herunter, damit du sie bearbeiten kannst. Hinweis: Mach immer zuerst ein Backup dieser Datei!



Finde die Zeile `/* That's all, stop editing! Happy blogging. */` und füge unmittelbar davor folgendes hinzu (wie unten gezeigt):

```
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');
```

Wenn der obige Code bereits in deiner `wp-config.php`-Datei existiert, aber auf „false“ gesetzt ist, ändere ihn einfach auf „true“. Dadurch wird der Debug-Modus aktiviert. Hinweis: Du wirst auch Warnungen oder Fehler in deinem WordPress-Administrator sehen, falls diese existieren.

Du kannst dann das Debug-Protokoll aktivieren, um alle Fehler in eine Datei zu senden, indem du den folgenden Code direkt nach der Zeile `WP_DEBUG` hinzufügst (wie unten gezeigt):

```
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');

```

Speichere deine Änderungen und lade sie erneut auf deinen Server hoch. Die Fehler werden dann in der Datei `debug.log` in deinem Ordner `/wp-content/` protokolliert. Wenn du diese Datei aus irgendeinem Grund nicht siehst, kannst du jederzeit eine erstellen.

Die MyKinsta Analytics verwenden

Als Kinsta-Kunde kannst du die Performance-Einblicke nutzen, die wir in unser MyKinsta Analytics-Tool integriert haben.

Im Abschnitt zur Leistungsüberwachung kannst du deine durchschnittliche PHP + MySQL-Antwortzeit, den PHP-Durchsatz, die AJAX-Nutzung, die höchste durchschnittliche Upstream-Zeit und die höchste maximale Upstream-Zeit anzeigen.

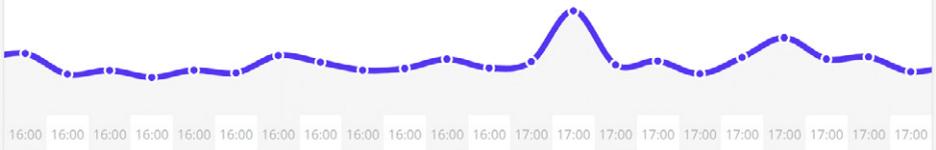
Durchschnittliche PHP + MySQL Antwortzeit

Wann immer du deine WordPress-Seite besuchst, werden PHP und MySQL verwendet, um die Daten, die du auf der Seite siehst, zu kompilieren und abzufragen. Dieses Diagramm zeigt dir die durchschnittliche Antwortzeit der PHP-Engine und der MySQL-Engine für jede nicht zwischengespeicherte dynamische Anforderung. Die Resultate dieser Reaktionszeiten können dir helfen, Fehler zu beheben.

Average PHP + MySQL Response Time

all sites | Nov 26 - Nov 27

0.2



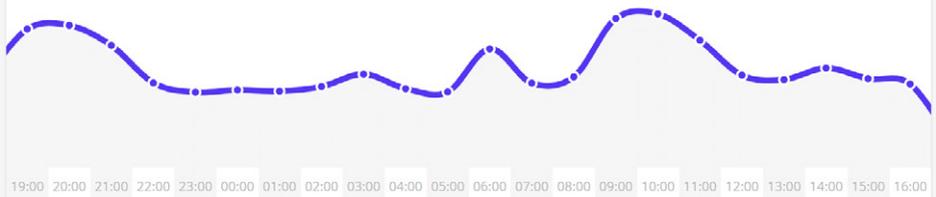
PHP Throughput

Der Throughput gibt die Anzahl der Transaktionen pro Sekunde an, die eine Anwendung verarbeiten kann, und in diesem Bericht bezieht er sich auf den PHP-Durchsatz von deiner WordPress-Seite. Mit anderen Worten, es zeigt dir, wie oft ein PHP-Asset angefordert wurde.

PHP Throughput

all sites | Nov 26 - Nov 27

50,033



Höchste durchschnittliche PHP + MySQL Antwortzeit

Diese Liste zeigt die durchschnittlich besten Antwortzeiten von PHP und MySQL. Diese Zahlen können einmalige Spitzenwerte sein, daher wird empfohlen, diese Liste mit „Top Maximum Upstream Time“ zu vergleichen.

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

Höchste maximale Upstream Zeit

Die Upstream-Zeit ist die Gesamtzeit, die NGINX (und die Upstream-Server) benötigen,

um eine Anfrage zu bearbeiten und eine Antwort zu senden. Die Zeit wird in Sekunden gemessen, mit einer Auflösung von Millisekunden. Erfahre mehr über die NGINX-Kennzahlen.

PATH	REQUESTS	TIME
/wp-cron.php?server_triggered_cronjob	93	180.00 s
/wp-admin/plugins.php?activate=true&plugin_status=all&paged=1&s=	2	2.60 s
/wp-admin/update.php?action=upload-plugin	1	2.19 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&plugin_status=all&paged=1&s&_wpnonce=70f7d2faca	1	1.47 s
/hello-world/	5	1.46 s
/wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=729c628974	1	1.07 s
/	83	0.93 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=70f7d2faca	1	0.91 s
/wp-admin/plugins.php	3	0.85 s
/wp-cron.php?doing_wp_cron=1542348999.81110280036926269531250	1	0.67 s

Deine Seite könnte gehackt werden

Wenn du Probleme hast, ein Performance-Problem aufzuspüren, kann es sehr gut sein, dass deine Webseite gehackt, mit Malware infiziert oder einem DDoS-Angriff ausgesetzt wurde. Das kann die Geschwindigkeit deiner Webseite und sogar die Reaktionsfähigkeit deines WordPress Admin Dashboards beeinträchtigen. In diesen Fällen empfehlen wir folgendes:

1. Implementiere einen Proxy-Server und WAF wie Cloudflare oder Sucuri.
2. Blockiere schlechte IP-Adressen über die oben genannten Dienste oder wenn du ein Kunde von Kinsta bist, kannst du auch IP-Adressen über unser MyKinsta-Dashboard blockieren.
3. Du kannst auch Geo-Blocking implementieren. Einige Länder schneiden sehr schlecht ab, wenn es um die Qualität des von ihnen erzeugten Traffic geht. Wenn du angegriffen wirst, musst du möglicherweise das ganze Land blockieren, entweder vorübergehend oder dauerhaft.

Fehlerbehebung mit Fehlercodes (HTTP Status Codes)

HTTP-Statuscodes sind wie eine kurze Notiz vom Webserver, die oben auf einer Webseite angehängt wird. Sie sind nicht Teil der Webseite. Stattdessen handelt es sich um eine Nachricht des Servers, der dich darüber informiert, wie es gelaufen ist, als die Anforderung, die Seite anzuzeigen, vom Server empfangen wurde. Diese können bei der Fehlersuche von unschätzbarem Wert sein!

Während es über 40 verschiedene Statuscodes gibt, führen wir unten die häufigsten an, mit denen WordPress-Nutzer zu kämpfen haben.

429: “Too many requests.” wird vom Server generiert, wenn der Benutzer zu viele Anfragen in einer bestimmten Zeitspanne gesendet hat (Ratenbegrenzung). Dies kann manchmal bei Bots oder Skripten auftreten, die versuchen, auf deine Webseite zuzugreifen.

In diesem Fall solltest du vielleicht versuchen, deine WordPress-Login-URL zu ändern.

429 Too Many Requests

nginx

500: „Es gab einen Fehler auf dem Server und die Anfrage konnte nicht abgeschlossen werden.“ Ein generischer Code, der einfach „interner Serverfehler“ bedeutet. Auf dem Server ist etwas schief gelaufen, und die angeforderte Ressource wurde nicht geliefert. Dieser Code wird typischerweise von Drittanbieter-Plugins, fehlerhaftem PHP oder sogar der Verbindung zur Datenbank erzeugt. In unseren Tutorials erfährst du, wie

du den Fehler beim Herstellen einer Datenbankverbindung beheben kannst und wie du einen internen 500-Serverfehler beheben kannst.

Error establishing a database connection

502: "Bad Gateway." Dieser Fehlercode bedeutet in der Regel, dass ein Server eine ungültige Antwort von einem anderen empfangen hat. Manchmal dauert eine Abfrage oder Anforderung zu lange, und so wird sie vom Server abgebrochen oder beendet und die Verbindung zu den Datenbanken bricht ab. In unserem ausführlichen Tutorial erfährst du, wie du den Fehler 502 Bad Gateway beheben kannst.

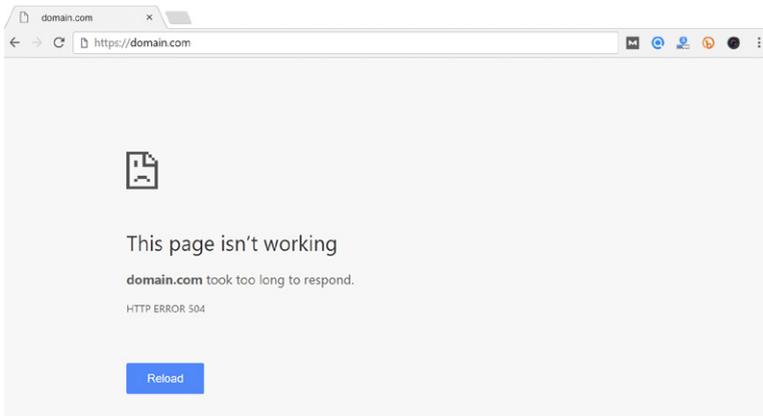


502 Bad Gateway

nginx

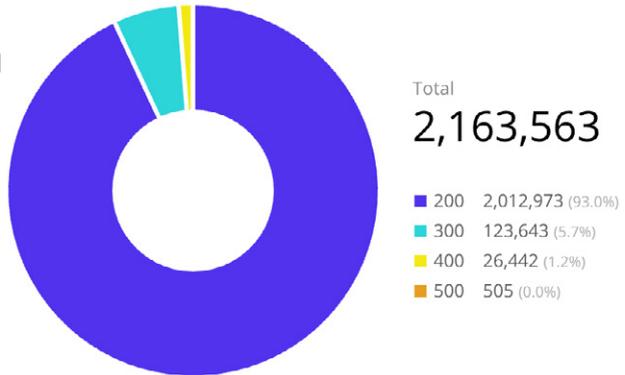
503: „Der Server ist im Moment nicht in der Lage, diese Anfrage zu bearbeiten.“ Die Anfrage kann zurzeit nicht bearbeitet werden. Dieser Code kann von einem überlasteten Server zurückgegeben werden, der keine zusätzlichen Anfragen bearbeiten kann.

504: „Der Server, der als Gateway fungiert, hat ein Timeout und wartet darauf, dass ein anderer Server antwortet.“ Der Code wird zurückgegeben, wenn zwei Server an der Verarbeitung einer Anforderung beteiligt sind, und der erste Server weiterhin auf die Antwort des zweiten wartet.



Du kannst dich auch in diese HTTP-Antwortcodes in unserem MyKinsta Analytics-Tool einarbeiten. Unser Antwortcode-Dashboard gibt dir einen Überblick über die Verteilung der HTTP-Statuscodes, die für die angeforderten Ressourcen bereitgestellt werden.

Antwortcode Aufschlüsselung



Im Bericht über die Antwortstatistiken siehst du die Gesamtzahl der Redirects, Fehler, Erfolgsrate und Fehlerquote. Jede WordPress-Seite hat typischerweise eine kleine Fehlerquote; das ist völlig normal.

Antwortstatistiken

513,037
Redirects

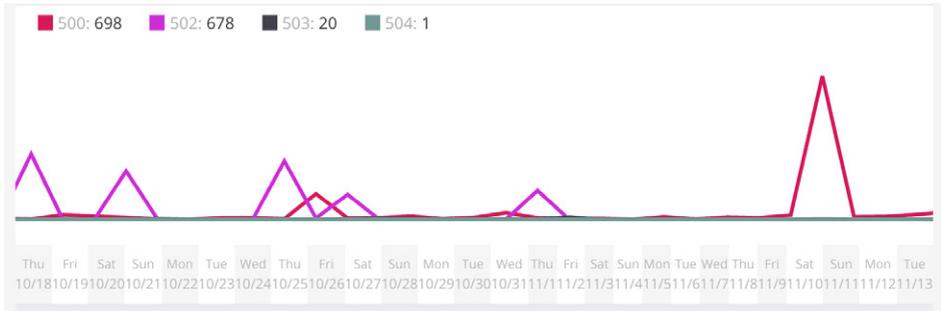
119,796
Errors

99.3%
Success Rate

0.7%
Error Ratio

Es gibt dann Ausfallberichte für jede Art von Fehlercode, wie z.B. 500 Fehler, 400 Fehler, Redirects etc.

500 Fehler Aufschlüsselung



KAPITEL 14:

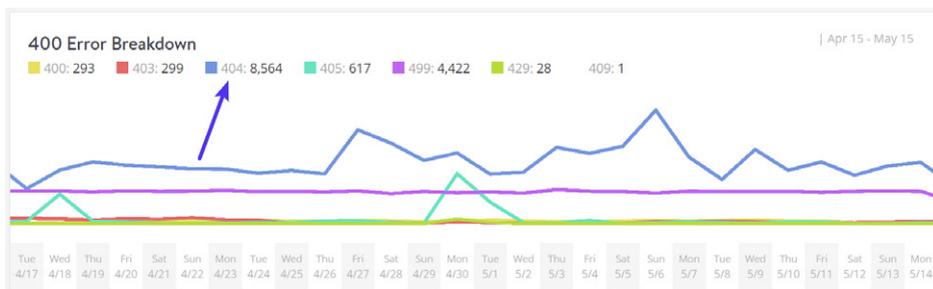
Empfehlungen zur Backend optimierung



Nun werden wir auf einige Möglichkeiten eingehen, wie du WordPress beschleunigen kannst, indem du das Backend optimierst. Backend beinhaltet typischerweise alles, was vollständig vom Server verwaltet wird, wie z.B. PHP, HTTP-Cache Header, GZIP-Kompression etc.

Erstelle eine einfache 404 Seite

Wir haben aus erster Hand gesehen, dass hochdynamische Webseiten typischerweise viele **404 Fehler** erzeugen. Deine Webseite generiert eventuell mehr, als du denkst! Unser Analyticstool in MyKinsta kann dir helfen, den genauen Betrag zu ermitteln (siehe unten).



Der Grund für diese Fehler ist, dass viele 404 Seiten **sehr ressourcenintensiv** sind. Für eine hochdynamische WordPress-Webseite solltest du eine übermäßige 404-Seite vermeiden. Erstelle eine einfache 404-Vorlage, die eine weitere Abfrage der Datenbank nach Möglichkeit vermeidet. Und natürlich solltest du einige Zeit damit verbringen, die 404-Fehler zu beheben, da das nicht nur ressourcenintensiv ist, sondern auch einfach schlecht für die Benutzerfreundlichkeit.

Erhöhe die Anzahl der PHP Worker

PHP-Worker sind vielleicht ein Begriff, von dem du noch nie gehört hast, aber damit bearbeiten viele Hosts, einschließlich Kinsta, Begrenzungsanfragen (anstatt dich durch CPU oder RAM einzuschränken, was normalerweise bei Shared Hosting-Providern der Fall ist).

PHP-Worker bestimmen, **wie viele gleichzeitige Anfragen deine Webseite zu einem bestimmten Zeitpunkt bearbeiten kann**. Einfach ausgedrückt, wird jede nicht gecachte Anfrage an deine Webseite von einem PHP-Worker bearbeitet. Wenn man zum Beispiel 4 Anfragen hat, die genau zur gleichen Zeit auf seine Webseite kommen und die Webseite 2 PHP-Worker hat, werden zwei dieser Anfragen bearbeitet, während die anderen beiden in der Warteschlange warten müssen, bis die ersten beiden verarbeitet wurden.

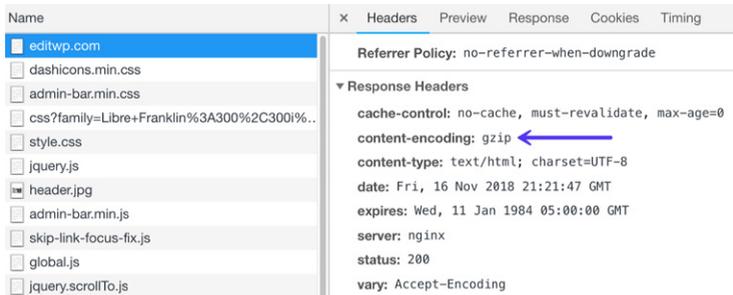
Erinnern wir uns, dass eines der größten Probleme mit WordPress-Mitgliederseiten all diese nicht gecachten Anfragen sind. Aus diesem Grund werden PHP-Worker sehr wichtig, da sie für jede dieser PHP-Requests zuständig sind. Daher werden diese Webseiten in der Regel zusätzliche PHP-Worker benötigen, um sicherzustellen, dass jede Anfrage ohne Verzögerung bearbeitet und erfolgreich abgeschlossen wird.

Was passiert, wenn du deine PHP-Worker kontinuierlich ausreizt? Im Grunde genommen beginnt die Warteschlange, ältere Anfragen zu entfernen, was zu 500 Fehlern auf deiner Webseite führen kann. Jedes der Hosting-Angebote von Kinsta beinhaltet eine vordefinierte Anzahl von PHP-Workern. Wenn du Schwierigkeiten hast, einzuschätzen, was deine Seite braucht, kannst du jederzeit mit unserem Verkaufs- oder Support-Team chatten.

Verwende die GZIP Komprimierung

GZIP ist ein Dateiformat und eine Softwareanwendung zur Komprimierung und Dekomprimierung von Dateien. Die GZIP-Komprimierung ist serverseitig aktiviert und ermöglicht eine weitere Reduzierung der Größe deiner HTML-, Stylesheets und JavaScript-Dateien.

Wenn ein Webbrowser eine Webseite besucht, wird überprüft, ob auf dem Webserver GZIP aktiviert ist, indem er prüft, ob die `content-encoding: gzip` HTTP-Header existieren. Wenn der Header erkannt wird, dient er zur Ausgabe der komprimierten und kleineren Dateien. Wenn nicht, werden die unkomprimierten Dateien bereitgestellt. Wenn du GZIP nicht aktiviert hast, wirst du höchstwahrscheinlich Warnungen und Fehler in Geschwindigkeitstest-Tools wie Google PageSpeed Insights und GTmetrix sehen.

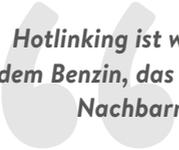


Die Aktivierung der GZIP-Komprimierung kann dazu beitragen, die Größe deiner Webseite zu reduzieren, was die Zeit für das Herunterladen der Ressource erheblich verkürzt, den Datenverbrauch für den Besucher reduziert und die Zeit bis zur ersten Darstellung deiner Seiten verringert. Das ist heute bei den meisten Hosting-Providern ziemlich üblich, aber an dieser Stelle überrascht uns nichts mehr.

Den Hotlink Schutz aktivieren

Das Konzept des Hotlinking ist ziemlich einfach. Du findest irgendwo im Internet ein Bild und verwendest die URL des Bildes direkt auf deiner Seite. Dieses Bild wird auf deiner Webseite angezeigt, aber es wird von dem ursprünglichen Standort aus bedient. Dies ist sehr praktisch für den Hotlinker, aber es ist eigentlich Diebstahl, da er die

Ressourcen der gehotlinkeden Seite nutzt. Es ist, als würden wir in unser Auto steigen und mit Benzin wegfahren, das wir aus dem Auto unseres Nachbarn abgezapft haben.



Hotlinking ist wie das Autofahren mit dem Benzin, das du aus dem Auto deines Nachbarn gestohlen hast.

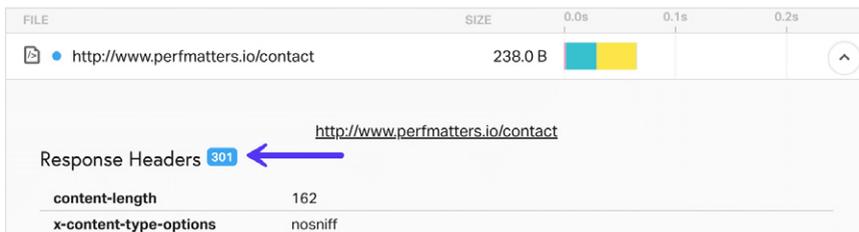
Hotlinking kann eine enorme Belastung für die Ressourcen des Zielsevers darstellen.

Man stelle sich vor, Du befindest dich auf einem freigegebenen WordPress Host und Spiegel Online verlinkt plötzlich auf deine Bilder. Du könntest von ein paar hundert Anfragen pro Stunde auf deiner Seite zu ein paar hunderttausend übergehen. Dies kann sogar zu einer Sperrung deines Hosting-Accounts führen. Das ist ein Grund, nicht nur einen leistungsstarken Host zu verwenden (der mit solchen Problemen umgehen kann), sondern auch den Hotlink-Schutz zu aktivieren, damit solch ein Problem vermieden werden kann.

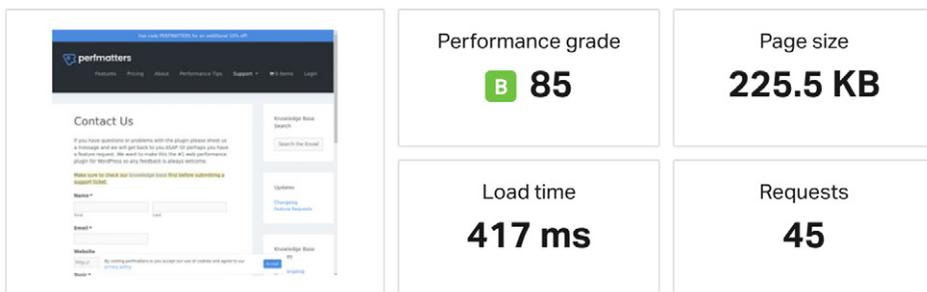
Minimiere Weiterleitungen und füge sie auf Serverebene hinzu

Zu viele Weiterleitungen sind immer etwas, auf das man achten muss. Einfache Weiterleitungen wie eine einzelne 301 Weiterleitung, HTTP zu HTTPS oder www zu non-www (oder umgekehrt) sind in Ordnung und oft werden diese in bestimmten Bereichen deiner Webseite benötigt. Allerdings beeinflusst jede davon die Geschwindigkeit Deiner Webseite. Und wenn du anfängst, Weiterleitungen übereinander zu stapeln, ist es wichtig zu erkennen, wie sie sich auf deine Webseite auswirken. Das gilt für Seiten- und Postverlinkungen, Bildverlinkungen, einfach alles.

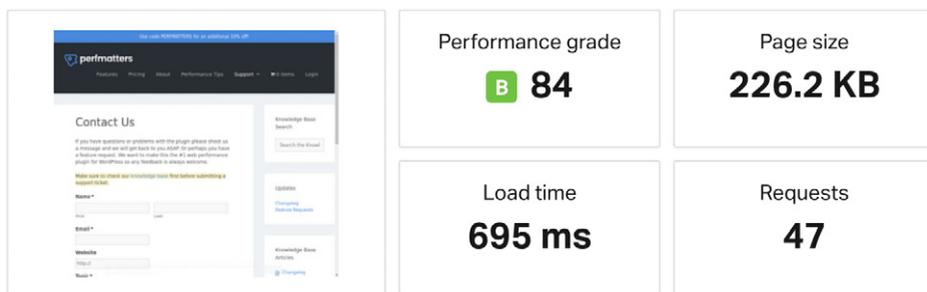
Eine Weiterleitung erzeugt ein 301 oder 302 auf dem Status des Antwortheaders.



Wie stark wirken sich Weiterleitungen auf deine Webseite aus? Lass uns einen kleinen Test durchführen. Zuerst führen wir einen Geschwindigkeitstest auf unserer Kontaktseite durch: <https://perfmatters.io/contact/>. Wie du unten sehen kannst, erhalten wir eine Gesamtladezeit von **417 ms**.

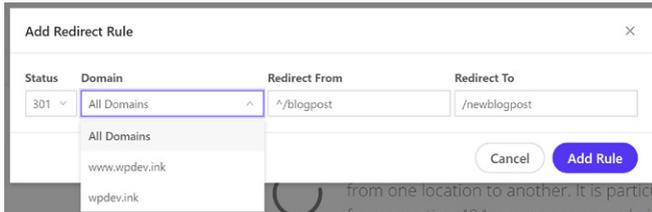


Wir ändern dann die URL leicht und führen einen weiteren Geschwindigkeitstest durch, um die Auswirkungen mehrerer Weiterleitungen zu sehen. <http://www.perfmatters.io/contact>. Wie du sehen kannst, benötigt die gleiche Seite nun **695 ms** zum Laden. Das ist eine Steigerung von 66%. Huch!

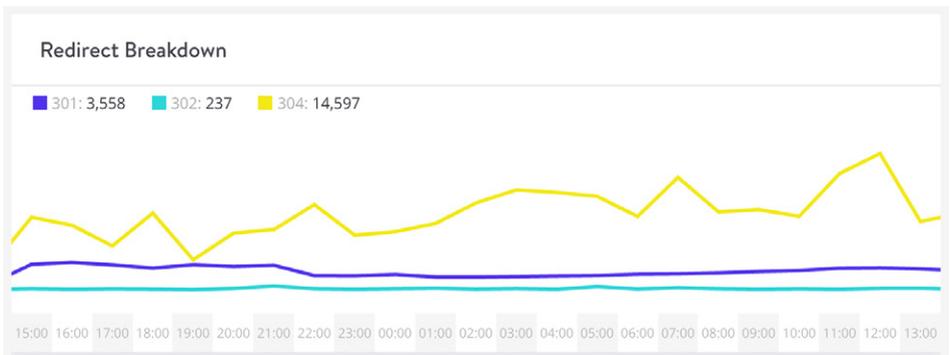


Die Verwendung kostenloser WordPress-Plugins zur Implementierung von Weiterleitungen kann manchmal zu Performance-Problemen führen, da die meisten von ihnen die `wp_redirect`-Funktion verwenden, die zusätzliche Codeausführung und

Ressourcen erfordert. Einige von ihnen fügen auch automatisch geladene Daten zu deiner `wp_options`-Tabelle hinzu, was deine Datenbankgröße nur weiter aufbläst. Stattdessen sollten sie auf Serverebene hinzugefügt werden. Wir ermöglichen es dir, das bei MyKinsta mit unserem Redirect Rules Tool zu tun.



Du kannst auch eine vollständige Aufschlüsselung der Anzahl der Weiterleitungen auf deinen Seiten in unserem MyKinsta Analysetool sehen. Schau dir die Gesamtzahl der 301er, 302er und 304er an.



Lasse nicht zu, dass Cronjobs außer Kontrolle geraten

CRON-Jobs (WP-Cron) werden genutzt, um wiederholende Aufgaben deiner WordPress-Seite zeitlich zu planen. Im Laufe der Zeit können diese jedoch außer Kontrolle geraten und Performance-Probleme verursachen. Du kannst das kostenlose WP Control Plugin verwenden, um einen Überblick über alle Cron Jobs zu erhalten, die auf deiner Seite stattfinden.

Wir haben auch Performance-Probleme mit WP Cron, dem in WordPress integrierten

Cron Programm festgestellt. Wenn eine Webseite nicht genügend PHP-Worker hat und eine Anfrage beginnt, startet WordPress Cron, aber der Cron muss nun auf den Worker warten und dreht daher solange Däumchen. Ein besserer Ansatz ist es, WP-Cron zu deaktivieren und stattdessen den System Cron zu verwenden. Dies wird sogar im offiziellen Plugin-Handbuch empfohlen.

Um WP-Cron zu deaktivieren, füge die folgende Zeile zu deiner Datei `wp-config.php` hinzu, Sorge nur dafür, dass dies vor der Zeile „That’s all, step editing! Happy blogging.“ geschieht. Hinweis: Dadurch wird verhindert, dass es beim Laden der Seite läuft, nicht aber, wenn du es direkt über `wp-cron.php` aufrufst.

```
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

Dann musst du `wp-cron.php` von deinem Server aus koordinieren. Wenn du ein Kinstakunde bist, sind die System-Crons bereits aktiviert und werden standardmäßig alle 15 Minuten ausgeführt. Du kannst die Frequenz auch erhöhen lassen, indem du dich an unser Support-Team wendest. Wenn du mit SSH vertraut bist, kannst du Server-Crons von der Befehlszeile aus verwalten.

Füge Cache-Control und Expires Headers hinzu (lege die Zeit fest wie lange etwas gecached wird)

Jedes Skript auf deiner WordPress-Seite muss mit einem HTTP-Cache-Header versehen sein (oder sollte es). Das **bestimmt, wann der Cache der Datei abläuft**. Um das zu beheben, stelle sicher, dass dein WordPress Host über die richtigen `cache-control` headers und `expires` für Header verfügt. Wenn du es nicht tust, wirst du höchstwahrscheinlich Warnungen in Geschwindigkeitstests sehen, die darauf hindeuten, dass du `expires` Header hinzufügen musst oder das Browser-Caching nutzen solltest.

Während der `cache-control` Header das clientseitige Caching aktiviert und das maximale Zeitlimit einer Ressource festlegt, wird die `expires` Header verwendet, um einen bestimmten Zeitpunkt anzugeben, an dem die Ressource nicht mehr gültig ist. Obwohl beide Header zusammen verwendet werden können, muss man nicht unbedingt beide hinzufügen. `cache-control` ist neuer und normalerweise die empfohlene Methode.

Kinsta fügt automatisch HTTP-Cache-Header auf allen Serveranfragen hinzu, und wenn du ein CDN verwendest, wird Kinsta höchstwahrscheinlich auch diese Header für dich hinzufügen.

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

Wenn deinem Server diese Header fehlen, kannst du sie manuell hinzufügen.

Einen Cache-Control Header in Nginx hinzufügen

Du kannst `cache-control` Headers in Nginx hinzufügen, indem du folgendes zum Serverstandort oder Block deiner Serverkonfiguration hinzufügst.

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

Einen Expires Header in Nginx hinzufügen

In Nginx kannst du `expires` hinzufügen, indem du folgendes zu deinem Serverblock hinzufügst. In diesem Beispiel wird erläutert, wie man je nach Dateityp unterschiedliche Ablaufzeiten angibt.

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

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

Einen Cache-Control Header in Apache hinzufügen

Du kannst `cache-control`-Header in Apache hinzufügen, indem du Folgendes zu deiner `.htaccess`-Datei hinzufügst. Codeausschnitte können oben oder unten in der Datei hinzugefügt werden (vor `# BEGIN WordPress` oder nach `# END WordPress`).

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

Einen Expires Header in Apache hinzufügen

Du kannst `expires` Header in Apache hinzufügen, indem Du Folgendes zu deiner `.htaccess`-Datei hinzufügst.

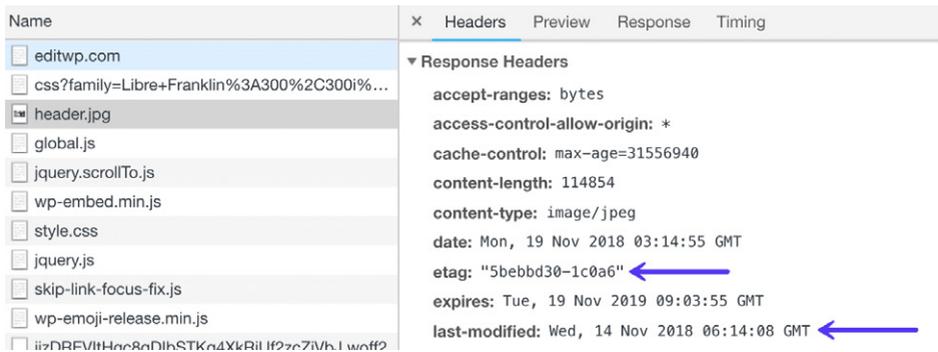
```
## EXPIRES HEADER CACHING ##
<IfModule mod_expires.c>
ExpiresActive On
ExpiresByType image/jpg "access 1 year"
ExpiresByType image/jpeg "access 1 year"
ExpiresByType image/gif "access 1 year"
ExpiresByType image/png "access 1 year"
ExpiresByType image/svg "access 1 year"
ExpiresByType text/css "access 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 ##
```

Es ist auch wichtig zu beachten, dass **du nur HTTP-Cache-Header zu Ressourcen auf deinem Server hinzufügen kannst**. Wenn du eine Warnung darüber erhältst, dass du vielleicht das Browser-Caching bei einer Anfrage eines Drittanbieters nutzen musst, gibt es nichts, was du tun kannst, da du keine Anfrage über dessen Server stellen kannst. Zu den häufigsten Verursachern gehören das Google Analytics-Skript und Marketing-Pixel wie Facebook und Twitter.

Wenn du versuchst, das mit dem Google Analytics-Skript zu beheben, kannst du es lokal oder auf deinem CDN (obwohl das nicht offiziell unterstützt wird) mit einem Plugin wie Perfmatters oder WP Rocket hosten.

Die Last-Modified und ETag Header hinzufügen (Cache validieren)

Als nächstes haben wir noch zwei weitere Header, `last-modified` und `etag`. `Cache-control` und `expires` Headers helfen dem Browser festzustellen, ob sich die Datei seit der letzten Anforderung geändert hat. Oder besser gesagt, sie validieren den Cache. Wenn diese nicht richtig eingestellt sind, wird möglicherweise eine Warnung angezeigt, dass du "einen Cache-Validierer angeben" musst.



Diese Header sollten in jeder Serverantwort enthalten sein, da sie **die Länge des Cache validieren und festlegen**. Wenn die Header nicht gefunden werden, wird jedes Mal eine neue Anforderung für die Ressource generiert, was die Belastung des Servers erhöht. Die Verwendung von Cache-Header stellt sicher, dass nachfolgende Anfragen nicht vom Server geladen werden müssen, was Bandbreite spart und die Performance für den Benutzer verbessert.

Kinsta fügt die obigen Header automatisch allen Serveranfragen hinzu, und wenn du ein CDN verwendest, wird Kinsta höchstwahrscheinlich auch diese Header für dich hinzufügen. Genau wie bei `cache-control` und `expires` kann man diese HTTP-Header nicht manuell auf externe Ressourcen verlegen.

Last-Modified Header

Der „**last modified**“ Header wird in der Regel automatisch vom Server gesendet. **Das**

ist ein Header, den du normalerweise nicht manuell hinzufügen musst. Er wird gesendet, um zu sehen, ob die Datei im Cache des Browsers seit der letzten Anforderung geändert wurde. Man kann sich die Header-Anfrage in Pingdom ansehen oder Chrome DevTools verwenden, um den Wert der „last modified“ Kopfzeile zu sehen.

ETag Header

Der „**ETag**“-Header ist dem „last modified“ Header sehr ähnlich. Er wird auch verwendet, um den Cache einer Datei zu validieren. Wenn Du Apache 2.4 oder höher verwendest, wird der „ETag“-Header bereits automatisch über die FileETag-Direktive hinzugefügt. Und was NGINX betrifft, so ist der „ETag“-Header seit 2016 standardmäßig aktiviert.

Du kannst den ETag header manuell in Nginx mit dem folgenden Code aktivieren.

```
etag on
```

Einen Vary: Accept-Encoding Header hinzufügen

Der `vary: Accept-Encoding` Header sollte bei jeder Serverantwort enthalten sein, da er dem Browser sagt, ob der Client mit komprimierten Versionen des Inhalts umgehen kann oder nicht. Wenn das nicht richtig eingestellt ist, wird möglicherweise eine Warnung angezeigt, dass du einen „Vary: Accept-Encoding Header spezifizieren“ sollst.

Angenommen, du hast einen alten Browser ohne gzip-Komprimierung und einen modernen Browser mit gzip-Komprimierung. Wenn du den `vary: Accept-Encoding` Header nicht verwendest, könnte dein Webserver oder CDN die unkomprimierte Version versehentlich an den modernen Browser liefern, was wiederum die Performance deiner WordPress-Seite beeinträchtigt. Durch die Verwendung des Headers kannst du sicherstellen, dass dein Webserver und/oder CDN die entsprechende Version liefert.

Name	× Headers	Preview	Response	Timing
editwp.com	Response Headers <ul style="list-style-type: none"> content-encoding: gzip content-type: text/html; charset=UTF-8 date: Mon, 19 Nov 2018 03:14:57 GMT server: nginx status: 200 vary: Accept-Encoding ← x-content-type-options: nosniff x-kinsta-cache: HIT 			
css?family=Libre+Franklin%3A300%2C300i%...				
header.jpg				
global.js				
jquery.scrollTo.js				
wp-embed.min.js				
style.css				
jquery.js				
skip-link-focus-fix.js				

Kinsta fügt die obigen Header automatisch auf allen Serveranfragen hinzu, und wenn du ein CDN verwendest, wird Kinsta höchstwahrscheinlich auch diese Header für dich hinzufügen. Genau wie bei den anderen Cache-Headern, die wir oben besprochen haben, kannst du diesen Header nicht manuell auf externe Ressourcen verlegen.

Einen Vary: Accept-Encoding Header in Apache hinzufügen

Du kannst den `Vary: Accept-Encoding` Header in Apache hinzufügen, indem du das Folgende zu deiner `.htaccess`-Datei hinzufügst.

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

Einen Vary: Accept-Encoding Header in Nginx hinzufügen

Du kannst `Vary: Accept-Encoding` Header in Nginx hinzufügen, indem du den folgenden Code zu deiner Konfigurationsdatei hinzufügst. Alle Nginx-Konfigurationsdateien befinden sich im Verzeichnis `/etc/nginx/`. Die primäre Konfigurationsdatei ist

```
/etc/nginx/nginx.conf.
```

```
gzip_vary on
```

Das WordPress Memory Limit in der wp-config.php Datei ändern

Wie im WordPress Codex angegeben, erlaubt dir die Option `WP_MEMORY_LIMIT` mit WordPress Version 2.5 die maximale Speichermenge anzugeben, die von PHP verbraucht werden kann. Diese Einstellung kann erforderlich sein, wenn du eine Meldung wie „Zulässige Speichergröße von xxxxxxxx Bytes erschöpft“ erhältst.

Standardmäßig wird WordPress versuchen den für PHP zugewiesenen Speicherplatz auf 40 MB für eine einzelne Seite und 64 MB für mehrere Seiten zu erhöhen.

Du definierst die Speichergrenzen in der Datei `./wp-includes/default-constants.php`, auf den Zeilen 32 – 44 (Quelle).

Du hast dann auch PHP `memory_limit` auf dem Server von deinem Hosting-Provider. Das sind zwei verschiedene Dinge. Bei Kinsta haben wir das Standard `memory_limit` auf 256M gesetzt. Wenn du auf den Fehler Speichergröße erschöpft stößt, kannst du versuchen, die PHP-Speichergrenze in WordPress zu erhöhen.

Füge das Folgende zu deiner `wp-config.php`-Datei, hinzu, unbedingt vor der Zeile „That’s all, step editing! Happy blogging.“

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

Jan Reilink hat auch einen sehr guten Blog-Post verfasst, der das Problem der WordPress-Speicherbegrenzung genauer beschreibt. Er gibt auch eine Alternative zu dem Code, den du verwenden kannst. Anstatt den Betrag manuell einzustellen, kannst du ihn auf den PHP `memory_limit`-Wert setzen.

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

KAPITEL 15:

Tipps für die Front-End Optimierung und für externe Services



Nun werden wir auf einige Möglichkeiten eingehen, wie du WordPress beschleunigen kannst, indem du das Frontend optimierst. Frontend beinhaltet typischerweise alles, was vollständig vom client-seitigen Browser verarbeitet wird, wie CSS, JavaScript, Bilder, etc. Dazu gehört auch die Analyse externer Dienste, die du auf deiner Webseite laden kannst und wie sie sich auf deine Gesamtladezeit auswirken.

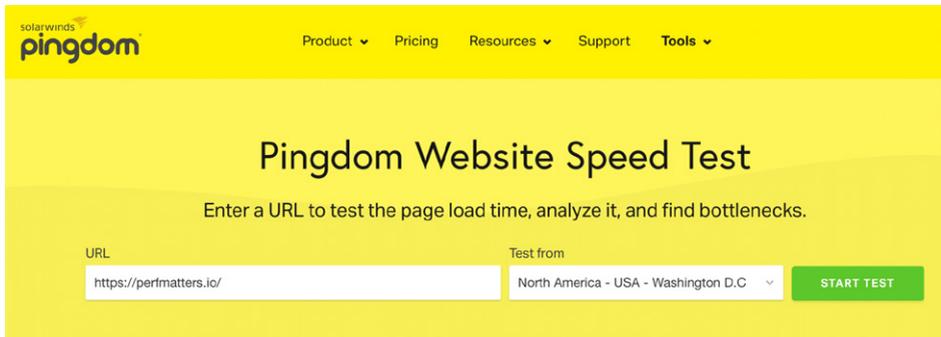
Zwei der wichtigsten Ziele, die du verfolgen solltest, wenn es um die Frontendoptimierung geht, sind:

- **Reduziere die Größe deiner gesamten Webseite.** Die Größe deines CSS, JavaScript, Bilder ist entscheidend. Eine 4 MB Webseite wird typischerweise viel langsamer geladen als eine 1 MB Webseite. Paul Calvano hat einen interessanten Artikel über die Auswirkungen der Größe einer Seite auf die Ladezeit verfasst und wie wichtig es ist sicherzustellen, dass es nicht das Einzige ist, was du verfolgst, da es manchmal irreführend sein kann.
- **Reduzierung von HTTP-Anfragen und externen Diensten.** Mit HTTP/2 können nun mehrere Anfragen und Antworten gleichzeitig über eine einzige TCP-Verbindung gesendet werden. Obwohl das für die Leistung fantastisch ist, kann die Reduzierung von HTTP-Anfragen dennoch helfen, deine WordPress-Seite zu beschleunigen. Dazu gehört auch die Reduzierung der Gesamtzahl

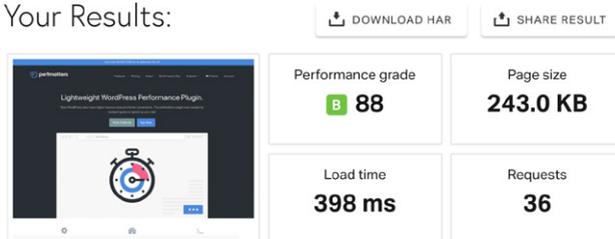
der externen Anfragen und Dienstleistungen. Denn diese fügen zusätzliche Verzögerungen wie DNS-Lookups, TLS-Verbindungen und Netzwerklatenz hinzu.

Teste die Ladezeit deiner WordPress Seite um einen Basiswert zu erhalten

Wenn es darum geht, das Frontend deiner Webseite zu optimieren, ist es immer gut, mit einem Basiswert zu beginnen. Das bedeutet in der Regel, dass du einen Geschwindigkeitstest durchführen musst. Es gibt eine Vielzahl von Möglichkeiten, wie du das tun kannst, schau dir unsere Liste von 15 sehr guten Webseiten-Geschwindigkeitstest-Tools an.



Your Results:



DOWNLOAD HAR	SHARE RESULT
Performance grade B 88	Page size 243.0 KB
Load time 398 ms	Requests 36

Hier sind ein paar Dinge, die man bei der Geschwindigkeitsprüfung beachten sollte:

1. Wähle ein Tool und bleibe dann dabei

Wir sind große Fans von Pingdom, GTmetrix, WebPageTest, PageSpeed Insights und Chrome DevTools. Es spielt jedoch keine Rolle, welches Tool du verwendest, es kommt darauf an, dass Du konsequent mit einem fährst. Sie alle haben verschiedene Möglichkeiten, die Geschwindigkeit zu messen und zu quantifizieren, also wähle ein Tool und führe damit alle deine Tests und Optimierungen durch. Sogar Google setzt darauf.



Es spielt viel weniger eine Rolle welches Tool du nutzt, viel wichtiger ist bei ein und demselben zu bleiben.

2. Fahre dich nicht fest nur weil du einen perfekten Score haben möchtest

Viele der Tools wie Google PageSpeed Insights haben alle eine Art Geschwindigkeits- oder Performancebewertung. Es ist wichtig, sich daran zu erinnern, dass das Ergebnis nicht immer so wichtig ist wie die Geschwindigkeit und die wahrgenommene Performance deiner Webseite durch den Benutzer. Das Ergebnis ist da, um dir eine Einschätzung zu liefern. Aber besessen perfekte 100 von 100 Punkten erreichen zu wollen, kann in manchen Fällen eine Zeitverschwendung sein. Und größere Webseiten mit vielen externen Skripten und Anzeigen werden nie eine perfekte Bewertung erhalten, was in Ordnung ist.

3. Der Standort spielt eine Rolle

Der Ort, den du wählst, ist nicht irrelevant bei der Geschwindigkeitsprüfung. Wie wir bereits in einem früheren Abschnitt besprochen haben, liegt der Grund darin, dass das alles mit dem von dir gewählten Serverstandort zusammenhängt. TTFB, Netzwerklatenz, all das spielt eine Rolle. Teste deine Webseite also sowohl von einem Standort, der sich

in der Nähe deines Rechenzentrums befindet, als auch von einem Standort, der weit entfernt ist. Das wird dir auch helfen einzuschätzen, wie stark sich ein CDN auf deine WordPress-Seite auswirken kann.

4. Mache mehrere Tests wegen dem Caching

Wie wir bereits im Abschnitt über das Caching erläutert haben, wird der Cache, wenn er kürzlich gelöscht wurde oder auf deinem WordPress Host oder CDN abgelaufen ist, ein „MISS“ im HTTP-Header anzeigen. Das bedeutet, dass deine Webseite oder dein Asset nicht aus dem Cache bedient wird.

The screenshot shows the 'Response Headers' section of a browser's developer tools. The URL is `https://perfmatters.io/` and the status is 200. The headers are as follows:

Header	Value
<code>status</code>	200
<code>x-content-type-options</code>	<code>nosniff</code>
<code>content-encoding</code>	<code>gzip</code>
<code>set-cookie</code>	<code>PHPSESSID=d33c7f3bfa8f6c0a9ba55f3a6804e1...</code>
<code>x-kinsta-cache</code>	<code>MISS</code> ←
<code>expires</code>	<code>Thu, 19 Nov 1981 08:52:00 GMT</code>

Um die Geschwindigkeit deiner gesamten Webseite richtig wahrnehmen zu können, musst du alles aus dem Cache laden, deine Startseite und alle Assets, alles muss auf „HIT“ eingestellt sein. Das erfordert manchmal, dass du deinen Geschwindigkeitstest mehrmals durchführst. Dann kannst du den Durchschnitt nehmen.

The screenshot shows the 'Response Headers' section of a browser's developer tools. The URL is `https://perfmatters.io/` and the status is 200. The headers are as follows:

Header	Value
<code>status</code>	200
<code>x-content-type-options</code>	<code>nosniff</code>
<code>content-encoding</code>	<code>gzip</code>
<code>set-cookie</code>	<code>PHPSESSID=d33c7f3bfa8f6c0a9ba55f3a6804e1...</code>
<code>x-kinsta-cache</code>	<code>HIT</code> ←
<code>expires</code>	<code>Thu, 19 Nov 1981 08:52:00 GMT</code>

Kommen wir nun zu einigen Front-End-Optimierungen, die du auf deiner WordPress-Seite vornehmen kannst.

Query Strings entfernen

Eine häufige Warnung oder Empfehlung, die Menschen in Geschwindigkeitstesttools sehen ist, dass man Query-Strings entfernen sollte. Worum geht es hier? Nun, im Grunde genommen funktioniert es so, dass deine CSS- und JavaScript-Dateien normalerweise die Dateiversion am Ende ihrer URLs haben, wie zum Beispiel `https://domain.com/file.min.css?ver=4.5.3`. Einige Server und Proxy-Server können Query-Strings nicht zwischenspeichern. Wenn du sie also entfernst, kannst du manchmal dein caching verbessern.

Du kannst das manuell entfernen, indem du den folgenden Code in die Datei `functions.php` deines Themes einträgst.

```
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');
```

Oder du kannst ein Premium-Plugin wie Perfmatters (verfügbar unter perfmatters.io) verwenden, um Suchbegriffe mit einem einzigen Klick zu entfernen.

Mit Query Strings

Hier ist ein Beispiel für das Laden von Skripten mit Query-Strings.

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

Ohne Query Strings (nach dem Code)

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

Hier ist ein Beispiel für das Laden von Skripten ohne Query-Strings.

Bevor du jedoch sofort Query-Strings auf deiner Webseite streichst, ist es wichtig zu wissen, warum Query-Strings verwendet werden. Die Versionierung von Dateien wird typischerweise von WordPress-Entwicklern verwendet, um Caching-Probleme zu umgehen.

Wenn ein Plugin-Entwickler beispielsweise ein Update herausbringt und `style.css` from `?ver=4.6` auf `?ver=4.7` ändert, wird es als eine völlig neue URL behandelt und nicht gecached. Wenn du die Query-Strings entfernst und ein Plugin aktualisierst, kann das dazu führen, dass die zwischengespeicherte Version weiterhin genutzt wird.

In einigen Fällen kann es das Aussehen deiner Webseite beeinträchtigen, solange bis die zwischengespeicherte Ressource abläuft oder der Cache vollständig geleert ist.

Außerdem können einige CDNs Query-Strings zwischenspeichern. Das Kinsta CDN kann und tut das standardmäßig. Wenn du also ein Kinsta-Kunde bist, werden Query-Strings bereits in deinen Assets zwischengespeichert.

Eliminiere das Render-Blocking JavaScript und CSS

Eine Warnung vor Rendering-Blocking von JavaScript und CSS kann angezeigt werden, wenn du Dateien hast, die das schnellstmögliche Laden der Seite verhindern. Spezifische JS und CSS sind manchmal bedingt, d.h. sie sind nicht erforderlich, um Inhalte above the fold anzuzeigen. Du kannst verhindern, dass sie zum Rendering-Blocking werden, indem du `async` und `defer` Attribute verwendest.

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

Um Rendering-Blocking von JavaScript und CSS zu vermeiden, muss man folgendes tun:

JavaScript aus dem kritischen Rendering-Pfad löschen

Das Verschieben von JavaScript aus dem kritischen Rendering-Pfad erfolgt in der Regel durch Hinzufügen des `defer` oder `async`-Attributs zu den HTML-Elementen des Skripts, die JavaScript-Ressourcen aufrufen.

- Das **async-Attribut** weist den Browser an, sofort mit dem Herunterladen der Ressource zu beginnen, ohne das HTML-Parsing zu verlangsamen. Sobald die Ressource verfügbar ist, wird das HTML-Parsing angehalten, damit die Ressource geladen werden kann.
- Das **defer-Attribut** weist den Browser an, den Download der Ressource zu stoppen, bis das HTML-Parsing abgeschlossen ist. Sobald der Browser mit dem HTML fertig ist, lädt er alle verzögerten Skripte herunter und rendert sie in der Reihenfolge, in der sie im Dokument erscheinen.

Optimiere die CSS Ressourcen Auslieferung

Die Optimierung der Bereitstellung von CSS bedeutet im Wesentlichen, dass man herausfinden muss, wie man es ohne Render Blocking einstellt.

- Identifiziere die styles, die erforderlich sind, um Inhalte übersichtlich darzustellen, und stelle diese styles in Einklang mit dem HTML ein.
- Verwende CSS auf Geräten nur bei Bedarf.
- Verbleibendes CSS asynchron laden.

Das kann manchmal definitiv ein kniffliger Prozess sein und erfordert hin und wieder einiges an Optimierung, je nachdem wie du Skripte auf deiner Seite lädst. Hier sind ein paar WordPress-Plugins, die dabei helfen können:

- Autoptimize
- Async JavaScript
- Hummingbird

Für eine detaillierte Erklärung und einen Leitfaden empfehlen wir dir, unseren Beitrag über die Beseitigung von Render-Blocking von JavaScript und CSS zu lesen.

Kombiniere externes CSS und JavaScript in WordPress

Die kombinierte externe CSS-Warnung wird typischerweise bei der Verwendung eines CDNs angezeigt, da du deine CSS-Dateien auf einer externen Domain, wie z.B. **cdn.domain.com**, hostest. In der Vergangenheit war eine einfache Lösung, die CSS-Dateien schlicht zu komprimieren oder zu kombinieren, sodass sie in einer einzigen Anfrage geladen werden.

Wenn du jedoch einen Provider, der HTTP/2 unterstützt über HTTPS nutzt, ist diese Warnung nicht mehr so relevant wie früher. Mit HTTP/2 können nun mehrere CSS-

Dateien parallel über eine einzige Verbindung geladen werden. Und über 86% der Browser unterstützen HTTP/2 bereits.

Das bedeutet aber nicht unbedingt, dass diese Optimierung völlig hilflos ist. In einigen Fällen haben wir gesehen, dass das die WordPress-Seiten noch schneller macht. Es hängt von der Größe der Dateien ab und wie viele es von ihnen gibt. Daher ist das ein Optimierungsweg, den wir dir weiterhin empfehlen können.

Eine der einfachsten Möglichkeiten, deine externen CSS- und JavaScript-Dateien zu kombinieren, ist das kostenlose Autooptimize-Plugin. Nach der Kombination siehst du eine Datei „autooptimize_xxxxxx.css“ oder „autooptimize_xxxxx.js“. Es unterstützt auch das Laden aus dem CDN. Das WP Rocket Plugin bietet ähnliche Funktionen.

Name	Status	Domain
 editwp.com	200	editwp.com
 autooptimize_23d90d8a1eaacef4363d9e02c50ae363.css ←	200	mk0editwpn6...
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600...	200	fonts.googlea...
 jquery.js?ver=1.12.4	200	mk0editwpn6...
 header.jpg	200	mk0editwpn6...
 autooptimize_3fde0f2c0341cdc5afd7cc0c0e403f92.js ←	200	mk0editwpn6...
 jizDREVIthGc8qDlbSTKq4XkRiUf2zcZiVbJ.woff2	200	fonts.gstatic.c...
 jizAREVIthGc8qDlbSTKq4XkRi24_Sl0q1vjitOh.woff2	200	fonts.gstatic.c...
 jizAREVIthGc8qDlbSTKq4XkRi20-Sl0q1vjitOh.woff2	200	fonts.gstatic.c...

Nutze die Komprimierung von HTML, CSS, und JavaScript

Wir können die Datenmenge, die der Browser herunterladen muss, reduzieren, indem wir HTML-, CSS- und JavaScript-Ressourcen anpassen. Minifizierung ist der Prozess der Entfernung unnötiger Zeichen wie Kommentare und Leerzeichen aus dem Quellcode. Diese Zeichen sind in der Entwicklung äußerst nützlich, aber sie sind für den Browser unwichtig, um die Seite darzustellen.

Nicht-komprimiertes HTML

Hier ist ein Beispiel für einen nicht-komprimierten 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' />
```

Komprimierter HTML Code

Hier ist ein Beispiel für einen komprimierten 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://mk0editwpn6qbfrow5o9.kinstacdn.com/wp-
7 content/cache/autoptimize/css/autoptimize_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-
```

Du kannst die kostenlosen Plugins Autoptimize oder WP-Rocket verwenden, um deine Dateien so auf einfache Weise zu verkleinern.

Nutze Cookie-freie Domains

Wenn man Inhalte wie Bilder, JavaScript oder CSS bereitstellt, gibt es in der Regel keinen Grund, einen HTTP-Cookie mitzusenden, da dies zusätzlichen Aufwand verursacht. Sobald der Server ein Cookie für eine bestimmte Domain setzt, müssen alle nachfolgenden HTTP-Anfragen für diese Domain das Cookie auch enthalten. Diese Warnung wird typischerweise auf Webseiten mit einer großen Anzahl von Anfragen angezeigt.

Wir haben einen ausführlichen Beitrag darüber, wie man mit dem statischen Inhalt des Servers aus einer Cookieless Domain Warnung umgeht. Oftmals kann man diese Warnung ignorieren, da das bei neuen Protokollen wie HTTP/2 an Bedeutung verliert. Die Kosten für eine neue Verbindung sind in der Regel teurer als das Streamen aller Daten über dieselbe Verbindung.

Eine einfache Möglichkeit, diese Warnung zu beheben, besteht darin, einen CDN-Anbieter zu verwenden, der Cookies sowohl ignorieren als auch entfernen kann, was den Kunden vollständig davor bewahrt, den Set-Cookie-Response-Header zu empfangen. KeyCDN ist ein CDN-Anbieter, der diese Funktion anbietet. Standardmäßig siehst du, dass die folgenden beiden Optionen aktiviert sind. Das ist eine einfache Alternative, ohne sich mit dem Verschieben und Konfigurieren deiner Webseite herumschlagen zu müssen, um statische Assets aus einer separaten Subdomain bereitzustellen.

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.**

Wenn du Cloudflare verwendest, kannst du Cookies für Ressourcen, die über ihr Netzwerk bereitgestellt werden, nicht deaktivieren. CloudFlare fügt ein eigenes Sicherheits-Cookie in den Header ein. Auch diese Cookies sind sehr klein und die Auswirkungen auf die Performance sind äußerst gering. Aber wenn du CloudFlare benutzt, gibt es keine Möglichkeit, diese Warnung zu verhindern.

Eine zweite Möglichkeit, diese Warnung zu umgehen, besteht darin, die WordPress-Seite neu zu konfigurieren, um die statischen Elemente von einer neuen Domain oder Subdomain bereitzustellen.

Deaktiviere Einbettungen in WordPress

Mit der Veröffentlichung von WordPress 4.4, wurde die oEmbed-Funktion in Core integriert. Das ermöglicht es YouTube-Videos, Tweets und viele andere Ressourcen auf Webseiten einzubetten, indem eine einfache URL eingefügt wird, die WordPress dann automatisch in Embedded Code umwandelt, um eine Live-Vorschau im visuellen Editor anzuzeigen. Mit dem Update wurde WordPress selbst zum oEmbed-Anbieter.

Diese Funktion ist für viele Menschen nützlich, und du solltest sie vielleicht aktiviert lassen. Das bedeutet jedoch, dass es auch einen zusätzlichen HTTP-Request auf deiner WordPress-Seite erzeugt, um die Datei wp-embed.min.js zu laden. Und das lädt seitenübergreifend. Während diese Datei nur 1,7 KB groß ist, summieren sich solche Dinge mit der Zeit. Die Anfrage selbst ist manchmal ein größeres Problem als die Größe des Content-Downloads.

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

Es gibt verschiedene Möglichkeiten, Einbettungen in WordPress zu deaktivieren. Du kannst es mit den folgenden Plugins tun:

- Disable Embeds (kostenlos, verfügbar auf WordPress.org)
- Perfmatters (Verfügbar unter perfmatters.io)

Oder du kannst auch Einbettungen mit einem Code deaktivieren. Aufgrund seiner Länge ist er nur auf unserer Webseite unter <https://kinsta.com/knowledgebase/disable-embeds-wordpress/> zu finden.

Emojis in WordPress deaktivieren

Neben Embeds wurde in WordPress 4.2 die Unterstützung für Emojis in älteren Browsern geliefert. Das große Problem dabei ist, dass es eine zusätzliche HTTP-Anfrage auf deiner WordPress-Seite erzeugt, um die Datei `wp-emoji-release.min.js` zu laden. Und das lädt seitenübergreifend. Während diese Datei nur 10,5 KB groß ist, ist sie nutzlos, wenn du keine Emojis auf deiner Webseite verwendest.

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....

Es gibt verschiedene Möglichkeiten, Emojis in WordPress zu deaktivieren. Du kannst es mit den folgenden Plugins tun:

- Disable Emojis (kostenlos, verfügbar auf WordPress.org)
- Perfmatters (Verfügbar unter perfmatters.io)

Oder du kannst auch Emojis mit einem Code deaktivieren. Aufgrund seiner Länge ist er nur auf unserer Webseite unter <https://kinsta.com/knowledgebase/disable-emojis-wordpress/> zu finden.

So beschleunigt man WordPress Kommentare oder deaktiviert das Theme

Ein beliebter Kommentarbereich auf einer Webseite kann viele Performance-Probleme verursachen. Denke nur an die Ressourcen, mit denen Kommentare funktionieren:

- Eine Datenbank wird abgefragt, um vorhandene Kommentare aufzurufen.

- Für jeden neuen Kommentar werden Datenbankeinträge angelegt.
- Kommentare und Kommentar-Metadaten werden vom Browser eines Besuchers empfangen und verarbeitet.
- Externe Ressourcen, wie z.B. Gravatare, werden angefordert, heruntergeladen und geladen (erfordert ein separates DNS-Lookup).
- In vielen Fällen müssen große JavaScript- und jQuery-Ressourcen heruntergeladen und verarbeitet werden, damit das kommentierende System so funktioniert, wie es soll.

Hier sind vier verschiedene Optionen, wie du WordPress-Kommentare beschleunigen kannst:

Möglichkeit 1 – Kommentare deaktivieren

Wenn deine Seite nicht sehr viele Kommentare erhält und du nicht denkst, dass sie einen Mehrwert bringen, könnte es besser sein, Kommentare ganz zu deaktivieren. Denk daran, dass Kommentare Auswirkungen auf dein SEO haben können, da Google diese in der Regel als zusätzlichen Inhalt auf der Seite durchsucht, so dass du nur qualitativ hochwertige Kommentare genehmigen solltest. Schau dir diese drei einfachen Möglichkeiten an, um Kommentare zu deaktivieren:

1. Kommentare in den WordPress Einstellungen deaktivieren

Du kannst Kommentare auf einer Seite oder einem Beitrag einzeln deaktivieren, indem du sie bearbeitest und die Option “Kommentare zulassen” im Abschnitt “Diskussion” deaktivierst. Wenn du diesen Abschnitt nicht siehst, musst du möglicherweise nach oben scrollen und ihn unter “Bildschirmoptionen” aktivieren.

Comments

Appearance

Plugins

Snippets

Users

Tools

Settings

Collapse menu

— Select —

Enter new

Add Custom Field

Custom fields can be used to add extra metadata to a post that you can [use in your theme](#).

Discussion

Allow comments

Allow [trackbacks and pingbacks](#) on this page

Du kannst Diskussionen auch global deaktivieren. Du kannst das tun, indem du in deinem WordPress-Dashboard unter Einstellungen in “Diskussion” klickst und dann die Option “Erlaube es Leuten, Kommentare zu neuen Artikeln zu schreiben” deaktivierst. Der Nachteil dieser Methode ist jedoch, dass es ältere Artikel nicht betrifft, sondern nur neue.

Wenn du mehr Kontrolle benötigst und einfach alle Kommentare auf einmal global deaktivieren möchtest, empfehlen wir dir, eine der folgenden Methoden zu verwenden.

Dashboard

Kinsta Cache

Posts

Media

Pages

Comments

Appearance

Plugins

Snippets

Users

Tools

Settings

General

Writing

Reading

Discussion

Media

Help

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.)

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 14 days

Show comments cookies opt-in checkbox.

Enable threaded (nested) comments 5 levels deep

Break comments into pages with

50 top level comments per page and the

last page displayed by default

Comments should be displayed with the older comments at the top of each page

2. Kommentare mit einem Plugin deaktivieren

Eine der einfachsten Möglichkeiten, Kommentare zu deaktivieren, ist die Verwendung eines kostenlosen Plugins wie Disable Comments (verfügbar auf WordPress.org). Du kannst Kommentare überall mit einem Klick deaktivieren oder sie für Beiträge, Seiten oder Medienposttypen deaktivieren.

3. Kommentare mit einem Code deaktivieren

Du kannst Kommentare auch mit Code deaktivieren, indem du einfach den Kommentarbereich entfernst, der in deinem WordPress-Theme eingebaut ist.

Ersetze in deiner `page.php`-Datei einfach den folgenden Code:

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

Mit diesem Code:

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

Dadurch wird der Bereich von deiner Seite entfernt. Um die Kommentare schließlich auch deinen Blog-Einträgen zu entfernen, navigiere zu deiner `single.php`-Datei im Theme und ersetze den folgenden Code:

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

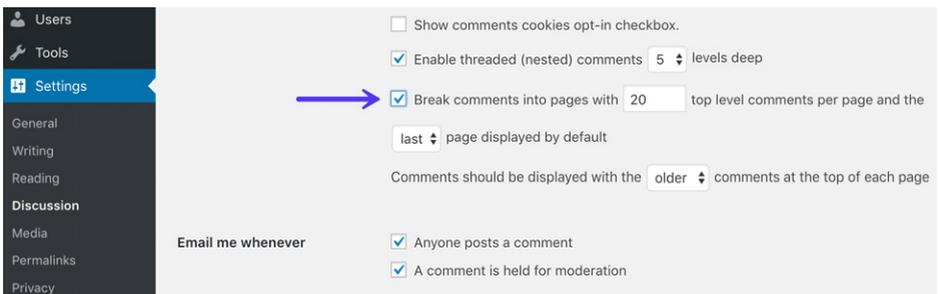
Mit diesem Code:

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

Möglichkeit 2 – Optimierte die Native WordPress Kommentare

Deine zweite Option wäre, das native WordPress-Kommentarsystem zu optimieren. Eine Möglichkeit ist hier die Anzahl der Kommentare zu reduzieren, die beim Aufrufen der ersten Seite geladen werden.

1. Gehe zu Einstellungen / Diskussion im WordPress-Administrationsbereich.
2. Suche den Abschnitt andere Kommentareinstellungen.
3. Wähle das Kontrollkästchen neben "Breche Kommentare in Seiten" und füge einen Wert für die Anzahl der Kommentare hinzu, die du beim Laden der ersten Seite anzeigen möchtest.



Eine weitere Möglichkeit, die du hast, ist die Verwendung von Host-Gravatoren in deinem CDN. Das ist der Ansatz, den wir bei Kinsta verfolgen.

Standardmäßig, wenn WordPress-Kommentare geladen werden, benötigt jeder einzelne eindeutige Gravatar eine HTTP-Anfrage. Wenn also eine Seite mit Kommentaren von 50 verschiedenen Kommentatoren geladen wird, sind 50 HTTP-Anfragen erforderlich, um alle diese Gravatars herunterzuladen. Wie du dir vorstellen kannst, kann das die Seitengeschwindigkeit beeinflussen. Ganz zu schweigen von der Tatsache, dass wir gesehen haben, wie das externe DNS-Lookup zu gravatar.com manchmal langsam wird und in einigen Fällen sogar Zeitüberschreitungen hervorruft.

Wenn du dir Gravatars im Kinsta-Blog ansiehst, kannst du sehen, dass sie von Kinsta.com

Achte immer darauf, dass du das Kommentarsystem eines Drittanbieters, das du gerade ausprobierst, kurz testest. Wirf einen Blick auf alle separaten Anforderungen, die beispielsweise Disqus generiert (wie oben gezeigt). Während die meisten dieser Anfragen asynchron geladen werden, wird dir bei der Verwendung von Disqus noch etwas zusätzliche Ladezeit auffallen.

Möglichkeit 4 – Kommentare mit der Lazy Load funktion ausstatten

Deine vierte Option besteht darin, Lazy Loading für Kommentare zu aktivieren, damit sie das anfängliche Darstellen der Seite nicht verlangsamen. Hier sind ein paar Plugins, die du vielleicht ausprobieren möchtest:

- **Lazy Load for Comments:** Dieses Plugin ermöglicht es dir, das Lazy Loading für native WordPress-Kommentare zu aktivieren.
- **Disqus Conditional Load:** Wenn Du das Disqus-Komentarsystem verwenden möchtest, ist dies ein Must-Plugin, um Kommentaren die Lazy Load funktion hinzuzufügen.

Deaktivieren des WordPress RSS Feeds

Wenn du den Blogging-Teil von WordPress auf deiner Webseite nicht verwendest, kannst du die WordPress-RSS-Feeds deaktivieren. Das hat zwar keinen großen Einfluss auf die Leistung, aber auch die kleinen Details helfen. Es ist auch eine Sache weniger, um die du dir Sorgen machen musst.

Sieh dir diese beiden verschiedenen Möglichkeiten an, um RSS-Feeds in WordPress zu deaktivieren:

1. Den RSS Feed mit einem Plugin deaktivieren

Die erste Möglichkeit, einen WordPress RSS-Feed zu deaktivieren, ist die Verwendung

eines kostenlosen Plugins wie Disable Feeds (verfügbar auf WordPress.org). Dieses Plugin deaktiviert alle RSS/Atom-Feeds auf Ihrer WordPress-Seite, indem es alle Anfragen umleitet.

Du kannst auch ein Premium-Plugin wie Perfmatters (entwickelt von einem Teammitglied bei Kinsta, verfügbar unter perfmatters.io) verwenden, mit dem du sowohl RSS-Feeds deaktivieren als auch RSS-Feed-Links deaktivieren kannst, zusammen mit anderen Optimierungen für deine WordPress-Seite.

2. Den RSS Feed mit einem Code deaktivieren

Wenn du kein Plugin verwenden möchtest, kannst du auch RSS-Feeds mit einem Code deaktivieren. Füge den folgenden Code in die Datei `functions.php` deines Themes hinzu.

```
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 generiert auch Links zu den RSS-Feeds im Header deiner Webseite. Du kannst einen Schritt weiter gehen und diese Links aus dem HTML-Code deiner Seiten entfernen. Füge dazu den folgenden Code in die Datei `functions.php` deines Themes ein.

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

Verwende Prefetch and Preconnect

Ressourcenverweise und Anweisungen wie prefetch und preconnect können eine gute Möglichkeit sein, WordPress hinter den Kulissen zu beschleunigen.

Prefetch

Mit dem DNS-prefetch kannst du Domainnamen auflösen (DNS-Lookup im Hintergrund durchführen), bevor ein Benutzer auf einen Link klickt, was wiederum zur Verbesserung der Leistung beitragen kann. Dazu wird ein `rel="dns-prefetch"`-Tag im Header deiner WordPress-Seite hinzugefügt.

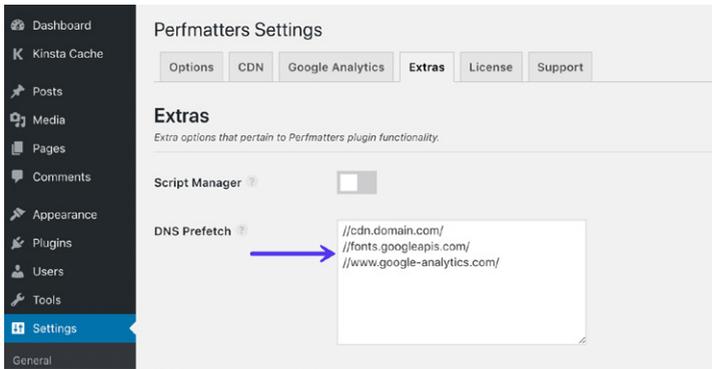
```
<link rel="dns-prefetch" href="//domaine.com">
```

Einige gängige Aspekte, für die man DNS-Prefetching verwenden sollte, sind die CDN-URL, Google-Fonts, Google Analytics, etc.

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

Prefetch wird auch von den meisten modernen Browsern unterstützt. Es gibt ein paar Möglichkeiten, das umzusetzen.

- 1.** Du kannst den Code zu deinem Header hinzufügen, indem du ein kostenloses Plugin wie Head, Footer und Post Injections verwendest (verfügbar auf WordPress.org).
- 2.** Du kannst den Code manuell in die `header.php`-Datei deines Themes einfügen.
- 3.** Implementiere DNS-Prefetch mit einem Premium-Plugin wie Perfmatters (verfügbar unter perfmatters.io). Klicke einfach auf die Registerkarte "Extras" im Perfmatters-Plugin und füge Domains hinzu. Format: `//domain.tld` (eine pro Zeile)



Preconnect

Preconnect ermöglicht es dem Browser, frühe Verbindungen vor einer HTTP-Anfrage einzurichten.

Preconnect ist ein wichtiges Werkzeug in deiner Optimierung-Toolbox... es kann viele teure Roundtrips von deinem Anforderungspfad eliminieren – in einigen Fällen reduziert es die Anforderungslatenz um Hunderte und sogar Tausende von Millisekunden.



Iya Grigorik

Das geschieht durch das Hinzufügen eines `rel="preconnect"`-Tags im Header deiner WordPress-Seite.

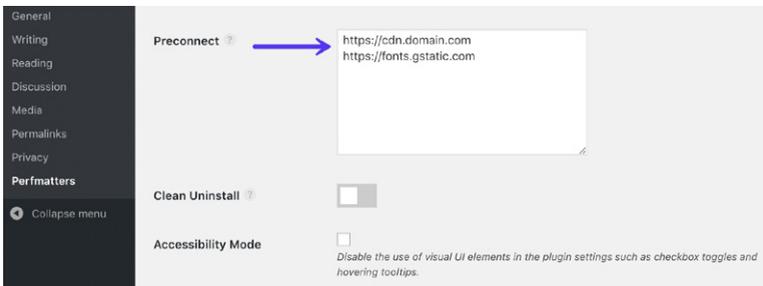
```
<link rel="preconnect" href="//domaine.com">
```

Einige Beispiele für Dinge, für die du es verwenden könntest, sind deine CDN-URL oder Google-Fonts.

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

Preconnect wird von den meisten modernen Browsern unterstützt, mit Ausnahme von Internet Explorer, Safari, IOS Safari und Opera Mini. Es gibt ein paar Möglichkeiten, das umzusetzen.

1. Du kannst den Code zu deinem Header hinzufügen, indem du ein kostenloses Plugin wie Head, Footer und Post Injections verwendest (verfügbar auf WordPress.org).
2. Du kannst den Code manuell in die `header.php`-Datei deines Themes einfügen.
3. Implementieren die Vorverbindung mit einem Plugin wie Perfmatters (verfügbar unter perfmatters.io). Klicke einfach auf die Registerkarte "Extras" im Perfmatters-Plugin und füge die Domains hinzu. Format: `scheme://domaine.tld` (eine pro Zeile).



Scripts auf Post- und Seitenebene deaktivieren

Eine weitere sehr effektive Möglichkeit, WordPress zu beschleunigen, besteht darin, jede Anfrage zu durchsuchen, die auf deinen Seiten und Beiträgen geladen wird. Du wirst höchstwahrscheinlich Skripte finden, die seitenübergreifend laden, was nicht sein sollte.

Du kannst ein Premium-Plugin wie Perfmatters verwenden, das über eine integrierte

„Script Manager“-Funktion verfügt. So kannst du Skripte (CSS und JavaScript) auf Seiten-/Postbasis oder sogar seitenübergreifend mit einem einzigen Klick deaktivieren. Auch dieses Plugin wird von einem Teammitglied bei Kinsta entwickelt.

Einige Beispiele dafür, wofür dies genutzt werden kann:

- Das beliebte Contact Form 7 Plugin lädt sich auf jeder Seite und jedem Beitrag. Du kannst es einfach überall mit einem Klick deaktivieren und nur auf deiner Kontaktseite aktivieren.
- Social Media Sharing Plugins sollten nur in deinen Beiträgen geladen werden. Du kannst es einfach überall deaktivieren und nur auf Posttypen oder sogar auf benutzerdefinierte Posttypen laden.
- Das Inhaltsverzeichnis Plugin (TOC) lädt auf jeder Seite und jedem Beitrag. Mit dem Skript-Manager kannst du ganz einfach steuern, wo er geladen werden soll.
- Wenn Du auf WordPress 5.0 aktualisiert hast und nicht den Gutenberg-Blockeditor verwendest - vielleicht verwendest du noch den klassischen Editor oder einen anderen Drittanbieter-Editor - gibt es zwei zusätzliche Frontend-Skripte, die seitenweit hinzugefügt werden, die du deaktivieren kannst: `/wp-includes/css/dist/block-library/style.min.css` und `/wp-includes/css/dist/block-library/theme.min.css`.

Warum sind manche Plugins so geschrieben?

Du fragst dich vielleicht, warum alle Plugin-Entwickler ihre Skripte nicht einfach nur dann laden, wenn das Plugin auf der Seite erkannt wird? Nun, es ist etwas komplizierter als das. Wenn du zum Beispiel ein Plugin wie das Contact Form 7 hast, hat es auch Shortcodes, mit denen du es überall platzieren kannst. Dazu gehört auch das Ablegen in einem Widget. Mit WordPress ist es viel schwieriger, Daten von ihnen abzufragen, wenn man Skripte aus der Warteschlange entfernt, als Daten aus den Post- oder Seitenmetadaten abzufragen.

Daher ist dies oft auf Usability-Probleme zurückzuführen. Je kleiner die Chancen,

dass ein Plugin aufhört zu funktionieren, desto weniger Supporttickets landen beim Entwickler. Allerdings zeigen viele Plugins heutzutage, dass es schon möglich wäre Plugins so zu schreiben, dass sie von grundauf für Performance optimiert sind. Leider macht manchmal die schiere Anzahl von Downloads und Benutzern die Programmierung für die Benutzerfreundlichkeit zu einer Priorität.

Eine Tour durch den Script Manager

Wir geben dir einen kleinen Einblick in den Script Manager. Nachdem du es in deiner Symbolleiste angeklickt hast, werden dir alle Skripte angezeigt, die auf dieser aktuellen URL geladen werden, sowohl JavaScript- als auch CSS-Dateien. Du hast dann die folgenden Möglichkeiten:

- 1. Status On :** (Standardeinstellung)
- 2. Status Off :** Überall deaktivieren (Du kannst dann wählen, für welche Beitragsarten du sie aktivieren möchtest, zusammen mit der aktuellen URL).
- 3. Status Off :** Deaktiviere nur die aktuelle URL (dies ist sehr nützlich für die Verwendung auf deiner Homepage).
- 4. Status Off :** Ausnahmen (aktuelle URL, Posttyp oder Archiv)

The screenshot displays the 'Script Manager' interface for Perfmatters. It features a sidebar on the left with 'Script Manager' selected, and a main content area. The main area is divided into two sections: 'Extended Widget Options' and 'Social Warfare'. Each section contains 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 'OFF' status includes options for 'Disabled', 'Everywhere', 'Current URL', 'Exceptions', and 'Post Types'.

Alles ist nach **dem Plugin- oder Theme-Namen gruppiert**. Das macht es super einfach, ein ganzes Plugin auf einmal zu deaktivieren. Normalerweise hat ein WordPress-Plugin sowohl eine JavaScript- als auch eine CSS-Datei. Ein WordPress-Theme kann mehr als 10 Dateien enthalten.

Nachdem du die Einstellungen ausgewählt und geändert hast, musst du unten auf „Speichern“ klicken. Du kannst dann mit einem Geschwindigkeitstool testen, ob die Skripte nicht mehr auf die Seite oder den Beitrag geladen werden. Stelle sicher, dass du zuerst deinen Cache leerst! Und wenn auf deiner Webseite visuell etwas schief geht, kannst du es in den Einstellungen jederzeit wieder aktivieren, um alles wieder herzustellen.

In einem Geschwindigkeitstest von woorkup konnten sie die **Gesamtladezeiten um 20,2% reduzieren**. Allein auf ihrer Homepage konnten sie die Anzahl der HTTP-Requests von 46 auf 30 reduzieren. Auch die Seitengröße **schrumpfte von 506,3 KB auf 451,6 KB**.

Analyse der Performance von Drittanbietern

Grundsätzlich hat alles, was du von deiner Webseite aus extern aufrufst, Konsequenzen für die Ladezeit. Was dieses Problem noch verschlimmert, ist, dass einige Abfragen nur sporadisch langsam sind, was die Identifizierung des Problems noch schwieriger macht.

Ein externer Dienst eines Drittanbieters könnte als alles angesehen werden, was mit deiner WordPress-Seite von außerhalb deines eigenen Servers kommuniziert. Hier sind einige häufige Beispiele, die uns regelmäßig begegnen:

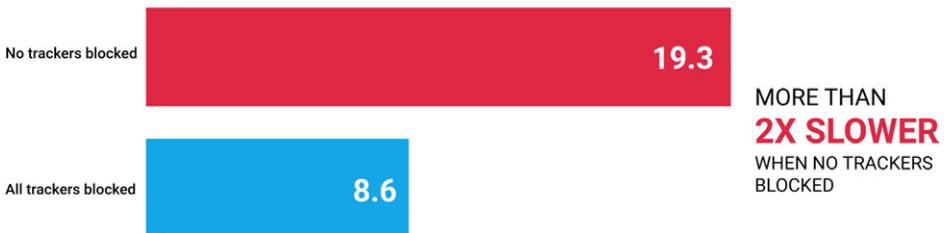
- Social Media Plattformen wie Twitter, Facebook und Instagram (Widgets oder Konvertierungspixel)
- Drittworbenetzwerke wie Google AdSense, Media.net, BuySellAds, Amazon Associates, etc.
- Seitenanalyse und Tracking-Skripte wie Google Analytics, Crazy Egg, Hotjar, AdRoll etc.
- A/B-Testtools wie Optimizely, VWO, Unbounce, etc.

- WordPress Kommentarsysteme wie Disqus, Jetpack, Facebook Kommentare
- Backup- und Sicherheitstools wie VaultPress, Sucuri, CodeGuard etc.
- Tools für den sozialen Austausch wie SumoMe und HelloBar
- CDN-Netzwerke wie KeyCDN, Amazon CloudFront, CDN77 und StackPath
- Extern gehostetes Javascript

Wie stark beeinflussen einige dieser Tracker von Drittanbietern die Leistung? In unserer eigenen Fallstudie haben wir festgestellt, dass Skripte von Drittanbietern **die Seitenladezeiten um 86,08% erhöht haben.**

Ghostery maß auch die Top 500 US-Domains in Alexa, und die Ergebnisse waren erstaunlich, wenn auch für uns nicht überraschend. Webseiten waren 2x langsamer, wenn überhaupt keine Tracker blockiert wurden. Das bedeutet, dass diese Tracking-Skripte von Drittanbietern einer der Hauptfaktoren für langsame Seitenladegeschwindigkeiten im Web sind.

AVERAGE PAGE LOAD TIME WITH vs. WITHOUT TRACKERS



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

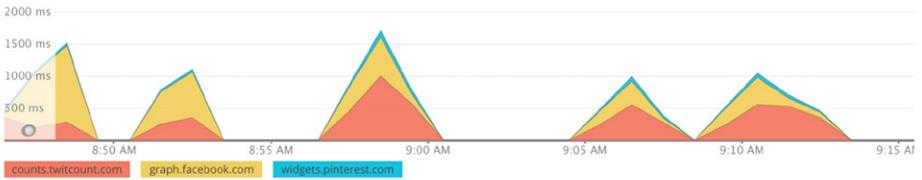
Bildquelle: Ghostery

Du musst sehr vorsichtig auf deiner WordPress-Seite sein. Nur ein einziger fehlerhafter API-Aufruf eines Drittanbieters könnte die gesamte Webseite ausfallen lassen!

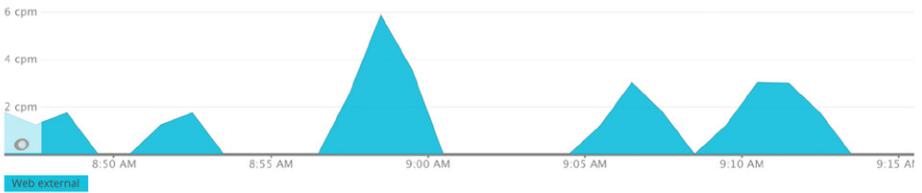
Ja, so sollte es nicht funktionieren, aber in vielen Fällen passiert das leider. Wir haben das öfter miterlebt, als wir zählen können.

New Relic bietet eine ausgezeichnete und einfache Möglichkeit, die externen Dienste im Laufe der Zeit zu überwachen. In diesem Beispiel unten sehen wir externe Anrufe an [twitcount.com](https://www.tweetcount.com), [graph.facebook.com](https://www.facebook.com) und [widgets.pinterest.com](https://www.pinterest.com).

Top 5 external services by total response time



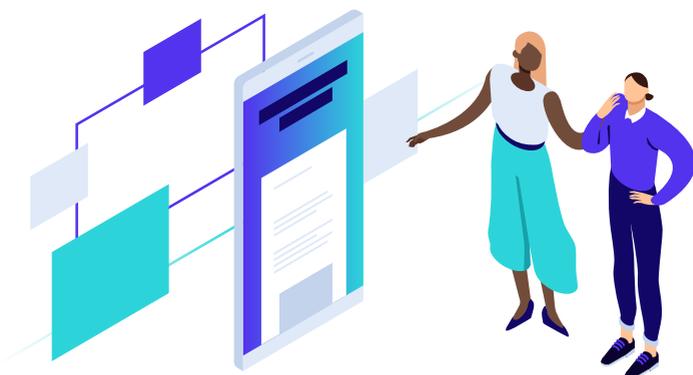
External calls per minute



Es ist wichtig, dass, wann immer du eine neue Funktion oder ein Plugin zu deiner Webseite hinzufügst, du die externen Ressourcen untersuchst, die von ihr geladen werden. Je weniger, desto besser!

KAPITEL 16:

Optimiere alles mit der Prämisse “Mobile First” im Hinterkopf



Google begann am 26. März 2018 mit der Einführung des Mobile-First-Index. Bisher haben die Crawler-, Indexierer- und Rankingsysteme von Google die Desktop-Version von Webseiten verwendet. Mobile-first indexing bedeutet, dass Googlebot nun die mobile Version deiner WordPress-Seite für die Indexierung und das Ranking verwendet. Dies trägt dazu bei, das Sucherlebnis für mobile Nutzer zu verbessern.

Wenn es darum geht, deine Webseite für mobile-first zu optimieren, ist **Geschwindigkeit einer der wichtigsten Faktoren**, auf die du dich konzentrieren solltest. Geschwindigkeit spielt eine große Rolle in allem, von der Benutzerfreundlichkeit bis hin zu Bounce-Raten und der Entscheidung, ob potenzielle Käufer auf deine Webseite zurückkehren werden oder nicht. Tatsächlich ist die Geschwindigkeit heute ein Schlüsselfaktor von Google Search und Anzeigen für die mobile Suche.

Schlechte mobile Erfahrungen werden die Mehrheit der Nutzer dazu bringen, nie wieder zurückzukehren. Laut dem neuesten Google Page Speed Report betrug die durchschnittliche Ladezeit einer mobilen Webseite im Jahr 2018 15 Sekunden. Kannst du dir vorstellen, so lange zu warten, um eine einzelne Seite zu laden? Erstaunlich.

Die Nutzer fordern (und verdienen) mehr. Laut dem gleichen Bericht über die **Seitengeschwindigkeit verlassen 53% der Besucher mobiler Webseiten, diejenigen Seiten, die länger als drei Sekunden zum Laden benötigen.**

Langsame mobile Erlebnisse verhindern nicht nur die Konvertierung. Sie hindern dich daran, überhaupt eine Chance zu bekommen, potenzielle Kunden zu konvertieren. Wenn die Ladezeiten der Seiten um nur wenige Sekunden zunehmen, steigt die Wahrscheinlichkeit, dass jemand bouncet, exponentiell an. Hier sind einige Dinge, die bei der Optimierung für das Handy zu beachten sind.

Überprüfe deinen mobilen Traffic

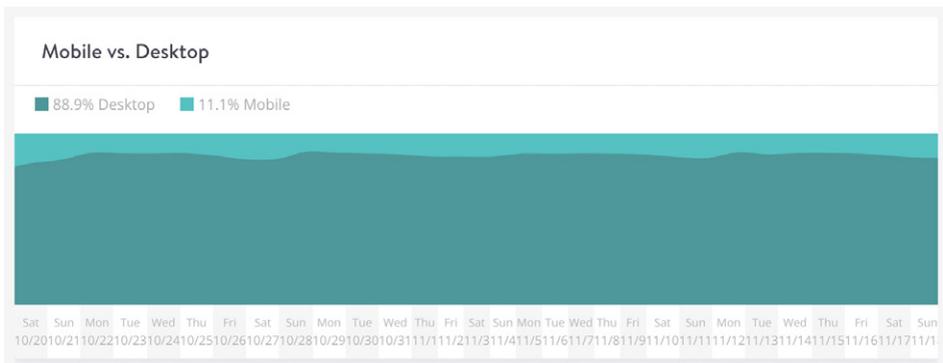
Es ist immer wichtig, einen Blick darauf zu werfen, wieviel mobilen Traffic du bekommst, da sich dadurch deine Prioritäten ein wenig verschieben können. Wie viele mobile Geräte deine Seite in Google Analytics besuchen, kannst du unter „Audience / Mobile / Overview“ sehen. Wie du auf dieser Seite sehen kannst, sind über 67% des gesamten Datenverkehrs vom Handy aus möglich. Das ist eine Menge!

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%)

Wenn du ein Kinsta-Kunde bist, kannst du auch deinen mobilen vs. Desktop-Traffic im MyKinsta Analytics-Tool überprüfen. Wie du auf dieser Seite sehen kannst, stammen über 88% des Datenverkehrs vom Desktop. Es ist immer wichtig zu überprüfen und nicht nur anzunehmen. Nur weil jeder sagt, dass die Dinge mobil werden, bedeutet das nicht immer, dass das für deine Seite gilt. Sieh dir unbedingt immer die Daten an.



Stelle sicher, dass deine Seite responsive ist

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

Im Jahr 2019 sollte deine Webseite responsive sein! Das bedeutet, dass es Medienabfragen nutzt, um die Dinge auf mobilen Geräten automatisch zu reduzieren. Wenn du das immer noch nicht getan hast, liegst du höchstwahrscheinlich schon hinter deiner Konkurrenz zurück. Alle WordPress-Themes, die wir bereits in diesem Beitrag erwähnt

haben, sind voll responsive und sehen auf allen Geräten fantastisch aus.

Verwende das Mobile-Friendly-Tool von Google, um zu testen und sicherzustellen, dass Deine Webseite alle Anforderungen erfüllt.

Überprüfe doppelt ob srcset funktioniert

Bisher war es sehr wichtig, dass man Bilder hochlädt, um sie zu skalieren und sie nicht von CSS skalieren zu lassen. Das ist jedoch nicht länger von Bedeutung, da WordPress 4.4 nun responsive Bilder unterstützt (nicht durch CSS verkleinert). WordPress erstellt automatisch mehrere Größen von jedem Bild, das in die Medienbibliothek hochgeladen wird. Durch das Einfügen aller verfügbaren Größen eines Bildes in ein srcset-Attribut können Browser nun die am besten geeignete Größe herunterladen und die anderen ignorieren. Sieh dir unten ein Beispiel an, wie dein Code aussehen könnte.

```

```

Aufgrund all der Bild-Plugins und Anpassungen von Drittanbietern gab es häufig Fälle, in denen wir gesehen haben, dass das nicht richtig funktioniert. Daher ist es wichtig, noch einmal zu überprüfen, ob die Bilder stimmig sind und das Attribut srcset mit verschiedenen Versionen für die unterschiedlichen Bildschirmgrößen hinzugefügt wird. Die Bildoptimierung ist nun stets von Bedeutung.

Google AMP ist eventuell eine Lösung für dich

Google AMP (Accelerated Mobile Pages Project) wurde ursprünglich bereits im Oktober 2015 eingeführt. Das Projekt basiert auf AMP HTML, einem neuen offenen Framework, das vollständig aus bestehenden Web-Technologien besteht und es Webseiten ermöglicht, leichte Seiten zu erstellen. Um es einfach auszudrücken, bietet es die Möglichkeit, eine abgespeckte Version deiner aktuellen Webseite bereitzustellen.

Wir haben eine Art Hassliebe zu Google AMP, und so geht es auch einem Großteil der Community. Wir haben die Technik selbst getestet und keine guten Ergebnisse erzielt. Das bedeutet jedoch nicht, dass es für dich keine Vorteile bringen könnte. Jede Webseite ist anders, und Google AMP wird ständig verbessert.

Du kannst mit einem der folgenden Plugins schnell mit Google AMP auf Deiner WordPress-Seite beginnen:

- AMP for WordPress (kostenlos, verfügbar auf WordPress.org)
- AMP for WP (kostenlos, verfügbar auf WordPress.org)

Zusammenfassung

Wie du wahrscheinlich bemerkt hast, lieben wir es, alle Methoden der WordPress Optimierung genauer zu untersuchen und auszuprobieren. Eine schnelle Webseite hilft, deine Rankings zu verbessern, die Crawlability für Suchmaschinen zu erhöhen, die Konversionsraten zu steigern, die Zeit, die Benutzer auf einer Webseite verbringen zu erhöhen und deine Bounce-Rate zu reduzieren. Ganz zu schweigen von der Tatsache, dass jeder gerne eine schnelle Webseite besucht!

Wir hoffen, dass dieser Guide hilfreich war und dass du ein paar Dinge mitnehmen und auf deine WordPress-Seite anwenden kannst.

Schaue dir auch unser WordPress Hosting-Partnerprogramm unter <https://kinsta.com/de/partnerprogramm/> an. Verdiane bis zu \$500 für jede Empfehlung + 10% monatlich wiederkehrende Lifetime-Provisionen. Das ist das beste Affiliate-Angebot, das du in der Branche finden wirst. Lasst uns gemeinsam das Web schneller machen!

Um mehr über Kinsta zu erfahren, besuche uns gerne unter <https://kinsta.com/de/>

Vielen Dank fürs Lesen!



Notizen

Die Geschwindigkeit deiner WordPress Webseite ist wichtiger, als du vielleicht denkst. Es spielt keine Rolle, welche Art von Webseite du betreibst; ob es sich nun um einen WooCommerce-Shop oder einen Blog handelt, die Ladezeiten deiner Seite ist direkt mit folgenden Aspekten verknüpft:

- Bessere Keyword-Rankings (Geschwindigkeit ist ein Faktor, den Google verwendet)
- Erhöhte Conversionrates
- Verbesserte Indexierung durch Suchmaschinen wie Google und Bing
- Längere durchschnittliche Verweildauer auf deiner Webseite
- Verringerte Bounce-Rates

Ganz zu schweigen von der Tatsache, dass eine schnell ladende Webseite ein besseres Gesamterlebnis für deine Besucher bietet.

Mehr als ein Drittel aller Webseiten im Internet werden heute mit WordPress betrieben. Obwohl das ein sehr guter Wert ist, bedeutet es auch, dass es Tausende von verschiedenen Themes, Plugins und Methoden gibt, die alle nebeneinander existieren müssen. Das kann Seiten oft verlangsamen. Für den alltäglichen WordPress-Benutzer kann das schnell zu einem Albtraum werden, wenn die Geschwindigkeit der eigenen Webseite darunter leidet. Viele wissen nicht, warum das der Fall ist oder wo sie mit der Fehlersuche beginnen sollen.

Das Team von Kinsta beschäftigt sich täglich mit diesen Problemen und hat schon alles gesehen! In diesem Leitfaden tauchen wir tief in die anwendbaren Schritte ein, denen du heute folgen kannst, um deine WordPress-Seiten zu beschleunigen. Alles, von Anfängertipps bis hin zu fortgeschrittenen Strategien, du wirst bestimmt etwas Nützliches finden. Der Guide enthält weit über 15 Jahre Erfahrung und viele Lektionen, die wir gelernt haben.

KINSTA