



Whitepaper zu AWS

Bewährte Methoden für WordPress in AWS



Bewährte Methoden für WordPress in AWS: Whitepaper zu AWS

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, die Kunden zu verwirren oder Amazon in einer Weise herabzusetzen oder zu diskreditieren. Alle anderen Marken, die nicht Eigentum von Amazon sind, sind Eigentum ihrer jeweiligen Inhaber, die mit Amazon verbunden oder nicht verbunden oder von Amazon gesponsert oder nicht gesponsert sein können.

Table of Contents

Überblick	1
Überblick	1
Einführung	2
Einfache Bereitstellung	3
Überlegungen	3
Verfügbare Ansätze	3
Amazon Lightsail	4
Auswahl eines Amazon Lightsail-Preisplans	5
Installieren von WordPress	5
Wiederherstellung nach Ausfällen	6
Verbessern der Leistung und Kosteneffizienz	7
Beschleunigen der Bereitstellung von Inhalten	7
Statische Inhaltsausladung	8
Dynamischer Inhalt	8
Datenbank-Caching	10
Bytecode-Cache	10
Elastische Bereitstellung	12
Referenzarchitektur	12
Skalieren der Web-Tier	14
Zustandslose Web-Ebene	16
Gemeinsam genutzter Speicher (Amazon S3 und Amazon EFS)	16
Datenschicht (Amazon Aurora und Amazon ElastiCache)	17
WordPress High Availability von Bitnami in AWS – Quick Start	18
Fazit	21
Mitwirkende	22
Dokumentversionen	23
Anhang A: CloudFront-Konfiguration	24
Ursprünge und Verhaltensweisen	24
Erstellung der CloudFront-Verteilung	24
Anhang B: Installation und Konfiguration von Plug-Ins	28
AWS for WordPress-Plug-In	28
Installation und Konfiguration des Plug-Ins	28
Amazon CloudFront und AWS Certificate Manager	30
Ihre Inhalte übersetzen und vokalisieren	30

Podcasting mit Amazon Pollycast	33
Ihre Inhalte auf Amazon Alexa-Geräten lesen	34
Konfiguration statischer Inhalte	35
Erstellung von IAM-Benutzern	35
Erstellung des Amazon S3-Buckets	35
Erstellung eines statischen Ursprungs	37
Anhang C: Backup und Wiederherstellung	38
Anhang D: Bereitstellen neuer Plug-Ins und Themes	40
Hinweise	41

Bewährte Methoden für WordPress in AWS

Veröffentlichungsdatum: 19. Oktober 2021 ([Dokumentversionen](#))

Überblick

Dieses Whitepaper bietet Systemadministratoren eine Anleitung für den Einstieg in WordPress in Amazon Web Services (AWS) und veranschaulicht, wie die Wirtschaftlichkeit der Bereitstellung und das Endbenutzererlebnis verbessert werden können. Es enthält auch eine Referenzarchitektur, mit der gängige Anforderungen an Skalierbarkeit und Hochverfügbarkeit erfüllt werden.

Einführung

WordPress ist ein Open-Source-Blogging-Tool und Content Management System (CMS) basierend auf PHP und MySQL, das für die unterschiedlichsten Anwendungen genutzt wird – von persönlichen Blogs bis zu Websites mit hohen Zugriffszahlen.

Die erste Version von WordPress wurde 2003 veröffentlicht. Deshalb war das System nicht auf moderne elastische und skalierbare cloudbasierte Infrastrukturen ausgelegt. Durch die Arbeit der WordPress-Community und die Veröffentlichung verschiedener WordPress-Module werden die Funktionen der CMS-Lösung fortwährend ausgebaut. Heute ist es möglich, eine WordPress-Architektur zu erstellen, die viele Vorteile der AWS Cloud nutzt.

Einfache Bereitstellung

Für Blogs oder Websites mit geringem Datenverkehr ohne strenge Hochverfügbarkeitsanforderungen kann eine einfache Bereitstellung eines Einzelservers geeignet sein. Diese Bereitstellung ist nicht die widerstandsfähigste oder skalierbarste Architektur, aber sie ist die schnellste und wirtschaftlichste Methode, um Ihre Website in Betrieb zu nehmen.

Themen

- [Überlegungen](#)
- [Verfügbare Ansätze](#)
- [Amazon Lightsail](#)

Überlegungen

Diese Erläuterung beginnt mit einer einzigen Webserver-Bereitstellung. Es kann vorkommen, dass Sie eine größere Kapazität benötigen, zum Beispiel:

- Die virtuelle Maschine, auf der Ihre WordPress-Website bereitgestellt wird, ist ein Single-Point-of-Failure. Ihre Website ist aufgrund eines Problems mit dieser Instance nicht erreichbar.
- Die Skalierung von Ressourcen zur Leistungssteigerung ist mit einer „vertikalen Skalierung“ möglich, d. h., indem die virtuelle Maschine vergrößert wird, auf der Ihre WordPress-Website ausgeführt wird.

Verfügbare Ansätze

AWS bietet verschiedene Methoden zur Bereitstellung virtueller Maschinen. Es gibt drei Möglichkeiten, Ihre eigene WordPress-Website auf AWS zu hosten:

- Amazon Lightsail
- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Marketplace

[Amazon Lightsail](#) ist ein Service, mit dem Sie schnell einen virtuellen privaten Server (eine Lightsail-Instance) starten können, um eine WordPress-Website zu hosten. Lightsail ist die einfachste

Einstiegsmethode, wenn Sie keine hoch konfigurierbaren Instance-Typen oder keinen Zugriff auf erweiterte Netzwerkfunktionen benötigen.

[Amazon EC2](#) ist ein Webservice mit anpassbarer Rechenkapazität, damit Sie in Minutenschnelle einen virtuellen Server starten können. Amazon EC2 bietet mehr Konfigurations- und Verwaltungsoptionen als Lightsail, was in fortgeschritteneren Architekturen wünschenswert ist. Sie haben administrativen Zugriff auf Ihre EC2-Instances und können alle von Ihnen ausgewählten Softwarepakete, einschließlich WordPress, installieren.

[AWS Marketplace](#) ist ein Online-Shop, in dem Sie Software für AWS finden, kaufen und schnell bereitstellen können. Sie können die Bereitstellung mit 1-Click verwenden, um vorkonfigurierte WordPress-Images in Minutenschnelle direkt in Amazon EC2 in Ihrem eigenen AWS-Konto zu starten. Es gibt verschiedene Marketplace-Anbieter, die einsatzbereite WordPress-Instances anbieten.

In diesem Whitepaper wird die Lightsail-Option als empfohlene Implementierung für eine WordPress-Website mit Einzelserver beschrieben.

Amazon Lightsail

Lightsail ist die einfachste Möglichkeit für Entwickler, kleine Unternehmen, Studenten und andere Benutzer, die eine einfache Lösung für einen virtuellen privaten Server (VPS) benötigen, in AWS zu starten.

Der Service eliminiert für den Benutzer viele der komplexeren Elemente des Infrastrukturmanagements. Er ist daher ein idealer Ausgangspunkt, wenn Sie weniger Infrastrukturerfahrung haben oder wenn Sie sich auf den Betrieb Ihrer Website konzentrieren müssen und ein vereinfachtes Produkt für Ihre Anforderungen ausreicht.

Mit Amazon Lightsail können Sie Windows- oder Linux-/Unix-Betriebssysteme und beliebte Webanwendungen, einschließlich WordPress, auswählen und diese mit einem einzigen Klick aus vorkonfigurierten Vorlagen bereitstellen.

Bei steigenden Anforderungen können Sie die ursprünglichen Grenzen reibungslos erweitern und sich mit zusätzlichen AWS-Datenbanken, Objektspeichern, Caching und Content-Verteilungsservices verbinden.

Auswahl eines Amazon Lightsail-Preisplans

Ein [Lightsail-Plan](#) legt die monatlichen Kosten der Lightsail-Ressourcen fest, die Sie zum Hosten Ihrer WordPress-Website verwenden. Es gibt verschiedene Pläne, die eine Vielzahl von Anwendungsfällen abdecken und unterschiedliche Ebenen von CPU-Ressourcen, Arbeitsspeicher, SSD-Speicher (Solid-State-Laufwerk) und Datenübertragung bieten. Wenn Sie eine komplexe Website haben, benötigen Sie möglicherweise eine größere Instance mit mehr Ressourcen. Migrieren Sie Ihren Server [mithilfe der Webkonsole](#) oder wie in der [Amazon Lightsail CLI – Dokumentation](#) beschrieben zu einem erweiterten Plan.

Installieren von WordPress

Lightsail bietet Vorlagen für häufig verwendete Anwendungen wie WordPress. Diese Vorlage ist ein guter Ausgangspunkt für den Betrieb Ihrer eigenen WordPress-Website, da ein Großteil der benötigten Software vorinstalliert ist. Sie können zusätzliche Software installieren oder die Softwarekonfiguration anpassen, indem Sie das Terminal im Browser, Ihren eigenen SSH-Client oder die WordPress-Administrationsweboberfläche verwenden.

Amazon Lightsail arbeitet mit dem GoDaddys Pro Sites-Produkt zusammen, damit WordPress Kunden ihre Instances einfach und kostenlos verwalten können. Die virtuellen Lightsail WordPress-Server sind vorkonfiguriert und für eine schnelle Leistung und hohe Sicherheit optimiert, sodass Sie Ihre WordPress-Website im Handumdrehen in Betrieb nehmen können. Kunden, die mehrere WordPress-Instances ausführen, empfinden es als schwierig und zeitaufwendig, alle ihre Websites zu aktualisieren, zu warten und zu verwalten. Mit dieser Integration können Sie Ihre verschiedenen WordPress-Instances mit nur wenigen Klicks ganz einfach in Minutenschnelle verwalten.

Weitere Informationen zum Verwalten von WordPress auf Lightsail finden Sie unter [Erste Schritte mit WordPress in Ihrer Amazon Lightsail-Instance](#). Wenn Sie Ihre WordPress-Website fertig angepasst haben, sollten Sie einen Snapshot Ihrer Instance erstellen.

Mit einem [Snapshot](#) können Sie ein Backup-Image Ihrer Lightsail-Instance erstellen. Dabei handelt es sich um eine Kopie der Systemfestplatte, die mit der Konfiguration der ursprünglichen Maschine übereinstimmt (d. h. Speicher, CPU, Festplattengröße und Datenübertragungsgeschwindigkeit). Snapshots können verwendet werden, um nach einer fehlerhaften Bereitstellung oder einem fehlerhaften Upgrade eine zweifelsfrei funktionierende Konfiguration wiederherzustellen.

Mit diesem Snapshot können Sie Ihren Server bei Bedarf wiederherstellen und neue Instances mit denselben Anpassungen starten.

Wiederherstellung nach Ausfällen

Ein einzelner Webserver ist ein Single-Point-of-Failure (SPOF), weswegen Ihre Website-Daten gesichert werden müssen. Der zuvor beschriebene Snapshot-Mechanismus kann auch zu diesem Zweck verwendet werden. Für eine Wiederherstellung nach einem Ausfall können Sie eine neue Instance aus Ihrem letzten Snapshot wiederherstellen. Damit bei einer Wiederherstellung möglichst wenig Daten verloren gehen, müssen Ihre Snapshots so aktuell wie möglich sein.

Um das Risiko eines Datenverlusts zu minimieren, sollten Sie regelmäßig Snapshots erstellen. Sie können automatische Snapshots Ihrer Lightsail-Linux/Unix-Instances planen. Informationen zu den einzelnen Schritten finden Sie unter [Aktivieren oder Deaktivieren automatischer Snapshots für Instances oder Datenträger in Amazon Lightsail](#).

AWS empfiehlt, eine statische IP-Adresse zu verwenden: eine feste, öffentliche IP-Adresse, die für Ihr Lightsail-Konto reserviert ist. Wenn Sie Ihre Instance durch eine andere ersetzen müssen, können Sie die statische IP-Adresse der neuen Instance zuweisen. Auf diese Weise müssen Sie nicht alle externen Systeme neu konfigurieren (z. B. DNS-Datensätze), um immer auf eine neue IP-Adresse zu verweisen, wenn Sie Ihre Instance ersetzen möchten.

Verbessern der Leistung und Kosteneffizienz

Möglicherweise wird Ihre Einzelserver-Bereitstellung irgendwann nicht mehr Ihren Anforderungen gerecht. In diesem Fall sollten Sie Möglichkeiten zur Verbesserung der Leistung Ihrer Website in Betracht ziehen. Bevor Sie zu einer skalierbaren Multi-Server-Bereitstellung migrieren (weiter unten in diesem Dokument erläutert), können Sie die Leistung und Kosteneffizienz auf verschiedene Weise steigern. Diese bewährten Methoden sollten Sie auch dann befolgen, wenn Sie auf eine Multi-Server-Architektur umsteigen.

In den folgenden Abschnitten werden verschiedene Optionen vorgestellt, mit denen Sie die Leistung und Skalierbarkeit Ihrer WordPress-Website verbessern können. Manche können auf eine Einzelserver-Bereitstellung angewendet werden, während sich andere die Skalierbarkeit mehrerer Server zunutze machen. Für viele dieser Änderungen müssen Sie ein oder mehrere WordPress-Plug-Ins verwenden. Obwohl mehrere Optionen verfügbar sind, ist [W3 Total Cache](#) eine beliebte Wahl, weil es viele dieser Änderungen in einem einzigen Plug-In kombiniert.

Themen

- [Beschleunigen der Bereitstellung von Inhalten](#)
- [Datenbank-Caching](#)
- [Bytecode-Cache](#)

Beschleunigen der Bereitstellung von Inhalten

Alle WordPress-Websites müssen eine Mischung aus statischen und dynamischen Inhalten anbieten. Statischer Inhalt umfasst Bilder, JavaScript-Dateien oder Stylesheets. Dynamische Inhalte umfassen alle auf Serverseite mithilfe des WordPress-PHP-Codes generierte Inhalte, z. B. Elemente Ihrer Website, die aus der Datenbank generiert oder für die einzelnen Viewer personalisiert werden.

Ein wichtiger Aspekt des Endbenutzererlebnisses ist die Netzwerklatenz bei der Bereitstellung älterer Inhalte an Benutzer auf der ganzen Welt. Eine schnellere Bereitstellung früherer Inhalte verbessert das Endbenutzererlebnis, insbesondere bei geografisch auf der ganzen Welt verteilten Benutzern. Verwenden Sie dazu ein Content Delivery Network (CDN) wie Amazon CloudFront.

[Amazon CloudFront](#) ist ein Webservice, der eine einfache und kostengünstige Möglichkeit bietet, Inhalte mit geringer Latenz und hohen Datenübertragungsgeschwindigkeiten über mehrere Edge-Standorte auf der ganzen Welt zu verteilen. Viewer-Anforderungen werden automatisch an einen

geeigneten CloudFront-[Edge-Standort](#) weitergeleitet, um die Latenz zu reduzieren. Wenn der Inhalt zwischengespeichert werden kann (für einige Sekunden, Minuten oder sogar Tage) und bereits an einem bestimmten Edge-Standort gespeichert ist, stellt CloudFront ihn sofort bereit. Wenn der Inhalt nicht zwischengespeichert werden soll, abgelaufen ist oder sich derzeit nicht an diesem Edge-Standort befindet, ruft CloudFront Inhalte aus einer oder mehreren Informationsquellen ab, die in der CloudFront-Konfiguration als Ursprung (in diesem Fall die Lightsail-Instance) bezeichnet werden. Dieser Abruf erfolgt über optimierte Netzwerkverbindungen, die die Bereitstellung von Inhalten auf Ihrer Website beschleunigen. Dieses Modell verbessert nicht nur das Endbenutzererlebnis, sondern entlastet auch Ihre Ursprungsserver und kann zu erheblichen Kosteneinsparungen führen.

Statische Inhaltsausladung

Dazu gehören CSS-, JavaScript- und Bilddateien – entweder Dateien, die zu Ihren WordPress-Themes gehören, oder Mediendateien, die von den Inhaltsadministratoren hochgeladen wurden. Alle diese Dateien können mit einem Plug-In wie W3 Total Cache im Amazon Simple Storage Service (Amazon S3) gespeichert und den Benutzern mit hoher Skalierbarkeit und Verfügbarkeit zur Verfügung gestellt werden. [Amazon S3](#) bietet eine hoch skalierbare, zuverlässige und latenzarme Datenspeicherinfrastruktur zu geringen Kosten, auf die über REST-APIs zugegriffen werden kann. Amazon S3 speichert Ihre Objekte redundant nicht nur auf mehreren Geräten, sondern in mehreren Einrichtungen in einer AWS-Region, wodurch ein besonders hohes Maß an Zuverlässigkeit erreicht wird.

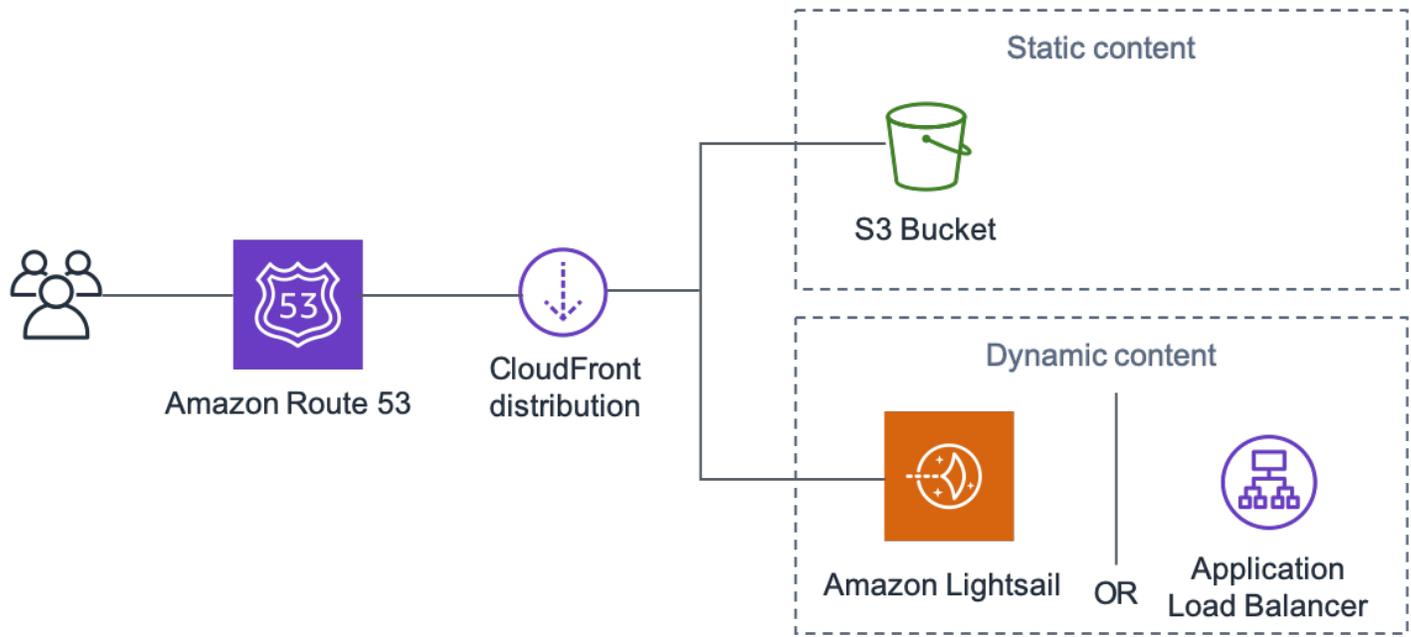
Dies hat den positiven Nebeneffekt, dass Ihre Lightsail-Instance von dieser Workload befreit wird, damit sie sich auf die dynamische Generierung von Inhalten konzentrieren kann. Dies reduziert die Serverlast und ist ein wichtiger Schritt zur Schaffung einer zustandslosen Architektur (eine Voraussetzung für die Implementierung der automatischen Skalierung).

Anschließend können Sie Amazon S3 als Ursprung für CloudFront konfigurieren, um die Bereitstellung dieser statischen Komponenten an Benutzer auf der ganzen Welt zu verbessern. Obwohl WordPress nicht in Amazon S3 und CloudFront vorintegriert ist, werden diese Services (zum Beispiel W3 Total Cache) von einer Vielzahl von Plug-Ins unterstützt.

Dynamischer Inhalt

Dynamische Inhalte beinhalten die Ausgabe von serverseitigen WordPress PHP-Skripten. Dynamische Inhalte können auch über CloudFront bereitgestellt werden, indem die WordPress-Website als Ursprung konfiguriert wird. Da dynamische Inhalte personalisierte Inhalte einschließen, müssen Sie CloudFront so konfigurieren, dass bestimmte HTTP-Cookies und HTTP-Header in

einer Anfrage an Ihren benutzerdefinierten Ursprungsserver weitergeleitet werden. CloudFront verwendet die weitergeleiteten Cookie-Werte als Teil des Schlüssels, der ein eindeutiges Objekt in seinem Cache identifiziert. Um die Effizienz des Cache zu maximieren, sollten Sie CloudFront so konfigurieren, dass nur die HTTP-Cookies und HTTP-Header weitergeleitet werden, die den Inhalt tatsächlich verändern (keine Cookies, die nur auf der Clientseite oder durch Anwendungen von Drittanbietern verwendet werden, z. B. für Webanalysen).



Bereitstellung der gesamten Website über Amazon CloudFront

Die vorhergehende Abbildung enthält zwei Ursprünge: einen für statischen Inhalt und einen weiteren für dynamischen Inhalt. Einzelheiten zur Implementierung finden Sie in [Anhang A: CloudFront-Konfiguration](#) und [Anhang B: Installation und Konfiguration von Plug-Ins](#).

CloudFront verwendet standardmäßige Cache-Control-Header, um zu ermitteln, ob und wie lange bestimmte HTTP-Antworten zwischengespeichert werden sollen. Diese Cache-Control-Header werden auch von Webbrowsern verwendet, um zu entscheiden, wann und wie lange Inhalte lokal zwischengespeichert werden sollen, um ein besseres Endbenutzererlebnis zu erzielen (z. B. wird eine bereits heruntergeladene .css-Datei nicht jedes Mal erneut heruntergeladen, wenn ein wiederkehrender Besucher eine Seite betrachtet). Sie können Cache-Control-Header auf Webserverebene konfigurieren (z. B. über .htaccess-Dateien oder Änderungen der httpd.conf-Datei) oder ein WordPress-Plug-In (z. B. W3 Total Cache) installieren, um zu bestimmen, wie diese Header für statische und dynamische Inhalte festgelegt werden.

Datenbank-Caching

Das Datenbank-Caching kann die Latenz erheblich reduzieren und den Durchsatz für leseintensive Anwendungsworkloads wie WordPress erhöhen. Die Anwendungsleistung wird verbessert, indem häufig aufgerufene Daten in Speichern für den Zugriff mit geringer Latenz gespeichert werden (z. B. die Ergebnisse von I/O-intensiven Datenbankabfragen). Wenn ein großer Prozentsatz der Abfragen aus dem Cache bereitgestellt wird, sinkt die Anzahl der Abfragen der Datenbank, wodurch die Kosten im Zusammenhang mit ihrer Ausführung reduziert werden.

Obwohl WordPress standardmäßig über begrenzte Caching-Funktionen verfügt, unterstützen eine Vielzahl von Plug-Ins die Integration in [Memcached](#), einem weit verbreiteten Caching-System für Speicherobjekte. Ein gutes Beispiel ist das W3 Total Cache-Plug-In.

In einfachen Szenarien installieren Sie Memcached auf Ihrem Webserver und erfassen das Ergebnis als neuen Snapshot. In diesem Fall sind Sie für die administrativen Aufgaben verantwortlich, die mit der Ausführung eines Caches verbunden sind.

Eine weitere Option besteht darin, einen verwalteten Service wie [Amazon ElastiCache](#) zu nutzen und diese operative Belastung zu vermeiden. ElastiCache erleichtert die Bereitstellung, den Betrieb und die Skalierung eines verteilten In-Memory-Cache in der Cloud. Informationen darüber, wie Sie eine Verbindung zu Ihren ElastiCache-Clusterknoten herstellen, finden Sie in der [Amazon ElastiCache-Dokumentation](#).

Wenn Sie Lightsail verwenden und privat auf ein ElastiCache-Cluster in Ihrem AWS-Konto zugreifen möchten, ist dies mit VPC-Peering möglich. Anweisungen zum Aktivieren von VPC-Peering finden Sie unter [Einrichtung eines Amazon VPC-Peerings, sodass es für AWS-Ressourcen außerhalb von Amazon Lightsail genutzt werden kann](#).

Bytecode-Cache

Wenn ein PHP-Skript ausgeführt wird, wird es jedes Mal analysiert und kompiliert. Durch die Verwendung eines PHP-Bytecode-Caches wird die Ausgabe der PHP-Kompilierung im RAM gespeichert, damit dasselbe Skript nicht immer wieder kompiliert werden muss. Dadurch wird der Aufwand für die Ausführung von PHP-Skripten reduziert, um die Leistung zu verbessern und die CPU-Anforderungen zu reduzieren.

Ein Bytecode-Cache kann auf jeder Lightsail-Instance installiert werden, die WordPress hostet, um die Last erheblich zu reduzieren. Für PHP 5.5 und höher empfiehlt AWS die Verwendung von [OPcache](#), einer gebündelten Erweiterung dieser PHP-Version.

Beachten Sie, dass OPcache in der Bitnami WordPress Lightsail-Vorlage standardmäßig aktiviert ist, sodass keine weiteren Aktionen erforderlich sind.

Elastische Bereitstellung

In manchen Szenarien reicht für Ihre Website eine Bereitstellung mit einem Server möglicherweise nicht aus. In diesen Fällen benötigen Sie eine skalierbare Architektur mit mehreren Servern.

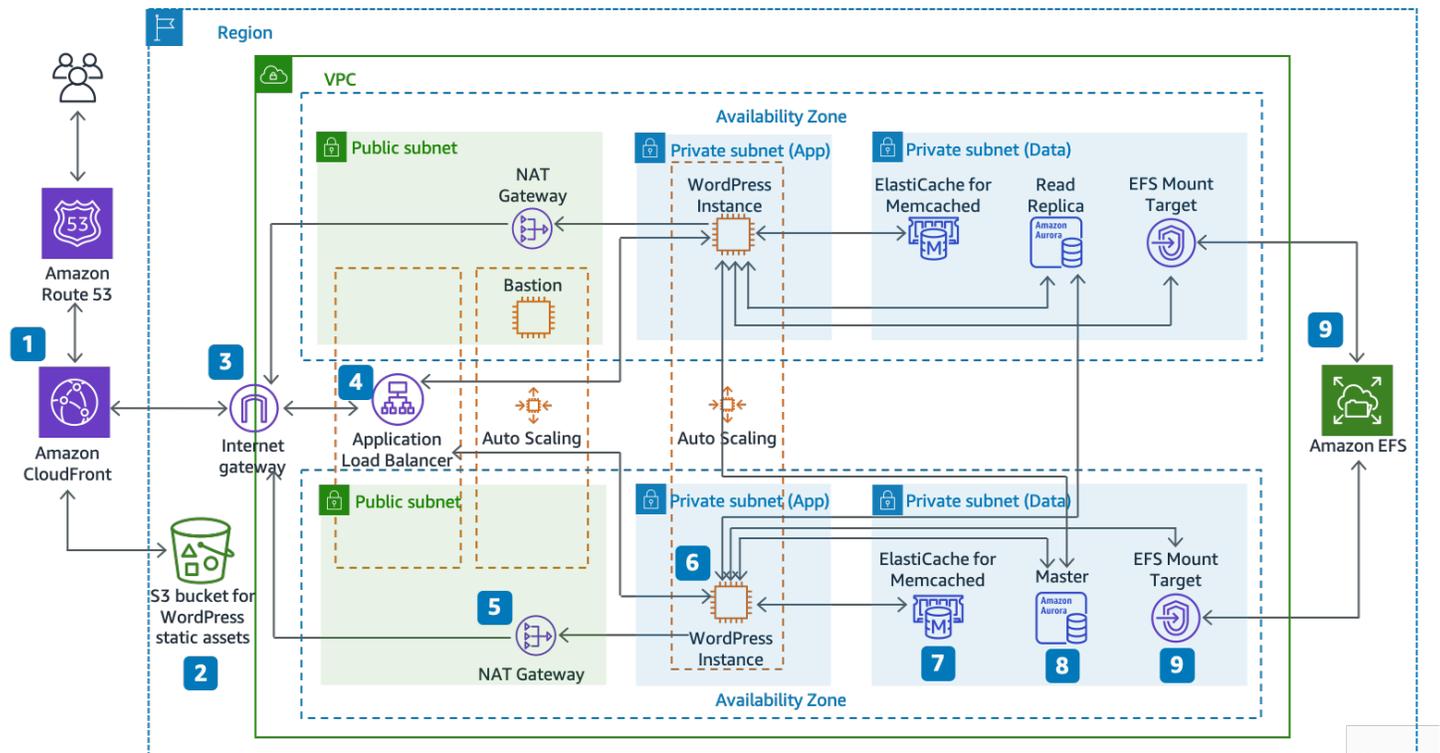
Themen

- [Referenzarchitektur](#)
- [Skalieren der Web-Tier](#)
- [Zustandslose Web-Ebene](#)
- [WordPress High Availability von Bitnami in AWS – Quick Start](#)

Referenzarchitektur

Das auf GitHub verfügbare Dokument zum [Hosten der Referenzarchitektur von WordPress in AWS](#) beschreibt bewährte Methoden für die Bereitstellung von WordPress in AWS und enthält eine Reihe von AWS CloudFormation-Vorlagen, mit denen Sie schnell einsatzbereit sind. Die folgende Architektur basiert auf dieser Referenzarchitektur. Im Rest dieses Abschnitts werden die Gründe für die Architekturwahl erörtert.

Das AMI im GitHub wurde im Juli 2021 von Amazon Linux1 auf Amazon Linux2 aktualisiert. Die Bereitstellungsvorlagen bei S3 wurden jedoch noch nicht geändert. Es wird empfohlen, die Vorlagen auf GitHub zu verwenden, wenn ein Problem bei der Bereitstellung der Referenzarchitektur mit Vorlagen bei S3 vorliegt.



Referenzarchitektur zum Hosten von WordPress in AWS

Architekturkomponenten

Die Referenzarchitektur veranschaulicht eine vollständige Bereitstellung nach bewährten Methoden von einer WordPress-Website in AWS.

- Zunächst wird ein Edge-Caching in Amazon CloudFront (1) durchgeführt, um Inhalte in Endbenutzernähe für eine schnellere Bereitstellung zu cachieren.
- CloudFront bezieht statischen Inhalt aus einem S3-Bucket (2) und dynamischen Inhalt aus einem Application Load Balancer (4) vor den Web-Instances.
- Die Web-Instances werden in einer Auto-Scaling-Gruppe von Amazon-EC2-Instances (6) ausgeführt.
- Ein ElastiCache-Cluster (7) speichert häufig abgefragte Daten im Cache, um die Antworten zu beschleunigen.

Die WordPress-Datenbank wird von einer Amazon Aurora MySQL-Instance (8) gehostet.

- Die WordPress EC2-Instances greifen über ein EFS-Mount-Target (9) in jeder Availability Zone auf freigegebene WordPress-Daten in einem Amazon EFS-Dateisystem zu.

- Ein Internet-Gateway (3) ermöglicht die Kommunikation zwischen Ressourcen in Ihrer VPC und dem Internet.
- NAT-Gateways (5) in jeder Availability Zone ermöglichen EC2-Instances in privaten Subnetzen (App und Daten) den Zugriff auf das Internet.

Innerhalb der Amazon VPC gibt es zwei Arten von Subnetzen: öffentlich (öffentliches Subnetz) und privat (App-Subnetz und Datensubnetz). In den öffentlichen Subnetzen bereitgestellte Ressourcen erhalten eine öffentliche IP-Adresse und sind im Internet öffentlich sichtbar. An dieser Stelle werden der Application Load Balancer (4) und ein Bastion-Host für die Verwaltung bereitgestellt. Ressourcen, die in den privaten Subnetzen bereitgestellt werden, erhalten nur eine private IP-Adresse und sind daher im Internet nicht öffentlich sichtbar, wodurch die Sicherheit dieser Ressourcen verbessert wird. Die WordPress-Webserver-Instances (6), ElastiCache-Cluster-Instances (7), Aurora MySQL-Datenbank-Instances (8) und EFS-Mount-Ziele (9) werden alle in privaten Subnetzen bereitgestellt.

Im Rest dieses Abschnitts werden diese Überlegungen ausführlicher erörtert.

Skalieren der Web-Tier

Um Ihre Einzelserver-Architektur zu einer skalierbaren Multi-Server-Architektur zu erweitern, benötigen Sie fünf Schlüsselkomponenten:

- Amazon-EC2-Instances
- Amazon Machine Images (AMIs)
- Lastenverteilungen
- Automatische Skalierung
- Zustandsprüfungen

AWS bietet eine Vielzahl von EC2-Instance-Typen, damit Sie die beste Serverkonfiguration zur Steigerung der Leistung und Senkung der Kosten auswählen können. Im Allgemeinen kann der rechenoptimierte Instance-Typ (z. B. C4) eine gute Wahl für einen WordPress-Webserver sein. Sie können Ihre Instances in mehreren Availability Zones in einer AWS-Region bereitstellen, um die Zuverlässigkeit der Gesamtarchitektur zu erhöhen.

Da Sie die vollständige Kontrolle über Ihre EC2-Instance haben, können Sie sich mit Root-Zugriff anmelden, um alle erforderlichen Softwarekomponenten für den Betrieb einer WordPress-Website zu installieren und zu konfigurieren. Wenn Sie fertig sind, können Sie diese Konfiguration als AMI

speichern, mit dem Sie neue Instances mit allen von Ihnen vorgenommenen Anpassungen starten können.

Zur Verteilung von Endbenutzeranfragen an mehrere Webserverknoten benötigen Sie eine Lastenverteilungslösung. AWS stellt diese Funktion mit [Elastic Load Balancing](#) bereit, einem hochverfügbaren Service, der den Datenverkehr auf mehrere EC2-Instances verteilt. Da Ihre Website Inhalte über HTTP oder HTTPS für Ihre Benutzer bereitstellt, empfehlen wir Ihnen die Verwendung des Application Load Balancer. Dabei handelt es sich um eine Lastenverteilung auf Anwendungsebene mit Inhaltsrouting und der Möglichkeit, bei Bedarf mehrere WordPress-Websites auf verschiedenen Domänen auszuführen.

Elastic Load Balancing unterstützt die Verteilung von Anfragen in mehreren Availability Zones innerhalb einer AWS-Region. Sie können eine Zustandsprüfung auch so konfigurieren, dass der Application Load Balancer automatisch den Versand von Datenverkehr an einzelne Instances einstellt, die ausgefallen sind (z. B. aufgrund eines Hardwareproblems oder eines Software-Absturzes). AWS empfiehlt, die WordPress-Admin-Anmeldeseite (`/wp-login.php`) für die Zustandsprüfung zu verwenden, da auf dieser Seite bestätigt wird, dass der Webserver ausgeführt wird und für die korrekte Bereitstellung von PHP-Dateien konfiguriert ist.

Sie können eine Seite für benutzerdefinierte Zustandsprüfungen erstellen, die andere abhängige Ressourcen wie Datenbank- und Cache-Ressourcen überprüft. Weitere Informationen finden Sie unter [Zustandsprüfungen für Ihre Zielgruppen](#) im Leitfaden zum Application Load Balancer.

Ein wesentliches Merkmal der AWS Cloud ist Elastizität. Sie können bei Bedarf mehr oder weniger Rechenkapazität (z. B. Webserver) launchen. [AWS Auto Scaling](#) ist ein AWS-Service, mit dem Sie diese Bereitstellung automatisieren können, um Ihre Amazon-EC2-Kapazität ohne manuelle Eingriffe Ihren Anforderungen entsprechend auf- oder abzuskalieren. Sie können AWS Auto Scaling so konfigurieren, dass die Anzahl der verwendeten EC2-Instances bei Nachfragespitzen nahtlos ansteigt, um die Leistung aufrechtzuerhalten, und automatisch abnimmt, wenn der Datenverkehr nachlässt, um Kosten zu minimieren.

Elastic Load Balancing unterstützt auch das dynamische Hinzufügen und Entfernen von Amazon-EC2-Hosts aus der Rotation der Lastenverteilung. Elastic Load Balancing erhöht und verringert auch dynamisch die Lastenverteilungskapazität, um sich ohne manuelle Eingriffe der Datenverkehrslast anzupassen.

Zustandslose Web-Ebene

Um mehrere Webserver in einer Konfiguration mit automatischer Skalierung nutzen zu können, muss Ihre Web-Ebene zustandslos sein. Eine zustandslose Anwendung muss frühere Interaktionen nicht kennen und speichert auch keine Sitzungsinformationen. Im Fall von WordPress bedeutet dies, dass alle Endbenutzer dieselbe Antwort erhalten, egal welcher Webserver ihre Anfrage bearbeitet hat. Eine zustandslose Anwendung kann horizontal skaliert werden, da jede Anforderung von den verfügbaren Rechenressourcen (d. h. Webserver-Instances) bearbeitet werden kann. Wenn diese Kapazität nicht mehr benötigt wird, können die einzelnen Ressourcen sicher beendet werden (nachdem die laufenden Aufgaben beendet wurden). Diese Ressourcen erkennen nicht immer ihre Peers – es bedarf lediglich einer Möglichkeit, die Workload an sie zu verteilen.

Bei der Speicherung von Benutzersitzungsdaten ist der WordPress-Kern völlig zustandslos, da er auf Cookies basiert, die im Client-Webbrowser gespeichert werden. Der Sitzungsspeicher stellt kein Problem dar, sofern Sie keinen benutzerdefinierten Code (z. B. ein WordPress-Plug-In) installiert haben, der auf nativen PHP-Sitzungen basiert.

Ursprünglich wurde WordPress für die Ausführung auf einem einzigen Server entwickelt. Deswegen werden einige Daten im lokalen Dateisystem des Servers gespeichert. Wenn WordPress in einer Multi-Server-Konfiguration ausgeführt wird, entstehen problematische Inkonsistenzen zwischen den Webservern. Wenn ein Benutzer beispielsweise ein neues Image hochlädt, wird es nur auf einem der Server gespeichert.

Diese Tatsache verdeutlicht, warum wir die laufende Standardkonfiguration von WordPress verbessern müssen, um wichtige Daten in den freigegebenen Speicher zu verschieben. Die empfohlene Architektur verfügt über eine Datenbank, die als separate Ebene außerhalb des Webserver fungiert und gemeinsam genutzten Speicher verwendet, um Benutzer-Uploads, Themes und Plug-Ins zu speichern.

Gemeinsam genutzter Speicher (Amazon S3 und Amazon EFS)

WordPress speichert Benutzer-Uploads standardmäßig im lokalen Dateisystem und ist daher nicht zustandslos. Daher müssen wir die WordPress-Installation und alle Benutzeranpassungen (wie Konfiguration, Plug-Ins, Themes und vom Benutzer generierte Uploads) in eine gemeinsam genutzte Datenplattform verschieben, um die Webserver zu entlasten und die Web-Ebene zustandslos zu machen.

[Amazon Elastic File System](#) (Amazon EFS) bietet skalierbare Netzwerkdateisysteme zur Verwendung mit EC2-Instances. Amazon EFS-Dateisysteme werden über eine unbegrenzte Anzahl von

Speicherservern verteilt. Dadurch können Dateisysteme elastisch skaliert und ein hochgradig paralleler Zugriff von EC2-Instances auf Ihre Daten eingerichtet werden. Mit dem verteilten Design von Amazon EFS werden die Engpässe und Einschränkungen von herkömmlichen Dateiservern umgangen.

Wenn Sie das gesamte WordPress-Installationsverzeichnis auf ein EFS-Dateisystem verschieben und es beim Booten in all Ihre EC2-Instances einbinden, werden Ihre WordPress-Site und alle zugehörigen Daten automatisch in einem verteilten Dateisystem gespeichert, das nicht von einer EC2-Instance abhängig ist, wodurch Ihr Web-Tier vollständig zustandslos wird. Der Vorteil dieser Architektur besteht darin, dass Sie nicht bei jedem neuen Start der Instance Plug-Ins und Themes installieren müssen. Außerdem werden die Installation und Wiederherstellung von WordPress-Instances erheblich beschleunigt. Es ist auch einfacher, Änderungen an Plug-Ins und Themes in WordPress bereitzustellen, wie im Abschnitt [Überlegungen zur Bereitstellung](#) dieses Dokuments beschrieben.

Um eine optimale Leistung Ihrer Website zu gewährleisten, wenn sie von einem EFS-Dateisystem ausgeführt wird, überprüfen Sie die empfohlenen Konfigurationseinstellungen für Amazon EFS und OPCache in der [AWS-Referenzarchitektur für WordPress](#).

Sie haben auch die Möglichkeit, alle statischen Komponenten wie Bild-, CSS- und JavaScript-Dateien in einen S3-Bucket mit CloudFront-Caching zu verlagern. Die Vorgehensweise in einer Multi-Server-Architektur ist genau die gleiche wie bei einer Einzelserver-Architektur, wie im Abschnitt [Statischer Inhalt](#) dieses Whitepapers erläutert. Die Vorteile sind dieselben wie bei der Einzelserver-Architektur: Sie können die aufwendige Bereitstellung Ihrer statischen Komponenten auf Amazon S3 und CloudFront verlagern, damit sich Ihre Webserver auf die Generierung dynamischer Inhalte konzentrieren und mehr Benutzeranfragen pro Webserver bearbeiten können.

Datenschicht (Amazon Aurora und Amazon ElastiCache)

Da die WordPress-Installation auf einem verteilten, skalierbaren, gemeinsam genutzten Netzwerkdateisystem gespeichert ist und statische Komponenten über Amazon S3 bereitgestellt werden, können Sie sich ganz der verbleibenden zustandsbehafteten Komponente widmen: der Datenbank. Wie bei der Speicherebene sollte die Datenbank nicht nur auf einen einzelnen Server angewiesen sein, damit sie nicht auf einem der Webserver gehostet werden kann. Hosten Sie stattdessen die WordPress-Datenbank auf Amazon Aurora.

[Amazon Aurora](#) ist eine MySQL- und PostgreSQL-kompatible relationale Datenbank für die Cloud, die die Leistung und Verfügbarkeit einer hochwertigen kommerziellen Datenbank mit der

Unkompliziertheit und Wirtschaftlichkeit einer Open Source-Datenbank verbindet. Aurora MySQL erhöht die Leistung und Verfügbarkeit von MySQL, indem die Datenbank-Engine weitgehend in ein speziell entwickeltes verteiltes SSD-gestütztes Speichersystem integriert wird. Das fehlertolerante und selbstreparierende Aurora SQL repliziert sechs Kopien Ihrer Daten über drei Availability Zones. Es ist für eine Verfügbarkeit von mehr als 99,99 % ausgelegt und sichert Ihre Daten kontinuierlich in Amazon S3. Amazon Aurora erkennt automatisch Datenbankausfälle und startet neu, ohne dass eine Wiederherstellung nach einem Ausfall oder eine Neuerstellung des Datenbank-Caches erforderlich ist.

Amazon Aurora bietet eine Reihe von [Instance-Typen](#) für verschiedene Anwendungsprofile, einschließlich speicheroptimierter und spitzenleistungsfähiger Instances. Zur Verbesserung der Leistung Ihrer Datenbank können Sie einen großen Instance-Typ auswählen, um mehr CPU- und Speicherressourcen bereitzustellen.

Amazon Aurora bearbeitet automatisch das Failover zwischen der primären Instance und [Aurora Replicas](#), damit Ihre Anwendungen den Datenbankbetrieb ohne manuelle administrative Eingriffe so schnell wie möglich fortsetzen können. Ein Failover dauert in der Regel weniger als 30 Sekunden.

Nach der Erstellung von mindestens einem Aurora Replica stellen Sie über den Cluster-Endpunkt eine Verbindung zu Ihrer primären Instance her, damit Ihre Anwendung bei einem Ausfall der primären Instance automatisch ein Failover durchführt. Sie können bis zu 15 Lesereplikate mit niedriger Latenz in drei Availability Zones erstellen.

Wenn Sie Ihre Datenbank skalieren, müssen Sie auch Ihren Datenbank-Cache skalieren. Wie bereits im Abschnitt [Datenbank-Caching](#) erläutert, bietet ElastiCache für eine höhere Verfügbarkeit Funktionen zum Skalieren des Caches über mehrere Knoten in einem ElastiCache-Cluster und über mehrere Availability Zones in einer Region. Stellen Sie beim Skalieren Ihres ElastiCache-Clusters sicher, dass Sie Ihr Caching-Plug-In so konfigurieren, dass es eine Verbindung über den Konfigurationsendpunkt herstellt, damit WordPress beim Hinzufügen neuer Clusterknoten verwenden kann und beim Entfernen alte Clusterknoten nicht mehr verwenden kann. Außerdem müssen Sie Ihre Webserver zur Verwendung des [ElastiCache-Cluster-Client für PHP](#) einrichten und Ihr AMI aktualisieren, um diese Änderung zu speichern.

WordPress High Availability von Bitnami in AWS – Quick Start

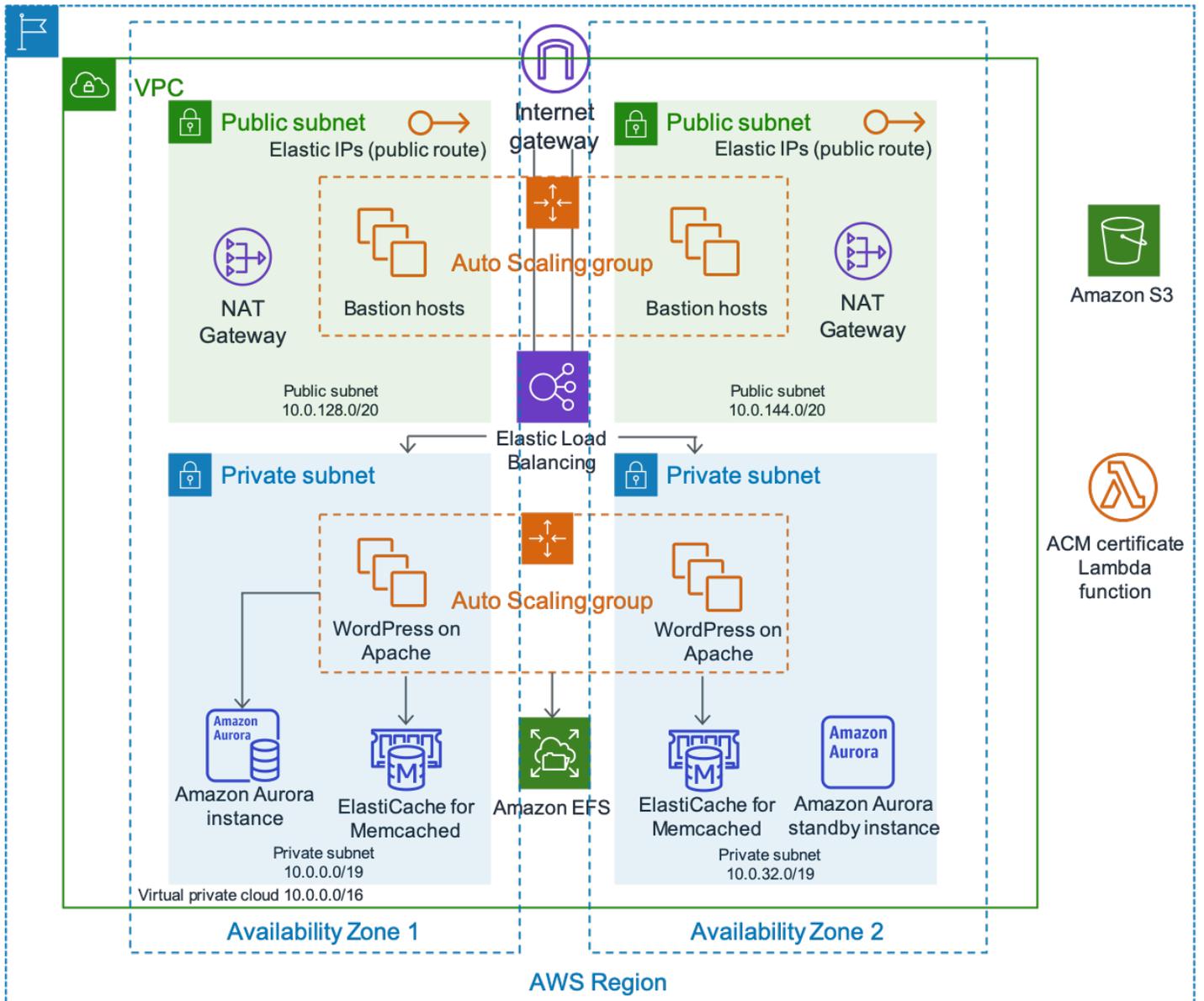
Quick Starts werden von AWS-Lösungsarchitekten und -Partnern entwickelt, um Sie bei der Bereitstellung beliebiger Technologien in AWS zu unterstützen, die auf bewährten Methoden von AWS für eine erhöhte Sicherheit und Hochverfügbarkeit basieren. Diese Beschleuniger

reduzieren hunderte von manuellen Vorgängen auf nur wenige Schritte. Dadurch können Sie Ihre Produktionsumgebung schnell aufbauen und umgehend verwenden. Jeder Quick Start enthält AWS CloudFormation-Vorlagen zur Automatisierung der Bereitstellung sowie eine Anleitung, in der die Architektur beschrieben wird und schrittweise Bereitstellungsanweisungen enthalten sind.

[WordPress High Availability von Bitnami in AWS – Quick Start](#) richtet die folgende konfigurierbare Umgebung auf AWS ein:

- Eine hoch verfügbare Architektur, die sich über zwei Availability Zones erstreckt.*
- Eine gemäß den bewährten Methoden von AWS mit öffentlichen und privaten Subnetzen konfigurierte Virtual Private Cloud (VPC). Dadurch wird die für Ihre Einrichtung nötige Netzwerkinfrastruktur bereitgestellt.*
- Ein Internet-Gateway für den Zugang zum Internet. Dieses Gateway wird von den Bastion-Hosts für eingehenden und ausgehenden Datenverkehr genutzt.*
- In den öffentlichen Subnetzen ermöglichen verwaltete NAT-Gateways den ausgehenden Internetzugriff auf Ressourcen in den privaten Subnetzen.*
- In den öffentlichen Subnetzen ermöglicht ein Linux Bastion Host in einer Auto-Scaling-Gruppe von AWS den eingehenden SSH-Zugriff (Secure Shell) auf EC2-Instances in öffentlichen und privaten Subnetzen.*
- Elastic Load Balancing (ELB) zum Verteilen der HTTP- und HTTPS-Anfragen über mehrere WordPress-Instances hinweg.
- In den privaten Subnetzen EC2-Instances, die die WordPress-Anwendung auf Apache hosten. Diese Instances werden in einer Auto-Scaling-Gruppe bereitgestellt, um eine hohe Verfügbarkeit zu gewährleisten.
- In den privaten Subnetzen durch Amazon Relational Database Service (Amazon RDS) verwaltete Amazon Aurora DB-Instances.
- In den privaten Subnetzen Amazon Elastic File System (Amazon EFS), um Komponenten (wie Plug-ins, Themes und Images) über WordPress-Instanzen hinweg zu teilen.
- In den privaten Subnetzen wird Amazon ElastiCache for Memcached zum Zwischenspeichern von Datenbankabfragen verwendet.

* Die Vorlage, mit der Sie den Quick Start in einer vorhandenen VPC bereitstellen, überspringt die mit Sternchen markierten Aufgaben und fordert Sie zur Eingabe der vorhandenen VPC-Konfiguration auf.



WordPress Hochverfügbarkeitsarchitektur von Bitnami

Eine ausführliche Beschreibung der Bereitstellung von WordPress High Availability von Bitnami in AWS würde den Rahmen dieses Dokuments sprengen. Informationen zur Konfiguration und Optionen finden Sie unter [WordPress High Availability von Bitnami in AWS](#).

Fazit

AWS bietet viele Architekturoptionen zur Ausführung von WordPress. Die einfachste Option ist eine Installation mit einem Server für Websites mit geringem Datenverkehr. Für fortgeschrittenere Websites können Site-Administratoren weitere Optionen hinzufügen, die die Verfügbarkeit und Skalierbarkeit schrittweise verbessern. Administratoren können die Funktionen auswählen, die ihren Anforderungen und ihrem Budget am ehesten entsprechen.

Mitwirkende

An diesem Dokument haben folgende Personen mitgewirkt:

- Paul Lewis, Lösungsarchitekt, Amazon Web Services
- Ronan Guilfoyle, Lösungsarchitekt, Amazon Web Services
- Andreas Chatzakis, Leitender Lösungsarchitekt, Amazon Web Services
- Jibril Touzi, Technischer Account-Manager, Amazon Web Services
- Hakmin Kim, Lösungsarchitekt für Migrationspartner, Amazon Web Services

Dokumentversionen

Abonnieren Sie den RSS-Feed, um über Aktualisierungen des Whitepapers benachrichtigt zu werden.

Update-Historie-Änderung	Update-Historie-Beschreibung	Update-Historie-Datum
Whitepaper aktualisiert	Aktualisiert, um die Referenzarchitektur und das AWS for WordPress-Plug-In zu ändern.	19. Oktober 2021
Whitepaper aktualisiert	Mit neuen Bereitstellungsansätzen und dem AWS for WordPress-Plug-In aktualisiert.	30. Oktober 2019
Whitepaper aktualisiert	Aktualisiert, um die Produktbootschaft von Amazon Aurora zu verdeutlichen.	1. Februar 2018
Whitepaper aktualisiert	Aktualisiert, um die AWS-Services einzuschließen, die seit der ersten Veröffentlichung gelauncht wurden.	1. Dezember 2017
Erste Veröffentlichung	Erstmals veröffentlicht.	1. Dezember 2014

Anhang A: CloudFront-Konfiguration

Um eine optimale Leistung und Effizienz bei der Verwendung von Amazon CloudFront mit Ihrer WordPress-Website zu erzielen, muss die Website für die verschiedenen bereitgestellten Inhalte korrekt konfiguriert werden.

Themen

- [Ursprünge und Verhaltensweisen](#)
- [Erstellung der CloudFront-Verteilung](#)

Ursprünge und Verhaltensweisen

Ein [Ursprung](#) ist ein Ort, an den CloudFront Anforderungen für Inhalte sendet, die es über die Edge-Standorte verteilt. Abhängig von Ihrer Implementierung können Sie einen oder zwei Ursprünge haben. Ein Ursprung für dynamische Inhalte (die Lightsail-Instance in der [Bereitstellungsoption für einen Server](#) oder der Application Load Balancer in der [elastischen Bereitstellungsoption](#)) mit einem benutzerdefinierten Ursprung. Möglicherweise haben Sie einen zweiten Ursprung, an den Sie CloudFront für Ihre statischen Inhalte weiterleiten können. In der vorhergehenden [Referenzarchitektur](#) war dies ein S3-Bucket. Wenn Sie Amazon S3 als Ursprung für Ihre Verteilung verwenden, benötigen Sie eine [Bucket-Richtlinie](#), um den Inhalt öffentlich zugänglich zu machen.

Mithilfe von [Verhaltensweisen](#) können Sie Regeln bestimmen, die festlegen, wie CloudFront Ihre Inhalte zwischenspeichert und wie effektiv der Cache ist. Mit Verhaltensweisen können Sie das Protokoll und die HTTP-Methoden steuern, über die Ihre Website abrufbar ist. Mit Ihnen können Sie auch steuern, ob Sie HTTP-Header, Cookies oder Abfragezeichenfolgen an Ihr Backend übergeben (und falls ja, welche). Verhaltensweisen gelten für bestimmte URL-Pfadmuster.

Erstellung der CloudFront-Verteilung

Erstellen Sie eine CloudFront-Webverteilung, indem Sie der Verteilung folgen. Der automatisch erstellte Standardursprung und die automatisch erstellte Verhaltensweise werden für dynamische Inhalte verwendet. Erstellen Sie vier zusätzliche Verhaltensweisen, um die Behandlung statischer und dynamischer Anforderungen weiter anzupassen. In der folgenden Tabelle sind die Konfigurationseigenschaften für die fünf Verhaltensweisen zusammengefasst. Sie können diese manuelle Konfiguration auch überspringen und das AWS for WordPress-Plug-In verwenden, das

in [Anhang B: Installation und Konfiguration von Plug-Ins](#) behandelt wird. Dies ist die einfachste Methode, CloudFront zu konfigurieren, um Ihre WordPress-Site zu beschleunigen.

Tabelle 1: Zusammenfassung der Konfigurationseigenschaften für CloudFront-Verhaltensweisen

Eigenschaft	Statisch	Dynamisch (Admin)	Dynamisch (Frontend)
Pfade (Verhaltensweisen)	wp-content/* wp-includes/*	wp-admin/* wp-login.php	Standard (*)
Protokolle	HTTP und HTTPS	Umleitung zu HTTPS	HTTP und HTTPS
HTTP-Methoden	GET, HEAD	ALL	ALL
HTTP-Header	NONE	ALL	Host CloudFront-Forwarded-Proto CloudFront-Is-Mobile-Viewer CloudFront-Is-Tablet-Viewer CloudFront-Is-Desktop-Viewer
Cookies	NONE	ALL	Kommentar_* Wordpress_* WP-Einstellungen-*
Abfragezeichenfolgen	JA (Ungültigkeit)	JA	JA

Für das Standardverhalten empfiehlt AWS die folgende Konfiguration:

- Erlauben Sie die Ursprungsprotokollrichtlinie im Match Viewer, damit CloudFront, wenn sich Betrachter über HTTPS mit CloudFront verbinden, auch über HTTPS mit Ihrem Ursprung verbindet und so eine End-to-End-Verschlüsselung erreicht. Beachten Sie, dass Sie dafür ein vertrauenswürdigen SSL-Zertifikat auf der Lastenverteilung installieren müssen. Ausführlichere Informationen erhalten Sie unter [Erzwingen der Verwendung von HTTPS für die Kommunikation zwischen CloudFront und Ihrem benutzerdefinierten Ursprung](#).
- Erlauben Sie alle HTTP-Methoden, da die dynamischen Bereiche der Website sowohl GET- als auch POST-Anforderungen erfordern (z. B. zur Unterstützung von POST für Formulare zum Einreichen von Kommentaren).
- Leiten Sie nur die Cookies weiter, die die WordPress-Ausgabe verändern; zum Beispiel `>wordpress_*`, `wp-settings-*` und `comment_*`. Sie müssen diese Liste erweitern, wenn Sie Plug-Ins installiert haben, die von anderen Cookies abhängen, die nicht in der Liste aufgeführt sind.
- Leiten Sie nur die HTTP-Header weiter, die sich auf die Ausgabe von WordPress auswirken, zum Beispiel `Host`, `CloudFront-Forwarded-Proto`, `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer` und `CloudFront-is-Tablet-Viewer`:
 - `Host` ermöglicht das Hosten mehrerer WordPress-Websites auf demselben Ursprung.
 - `CloudFront-Forwarded-Proto` ermöglicht das Zwischenspeichern verschiedener Versionen von Seiten, je nachdem, ob auf sie über HTTP oder HTTPS zugegriffen wird.
 - `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer` und `CloudFront-is-Tablet-Viewer` ermöglichen Ihnen, die Ausgabe Ihrer Themes basierend auf dem Gerätetyp des Endbenutzers anzupassen.
- Leiten Sie alle Abfragezeichenfolgen basierend auf ihren Werten an den Cache weiter, da WordPress auf sie angewiesen ist. Sie können auch verwendet werden, um zwischengespeicherte Objekte ungültig zu machen.

Wenn Sie Ihre Website unter einem benutzerdefinierten Domännennamen anbieten möchten (d. h. nicht `*.cloudfront.net`), geben Sie in den Verteilungseinstellungen unter Alternative Domainnamen die entsprechenden URIs ein. In diesem Fall benötigen Sie auch ein SSL-Zertifikat für Ihren benutzerdefinierten Domännennamen. Sie können SSL-Zertifikate über den AWS Certificate Manager [anfordern](#) und sie für eine CloudFront-Verteilung konfigurieren.

Erstellen Sie nun zwei weitere Cache-Verhaltensweisen für dynamische Inhalte: eine für die Anmeldeseite (Pfadmuster: `wp-login.php`) und eine für das Admin-Dashboard (Pfadmuster: `wp-admin/*`). Diese beiden Verhaltensweisen haben exakt dieselben Einstellungen, und zwar:

- Eine Viewer-Protokollrichtlinie nur für HTTPS erzwingen.
- Alle HTTP-Methoden erzwingen.
- Der Cache basiert auf allen HTTP-Headern.
- Alle Cookies weiterleiten.
- Weiterleitung und Caching basieren auf allen Abfragezeichenfolgen.

Der Grund für diese Konfiguration ist, dass dieser Bereich der Website hochgradig personalisiert ist und in der Regel nur wenige Benutzer hat, sodass die Effizienz des Cachings kein Hauptanliegen ist. Im Vordergrund steht die Einfachheit der Konfiguration, um eine größtmögliche Kompatibilität mit allen installierten Plug-Ins zu gewährleisten, indem alle Cookies und Header an den Ursprung übergeben werden.

Das [AWS for WordPress-Plug-In](#), das in Anhang B behandelt wird, erstellt automatisch eine CloudFront-Verteilung, die der vorherigen Konfiguration entspricht.

WordPress speichert standardmäßig alles lokal auf dem Webserver, der als Blockspeicher (Amazon EBS) für die [Bereitstellung auf einem Server](#) und als Dateispeicher (Amazon EFS) für die [elastische Bereitstellung](#) fungiert. Neben der Reduzierung der Speicher- und Datenübertragungskosten bietet die Verlagerung statischer Komponenten auf Amazon S3 eine erhöhte Skalierbarkeit, Datenverfügbarkeit, Sicherheit und Leistung. Es gibt mehrere Plug-Ins, die das Verschieben statischer Inhalte auf Amazon S3 vereinfachen. Eines davon ist [W3 Total Cache](#), das auch in [Anhang B: Installation und Konfiguration von Plug-Ins](#) behandelt wird.

Anhang B: Installation und Konfiguration von Plug-Ins

Themen

- [AWS for WordPress-Plug-In](#)
- [Konfiguration statischer Inhalte](#)

AWS for WordPress-Plug-In

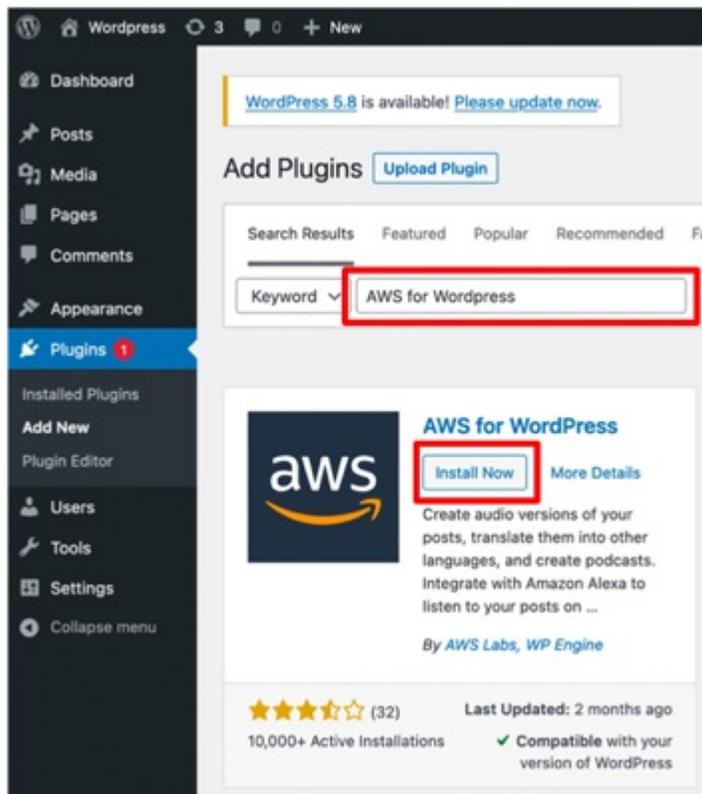
Das AWS for WordPress-Plug-In ist das einzige WordPress-Plug-In, das von AWS geschrieben und aktiv verwaltet wird. Mit ihm können Kunden [Amazon CloudFront](#) und [AWS Certificate Manager \(ACM\)](#) einfach für WordPress-Websites konfigurieren, um die Leistung und Sicherheit zu verbessern. Das Plug-In verwendet [Amazon Machine Learning](#)-Services, um Inhalte in eine oder mehrere Sprachen zu übersetzen, Audioversionen aller Übersetzungen zu produzieren und WordPress-Websites über Amazon Alexa-Geräte zu lesen.

Das Plug-In ist bereits in [WordPress High Availability by Bitnami auf AWS – Quick Start](#) installiert.

Installation und Konfiguration des Plug-Ins

So installieren Sie das Plug-In:

1. Um das AWS for WordPress-Plug-In zu verwenden, müssen Sie einen IAM-Benutzer für das Plug-In erstellen. Ein IAM-Benutzer ist eine Person oder Anwendung unter einem AWS-Konto, die berechtigt ist, API-Aufrufe an AWS-Services zu tätigen.
2. Sie benötigen eine IAM-Rolle ([AWS Identity and Access Management](#)) oder einen IAM-Benutzer, um die Authentifizierung und Autorisierung für Ihr AWS-Konto zu steuern. Um zu verhindern, dass nicht autorisierte Benutzer diese Berechtigungen erhalten, schützen Sie die Anmeldeinformationen des IAM-Benutzers. Behandeln Sie den geheimen Zugriffsschlüssel wie ein Passwort; speichern Sie ihn an einem sicheren Ort und teilen Sie ihn mit niemandem. [Rotieren Sie den Zugriffsschlüssel regelmäßig](#) wie ein Passwort. Wenn der geheime Zugriffsschlüssel versehentlich bekannt wurde, [löschen Sie ihn sofort](#). Anschließend können Sie einen neuen Zugriffsschlüssel erstellen, der mit dem AWS for WordPress-Plug-In verwendet werden kann.
3. Suchen Sie im Plug-Ins-Menü des WordPress-Adminbereichs nach AWS for WordPress und wählen Sie Jetzt installieren.



4. Wenn die Installation des Plug-Ins nicht funktioniert, liegt möglicherweise ein Problem mit der Benutzerberechtigung vor. Verbinden Sie sich mit dem WordPress-Webserver und befolgen Sie die nachstehenden Anweisungen, um das Problem zu beheben.
- Öffnen Sie die `wp-config.php`-Datei im WordPress-Installationsverzeichnis und schreiben Sie den folgenden Code an das Ende der `wp-config.php`-Datei:

```
define('FS_METHOD', 'direct');
```

- Starten Sie den folgenden Befehl, um Schreibberechtigungen zu erteilen:

```
chmod 777 <WordPress install directory>/wp-content
```

Warnung: Die Erhaltung der Schreibberechtigung als 777 ist mit Risiken verbunden. Wenn die Berechtigung als 777 beibehalten wird, können alle Benutzer diesen Ordner bearbeiten oder löschen. Ändern Sie die Schreibberechtigung in 755 oder niedriger, wenn Sie mit dem Plug-In fertig sind.

- Wenn die Referenzarchitektur verwendet wird, lautet das WordPress-Installationsverzeichnis `/var/www/wordpress/<site directory>`.

Eine ausführliche Beschreibung aller Einstellungen von AWS for WordPress würde den Rahmen dieses Dokuments sprengen. Informationen zu der Konfiguration und den Optionen finden Sie unter [Erste Schritte mit dem AWS for WordPress-Plug-In](#).

Amazon CloudFront und AWS Certificate Manager

So richten Sie CloudFront und AWS Certificate Manager ein:

1. Wählen Sie im Plug-In-Menü CloudFront und geben Sie die folgenden Parameter ein:
 - Domänenname des Ursprungs: DNS-Domäne des HTTP-Ursprungsservers, von dem CloudFront den Inhalt Ihrer Website abrufen (z. B. `example.com`).
 - Alternativer Domänenname (CNAME): Domänenname, den Ihre Besucher für eine schnellere Erfahrung auf der Website verwenden. AWS empfiehlt, „www“ vor der Domäne zu verwenden (z. B. `www.example.com`).
2. Wählen Sie Setup starten, um mit der Konfiguration zu beginnen.

Das Plug-In fordert über ACM automatisch ein SSL-Zertifikat für den CNAME an. Sobald Sie das ACM-Token durch [Aktualisieren der DNS-Einträge](#) mit den CNAME-Einträgen validiert haben, erstellt das Plug-In eine CloudFront-Verteilung, die den in [Anhang A](#) definierten bewährten Methoden entspricht.

Note

Das AWS for WordPress-Plug-In erfordert HTTPS für die Kommunikation zwischen CloudFront und Ihrem benutzerdefinierten Ursprung. Stellen Sie sicher, dass Ihr Ursprung über ein SSL-Zertifikat verfügt, das für den Domännennamen des Ursprungs gültig ist. Ausführlichere Informationen erhalten Sie unter [Erzwingen der Verwendung von HTTPS für die Kommunikation zwischen CloudFront und Ihrem benutzerdefinierten Ursprung](#).

Ihre Inhalte übersetzen und vokalisieren

Mit dem AWS for WordPress-Plug-In können Sie Text automatisch in verschiedene Sprachen übersetzen und den geschriebenen Inhalt in mehrsprachige Audioformate konvertieren. Diese Funktionen werden von Amazon Machine Learning-Services unterstützt.

[Amazon Polly](#) ist ein Service, der Text in natürliche Sprache konvertiert. Amazon Polly beherrscht Dutzende von Stimmen. Für Sprachen stehen Ihnen die idealen Stimmen zur Auswahl, um eine

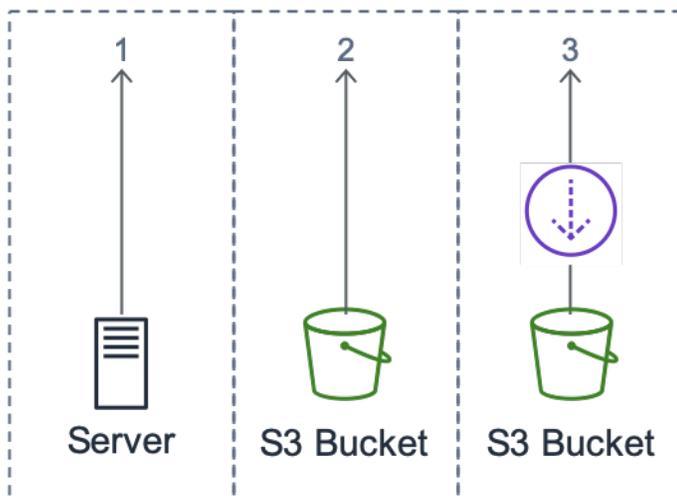
ansprechende sprachgestützte Anwendung für internationale Märkte zu entwickeln. Verwenden Sie das Plug-In zum Erstellen von Audiodateien in einer der von Amazon Polly unterstützten Stimmen und Sprachen. Ihre Besucher können die Audiodaten mithilfe von eingebetteten Audio-Playern und mobilen Anwendungen nach Belieben streamen.

Standardmäßig speichert das Plug-In neue Audiodateien auf Ihrem Webserver. Sie können die Dateien auf Amazon S3 oder Amazon CloudFront speichern. Unabhängig davon, wie Sie Ihre Audiodateien speichern, erhalten Benutzer das gleiche Hörerlebnis. Nur der Übertragungsort ändert sich:

- Für Audiodateien, die auf dem WordPress-Server gespeichert sind, werden die Dateien direkt vom Server aus übertragen.
- Für Dateien, die in einem S3-Bucket gespeichert sind, werden die Dateien vom Bucket aus übertragen.
- Wenn Sie CloudFront verwenden, werden die Dateien auf Amazon S3 gespeichert und mit CloudFront übertragen.



Listening...



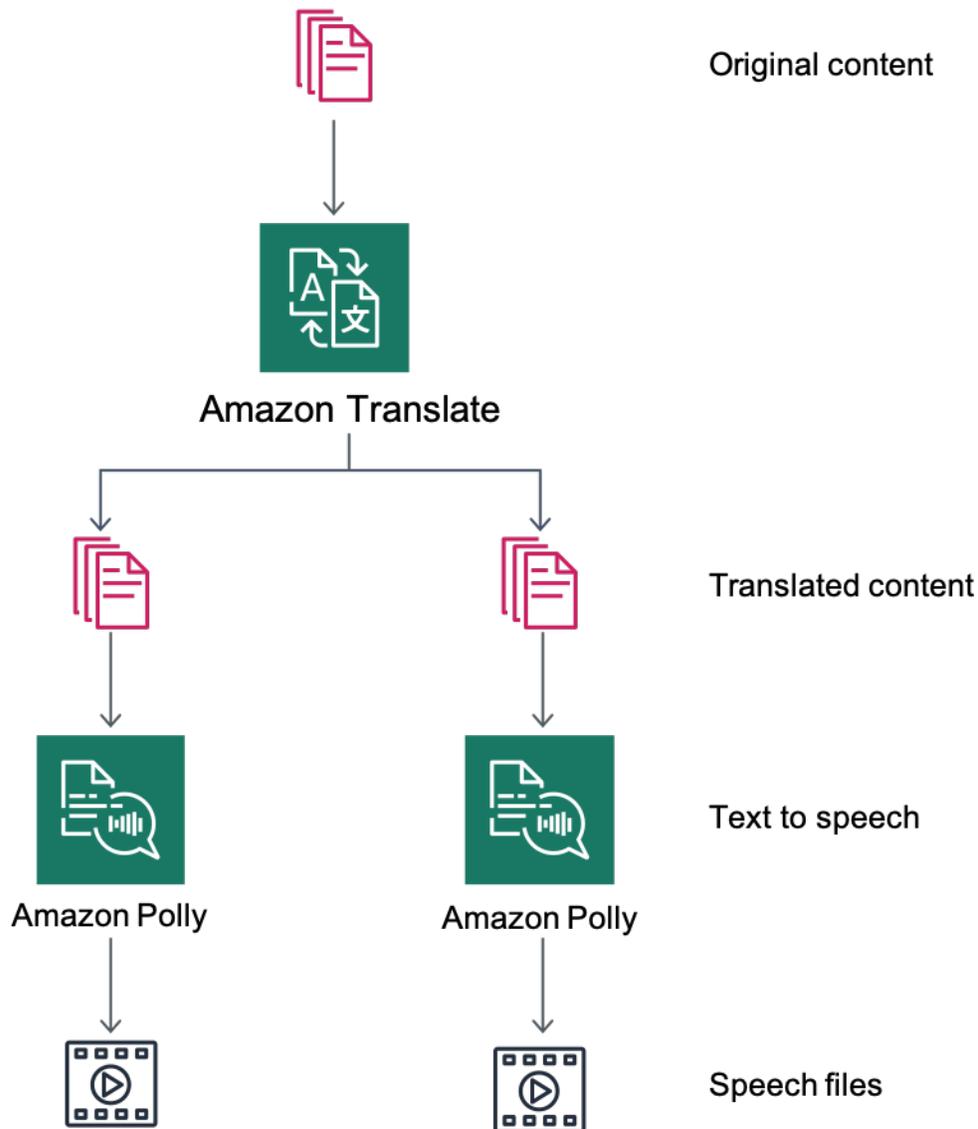
Übertragungsort

[Amazon Translate](#) ist ein Service für maschinelle Übersetzung, der schnell und kostengünstig qualitativ hochwertige Übersetzungen liefert. Die Bereitstellung mehrsprachiger Inhalte bietet eine

großartige Gelegenheit für Website-Besitzer. Obwohl Englisch die dominierende Sprache des Internets ist, machen englische Muttersprachler nur 26 % des gesamten Online-Publikums aus.

Indem Sie Text- und Audioversionen Ihrer WordPress-Inhalte in mehreren Sprachen anbieten, können Sie die Anforderungen eines größeren internationalen Publikums erfüllen. Sie können das Plug-In konfigurieren, um folgende Aktionen durchzuführen:

- Automatische Übersetzung in verschiedene Sprachen und Erstellung von Audioaufnahmen aller Übersetzungen zur Generierung neuer Inhalte bei der Veröffentlichung, oder Übersetzung und Erstellung von Aufzeichnungen für einzelne Beiträge
- Übersetzung in verschiedene Sprachen und Erstellung von Audioaufnahmen aller Übersetzungen Ihres archivierten Inhalts
- Verwendung des Amazon Pollycast RSS-Feeds, um Audio-Inhalt als Podcast bereitzustellen.



Übersicht über die Übersetzung von Inhalten und Text-zu-Sprache

Podcasting mit Amazon Pollycast

Mit den Amazon Pollycast-Feeds können Ihre Besucher Ihre Audioinhalte mithilfe von standardmäßigen Podcast-Anwendungen anhören. RSS 2.0-konforme Pollycast-Feeds bieten die XML-Daten, die erforderlich sind, um Podcasts von beliebten mobilen Podcast-Anwendungen, wie iTunes und Podcast-Verzeichnissen, zu aggregieren.

Wenn Sie das AWS for WordPress-Plug-In installieren, haben Sie auf der Registerkarte Podcast-Konfiguration die Möglichkeit, die Generierung von XML-Feeds zu aktivieren. Dort können Sie auch mehrere optionale Eigenschaften konfigurieren. Nachdem Sie die Funktion aktiviert haben, erhalten Sie einen Link zum Feed.

Ihre Inhalte auf Amazon Alexa-Geräten lesen

Sie können WordPress-Websites und -Blogs über Alexa-Geräte erweitern. Dadurch können Designer und Autoren von Websites ein noch breiteres Publikum erreichen. Außerdem können die Benutzer leichter ihre Lieblingsblogs anhören, indem sie Alexa einfach bitten, sie vorzulesen.

Aktivieren Sie folgende Optionen, damit Alexa auf die WordPress-Website zugreifen kann:

- AWS for WordPress-Plug-In
- Die Text-zu-Sprache- und Amazon Pollycast-Funktionen. Diese Funktionen generieren einen RSS-Feed auf Ihrer WordPress-Site, der von Amazon Alexa verwendet wird.
- Amazon S3 als Standardspeicher für Ihre Dateien in Text-zu-Sprache. Ihre Website muss eine sichere HTTPS-Verbindung verwenden, damit Alexa auf den Feed zugreifen kann.

Das folgende Diagramm zeigt den Interaktionsfluss und die Komponenten an, die erforderlich sind, um Ihre Website über Alexa verfügbar zu machen.



Erforderlicher Interaktionsfluss, um WordPress-Websites über Alexa verfügbar zu machen

1. Der Benutzer ruft einen neuen Alexa-Skill auf, indem er beispielsweise sagt: Alexa, rufe auf dem Demoblog das neueste Update ab. Der Skill wird mit einer der Alexa Skill-Vorlagen erstellt. Auf diese Weise können Sie Ihre Skills auch dann über Alexa-Geräte zur Verfügung stellen, wenn Sie über keine umfassenden technischen Kenntnisse verfügen.
2. Der Alexa-Skill analysiert den Aufruf und den RSS-Feed, der vom AWS for WordPress-Plug-In generiert wurde, und gibt dann den Link zur Audioversion des aktuellen Artikels zurück.
3. Über den vom Feed bereitgestellten Link liest Alexa den Artikel, indem sie die auf Amazon S3 gespeicherte Audiodatei wiedergibt.

Auf der [Plug-In-Seite](#) im WordPress-Marketplace finden Sie eine ausführliche schrittweise Anleitung zum Installieren und Konfigurieren des Plug-Ins und seiner Funktionen.

Konfiguration statischer Inhalte

WordPress speichert standardmäßig alles lokal auf dem Webserver, der als Blockspeicher (Amazon EBS) für die [Bereitstellung auf einem Server](#) und als Dateispeicher (Amazon EFS) für die [elastische Bereitstellung](#) fungiert. Neben der Reduzierung der Speicher- und Datenübertragungskosten bietet die Verlagerung statischer Komponenten auf Amazon S3 eine erhöhte Skalierbarkeit, Datenverfügbarkeit, Sicherheit und Leistung.

In diesem Beispiel wird das W3TC-Plug-In (W3 Total Cache) verwendet, um statische Komponenten auf Amazon S3 zu speichern. Es gibt jedoch auch andere Plug-Ins mit ähnlichen Funktionen. Wenn Sie eine Alternative verwenden möchten, können Sie die folgenden Schritte entsprechend anpassen. Die Schritte beziehen sich nur auf Funktionen oder Einstellungen, die für dieses Beispiel von Belang sind. Eine ausführliche Beschreibung aller Einstellungen würde den Rahmen dieses Dokuments sprengen. Weitere Informationen finden Sie auf der [W3 Total Cache-Plug-In-Seite](#) unter wordpress.org.

Erstellung von IAM-Benutzern

Sie müssen einen IAM-Benutzer (AWS Identity and Access Management) für das WordPress-Plug-In erstellen, um statische Komponenten in Amazon S3 zu speichern. Informationen zu einzelnen Schritten finden Sie unter [Erstellen eines IAM-Benutzers in Ihrem AWS-Konto](#).

Hinweis: Mit IAM-Rollen lässt sich der Zugriff auf AWS-Ressourcen leichter verwalten, aber beim Schreiben unterstützt das W3 Total Cache-Plug-In keine [IAM-Rollen](#).

Notieren Sie sich die Sicherheitsanmeldeinformationen der Benutzer und speichern Sie sie an einem sicheren Ort – Sie werden diese Anmeldeinformationen später brauchen.

Erstellung des Amazon S3-Buckets

1. Erstellen Sie zunächst einen Amazon S3-Bucket in der AWS-Region Ihrer Wahl. Informationen zu einzelnen Schritten finden Sie unter [Erstellen eines Buckets](#). Aktivieren Sie statisches Website-Hosting für den Bucket, indem Sie das [Tutorial: Konfigurieren einer statischen Website auf Amazon S3](#) befolgen.
2. Erstellen Sie eine IAM-Richtlinie, um dem zuvor erstellten IAM-Benutzer Zugriff auf den angegebenen S3-Bucket zu gewähren, und fügen Sie dem IAM-Benutzer die Richtlinie bei.

Informationen zu den einzelnen Schritten zum Erstellen der folgenden Richtlinie finden Sie unter [Verwalten von IAM-Richtlinien](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1389783689000",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::wp-demo",
        "arn:aws:s3:::wp-demo/*"
      ]
    }
  ]
}
```

3. Installieren und aktivieren Sie das W3TC-Plug-In über den WordPress-Adminbereich.
4. Navigieren Sie in der Konfiguration des Plug-Ins zum Abschnitt „Allgemeine Einstellungen“ und stellen Sie sicher, dass sowohl der Cache des Browsers als auch das CDN aktiviert sind.
5. Wählen Sie in der Dropdown-Liste in der CDN-Konfiguration Ursprungs-Push: Amazon CloudFront (diese Option hat Amazon S3 als Ursprung).
6. Navigieren Sie in der Konfiguration des Plug-Ins zum Bereich „Browser-Cache“ und aktivieren Sie die Header Ablaufdatum, Cache-Kontrolle und Entitäts-Tag (ETag).
7. Aktivieren Sie außerdem die Option Caching von Objekten nach Änderung der Einstellungen verhindern, sodass immer, wenn Einstellungen geändert werden, eine neue Abfragezeichenfolge generiert und an Objekte angehängt wird.
8. Navigieren Sie zum CDN-Abschnitt der Konfiguration des Plug-Ins und geben Sie die Sicherheitsanmeldeinformationen des zuvor erstellten IAM-Benutzers sowie den Namen des S3-Buckets ein.

9. Wenn Sie Ihre Website über die CloudFront-URL bereitstellen, geben Sie den Namen der Verteilungsdomäne in das entsprechende Feld ein. Andernfalls geben Sie einen oder mehrere CNAMEs für Ihren oder Ihre benutzerdefinierten Domänennamen ein.
10. Exportieren Sie abschließend die Medienbibliothek und laden Sie die WP-Includes, Theme-Dateien und benutzerdefinierten Dateien mithilfe des W3TC-Plug-Ins auf Amazon S3 hoch. Diese Upload-Funktionen sind im Abschnitt Allgemeines der CDN-Konfigurationsseite verfügbar.

Erstellung eines statischen Ursprungs

Nachdem die statischen Dateien nun auf Amazon S3 gespeichert sind, kehren Sie in der CloudFront-Konsole zur CloudFront-Konfiguration zurück und konfigurieren Sie Amazon S3 als Ursprung für statischen Inhalt. Fügen Sie dazu einen zweiten Ursprung hinzu, der auf den S3-Bucket verweist, den Sie zu diesem Zweck erstellt haben. Erstellen Sie dann zwei weitere Cache-Verhaltensweisen, jeweils eine für beide Ordner (`wp-content` und `wp-includes`), die den S3-Ursprung anstelle des Standardursprungs für dynamischen Inhalt verwenden sollen. Konfigurieren Sie beide auf dieselbe Weise:

- Bearbeitet nur HTTP-GET-Anfragen.
- Die Ausgabe von Amazon S3 wird durch Cookies oder HTTP-Header nicht beeinflusst. Das bedeutet, dass Sie die Caching-Effizienz verbessern können, indem Sie sie nicht über CloudFront an den Ursprung weiterleiten.
- Leiten Sie Abfragezeichenfolgen an den Ursprung weiter, obwohl diese Verhaltensweisen nur statischen Inhalt (der keine Parameter akzeptiert) bereitstellen. Auf diese Weise können Sie Abfragezeichenfolgen als Versionskennungen verwenden, um beispielsweise ältere CSS-Dateien bei der Bereitstellung neuer Versionen sofort ungültig zu machen. Weitere Informationen finden Sie im [Amazon CloudFront-Entwicklerhandbuch](#).

Note

Nachdem Sie Ihrer CloudFront-Verteilung die Verhaltensweisen des statischen Ursprungs hinzugefügt haben, überprüfen Sie die Reihenfolge, um sicherzustellen, dass die Verhaltensweisen für `wp-admin/*` und `wp-login.php` eine höhere Priorität haben als die Verhaltensweisen für statischen Inhalt. Andernfalls können beim Zugriff auf Ihren Adminbereich seltsame Verhaltensweisen auftreten.

Anhang C: Backup und Wiederherstellung

Die Durchführung einer Wiederherstellung nach einem Ausfall in AWS ist im Vergleich zu herkömmlichen Hosting-Umgebungen wesentlich schneller und einfacher. Sie können beispielsweise bei einem Hardwarefehler innerhalb von Minuten eine Ersatz-Instance starten, oder Sie können in vielen unserer verwalteten Services ein automatisiertes Failover durchführen, damit ein Neustart aufgrund von Routinewartungen keine negativen Auswirkungen hat.

Sie müssen jedoch weiterhin sicherstellen, dass Sie die richtigen Daten sichern, um sie erfolgreich wiederherzustellen. Um eine WordPress-Website wieder verfügbar zu machen, müssen Sie in der Lage sein, folgende Komponenten wiederherzustellen:

- Installation und Konfiguration des Betriebssystems (OS) und der Services (Apache, MySQL usw.)
- WordPress-Anwendungscode und Konfiguration
- WordPress-Themes und -Plug-Ins
- Uploads (z. B. Mediendateien für Beiträge)
- Datenbankinhalt (Beiträge, Kommentare usw.)

AWS bietet eine Vielzahl von Methoden zum Sichern und Wiederherstellen Ihrer Webanwendungsdaten und -komponenten.

In diesem Whitepaper wurde zuvor die Verwendung von Lightsail-Snapshots zum Schutz aller im lokalen Speicher der Instance gespeicherten Daten erörtert. Wenn Ihre WordPress-Website nur von der Lightsail-Instance ausgeführt wird, sollten regelmäßige Lightsail-Snapshots ausreichen, damit Sie Ihre WordPress-Website vollständig wiederherstellen können. Trotzdem verlieren Sie alle Änderungen, die seit dem letzten Snapshot auf Ihrer Website vorgenommen wurden, wenn Sie eine Wiederherstellung aus einem Snapshot durchführen.

In einer Multi-Server-Bereitstellung müssen Sie alle zuvor besprochenen Komponenten mithilfe verschiedener Mechanismen sichern. Jede Komponente kann andere Anforderungen bezüglich der Sicherungshäufigkeit haben. Beispielsweise ändert sich die Installation und Konfiguration des Betriebssystems und von WordPress viel seltener als benutzergenerierter Inhalt, wodurch dieser weniger oft gesichert werden muss, ohne dass bei einer Wiederherstellung Daten verloren gehen.

Um die Installation und Konfiguration des Betriebssystems und der Services sowie die Anwendungscode und Konfiguration von WordPress zu sichern, können Sie ein AMI einer

ordnungsgemäß konfigurierten EC2-Instance erstellen. Mit AMIs werden zwei Zwecke verfolgt: zur Sicherung des Instance-Status und als Vorlage beim Starten neuer Instances.

Sie müssen AMIs und Aurora-Backups verwenden, um den Anwendungscode und die Konfiguration von WordPress zu sichern.

Um die auf Ihrer Website installierten WordPress-Themes und -Plug-Ins zu sichern, erstellen Sie ein Backup des Amazon S3-Buckets oder des Amazon EFS-Dateisystems, in dem sie gespeichert sind.

- Für Themes und Plug-Ins, die in einem S3-Bucket gespeichert sind, können Sie die [regionsübergreifende Replikation](#) aktivieren, damit alle in Ihren primären Bucket hochgeladenen Objekte automatisch in Ihren Backup-Bucket in einer anderen AWS-Region repliziert werden. Für die regionsübergreifende Replikation muss das [Versioning](#) sowohl für Ihren Quell- als auch für den Ziel-Bucket aktiviert sein. Dadurch erhalten Sie eine zusätzliche Schutzebene und können eine frühere Version eines beliebigen Objekts in Ihrem Bucket wiederherstellen.
- Für in einem EFS-Dateisystem gespeicherte Themes und Plug-Ins können Sie eine AWS Data Pipeline erstellen, um Daten aus Ihrem EFS-Produktionsdateisystem in ein anderes EFS-Dateisystem zu kopieren, wie auf der Dokumentationsseite [Sichern Ihrer Amazon EFS-Dateisysteme](#) beschrieben. Sie können ein EFS-Dateisystem auch mit einer beliebigen Backup-Anwendung sichern, mit der Sie bereits vertraut sind.
- Um Benutzer-Uploads zu sichern, müssen Sie die zuvor beschriebenen Schritte zum Sichern der WordPress-Themes und -Plug-Ins ausführen.
- Verwenden Sie die Option [Aurora-Backup](#), um ein Backup der Datenbankinhalte zu erstellen. Aurora sichert Ihr Cluster-Volume automatisch für die eingestellte Dauer Aufbewahrungszeitraum für Backups und stellt es, wenn erforderlich, wieder her. Aurora-Backups werden fortlaufend und inkrementell erstellt, damit Sie schnell zu jedem Zeitpunkt innerhalb des Aufbewahrungszeitraums für Backups eine Wiederherstellung durchführen können. Es gibt keine Auswirkungen auf die Leistungsfähigkeit oder Unterbrechung von Datenbankservices, während Backup-Daten geschrieben werden. Sie können den Aufbewahrungszeitraum für Backups auf 1 bis 35 Tage festlegen. Sie können auch [manuelle Datenbank-Snapshots](#) erstellen, die so lange erhalten bleiben, bis Sie sie löschen. Manuelle Datenbank-Snapshots sind für langfristige Backups und zur Archivierung nützlich.

Anhang D: Bereitstellen neuer Plug-Ins und Themes

Einige Websites bleiben statisch. In den meisten Fällen fügen Sie regelmäßig öffentlich verfügbare WordPress-Themes und -Plug-Ins hinzu oder aktualisieren auf eine neuere WordPress-Version. Andere Male entwickeln Sie Ihre eigenen benutzerdefinierten Themes und Plug-Ins aber auch von Grund auf neu.

Bei jeder strukturellen Änderung an Ihrer WordPress-Installation besteht ein gewisses Risiko für unvorhergesehene Probleme. Erstellen Sie zumindest ein Backup Ihres Anwendungscodes, Ihrer Konfiguration und Ihrer Datenbank, bevor Sie wichtige Änderungen vornehmen (z. B. die Installation eines neuen Plug-Ins). Testen Sie diese Änderungen für Websites mit geschäftlichem oder anderem Wert zuerst in einer separaten Staging-Umgebung. Mit AWS lässt sich die Konfiguration Ihrer Produktionsumgebung leicht replizieren und der gesamte Bereitstellungsprozess auf sichere Weise ausführen. Wenn Sie mit Ihren Tests fertig sind, können Sie einfach Ihre Testumgebung beenden und die Bezahlung der Ressourcen einstellen. Später werden in diesem Whitepaper einige WordPress-spezifische Überlegungen erörtert.

Einige Plug-Ins schreiben Konfigurationsinformationen in die `wp_options`Datenbanktabelle (oder implementieren Änderungen des Datenbankschemas) und andere erstellen Konfigurationsdateien im WordPress-Installationsverzeichnis. Da wir die Datenbank und den Speicher auf gemeinsam genutzte Plattformen verschoben haben, sind diese Änderungen sofort für alle Ihre laufenden Instances verfügbar, ohne dass Sie weitere Aktionen ausführen müssen.

Bei der Bereitstellung neuer Themes in WordPress ist möglicherweise etwas mehr Aufwand erforderlich. Wenn Sie Amazon EFS nur zum Speichern all Ihrer WordPress-Installationsdateien verwenden, sind neue Themes sofort für alle laufenden Instances verfügbar. Wenn Sie jedoch statischen Inhalt nach Amazon S3 auslagern, müssen Sie eine Kopie im richtigen Bucket speichern. Mit Plug-Ins wie W3 Total Cache können Sie diese Aufgabe manuell einleiten. Alternativ können Sie diesen Schritt auch im Rahmen eines Build-Prozesses automatisieren.

Da Theme-Komponenten in CloudFront und im Browser zwischengespeichert werden können, müssen Sie bei der Bereitstellung von Änderungen ältere Versionen ungültig machen. Integrieren Sie dazu am besten eine Art Versionskennung in Ihr Objekt. Diese Kennung kann eine Abfragezeichenfolge mit einem Datums- und Zeitstempel oder einer zufälligen Zeichenfolge sein. Wenn Sie das W3 Total Cache-Plug-In verwenden, können Sie eine Medienabfragezeichenfolge aktualisieren, die den URLs von Mediendateien angehängt wird.

Hinweise

Kunden sind eigenverantwortlich für die unabhängige Bewertung der Informationen in diesem Dokument zuständig. Dieses Dokument: (a) dient rein zu Informationszwecken, (b) spiegelt die aktuellen Produktangebote und Verfahren von AWS wider, die sich ohne vorherige Mitteilung ändern können, und (c) impliziert keinerlei Verpflichtungen oder Zusicherungen seitens AWS und dessen Tochtergesellschaften, Lieferanten oder Lizenzgebern. AWS-Produkte oder -Services werden im vorliegenden Zustand und ohne ausdrückliche oder stillschweigende Gewährleistungen, Zusicherungen oder Bedingungen bereitgestellt. Die Verantwortung und Haftung von AWS gegenüber seinen Kunden wird durch AWS-Vereinbarungen geregelt. Dieses Dokument ist weder ganz noch teilweise Teil der Vereinbarungen zwischen AWS und seinen Kunden und ändert diese Vereinbarungen auch nicht.

© 2021 Amazon Web Services Inc. bzw. Tochtergesellschaften des Unternehmens. Alle Rechte vorbehalten.