Whitepaper zu AWS

# Hosting von Webanwendungen in der AWS Cloud

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

# Hosting von Webanwendungen in der AWS Cloud: 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
Herkömmliches Webhosting - eine Übersicht ..... 2
Webanwendungshosten in der Cloud mithilfe von AWS ..... 4
Wie AWS allgemeine Hostingprobleme von Webanwendungen lösen kann ..... 4
Eine kostengünstige Alternative zu übergroßen Flotten, die zur Handhabung von Spitzenkapazitäten benötigt werden. ..... 4
Eine skalierbare Lösung zur Handhabung unerwarteter Verkehrsspitzen ..... 5
Als On-Demand-Lösung für Test-, Load-, Beta- und Vorproduktionsumgebungen ..... 5
Eine AWS Cloud-Architektur für Webhosting ..... 6
Schlüsselkomponenten einer AWS Webhosting-Architektur ..... 8
Netzwerkmanagement ..... 8
Bereitstellung von Inhalten ..... 9
Verwaltung des öffentlichen DNS ..... 9
Host-Sicherheit ..... 10
Lastenverteilung über Cluster hinweg ..... 10
Suchen anderer Hosts und Services ..... 11
Caching innerhalb der Webanwendung ..... 11
Datenbankkonfiguration, Sicherung und Ausfallsicherung ..... 11
Speicherung und Sicherung von Daten und Komponenten ..... 14
Automatische Skalierung der Flotte ..... 15
Zusätzliche Sicherheitsfunktionen ..... 16
Failover mit AWS ..... 17
Wichtige Überlegungen bei der Verwendung von AWS für Webhosting ..... 18
Keine physischen Netzwerkgeräte mehr ..... 18
Überall Firewalls ..... 18
Bedenken Sie die Verfügbarkeit mehrerer Rechenzentren ..... 18
Hosts werden als ephemer und dynamisch behandelt ..... 19
Bedenken Sie Container und Serverless ..... 19
Bedenken Sie automatisierte Bereitstellung ..... 19
Fazit und Mitwirkende ..... 21
Fazit ..... 21
Mitwirkende ..... 21
Weitere Informationen ..... 22

Dokumentversionen ..... 23
Hinweise ..... 25

# Hosting von Webanwendungen in der AWS Cloud

Veröffentlichungsdatum: 20. August 2021 (Dokumentversionen)

## Überblick

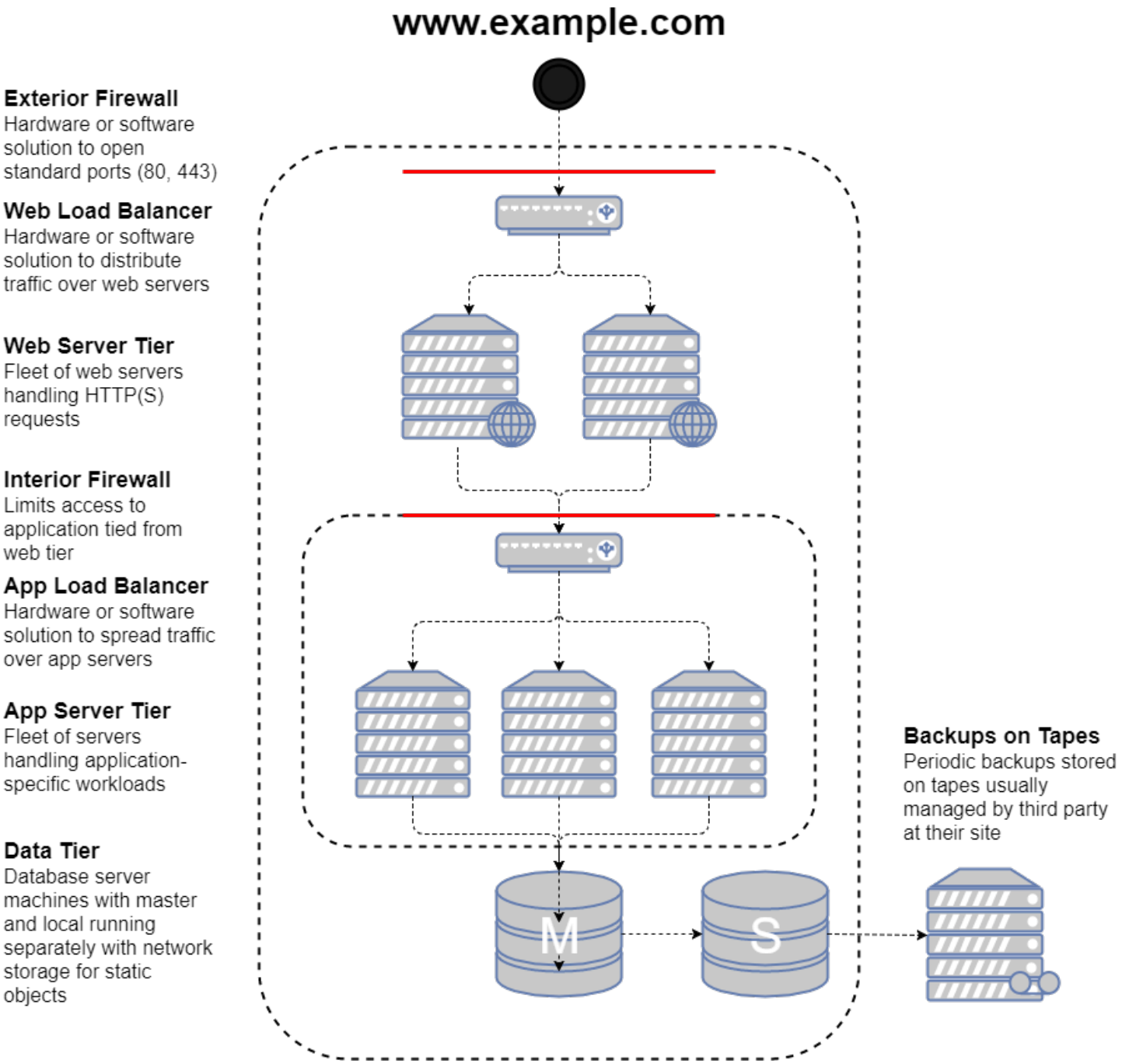
Herkömmliche On-Premises-Webarchitekturen erfordern komplexe Lösungen und eine genaue Prognose der reservierten Kapazität, um die Zuverlässigkeit zu gewährleisten. Dichte Spitzenverkehrszeiten und starke Schwankungen der Verkehrsmuster führen zu niedrigen Auslastungsraten teurer Hardware. Dies führt zu hohen Betriebskosten für die Wartung von inaktiver Hardware und zu einem ineffizienten Einsatz von Kapital für nicht ausreichend genutzte Hardware.

Amazon Web Services (AWS) bietet eine zuverlässige, skalierbare, sichere und leistungsfähige Infrastruktur für die anspruchsvollsten Webanwendungen. Diese Infrastruktur gleicht die IT-Kosten nahezu in Echtzeit mit den Verkehrsmustern der Kunden ab.

Dieses Whitepaper richtet sich an IT-Manager und Systemarchitekten, die verstehen möchten, wie herkömmliche Webarchitekturen in der Cloud ausgeführt werden, um Elastizität, Skalierbarkeit und Zuverlässigkeit zu erreichen.

# Herkömmliches Webhosting - eine Übersicht

Skalierbares Webhosting ist ein bekannter Problembereich. Das folgende Bild zeigt eine herkömmliche Webhosting-Architektur, die ein gängiges dreistufiges Webanwendungsmodell implementiert. In diesem Modell ist die Architektur in Darstellungs-, Anwendungs- und Persistenzebenen unterteilt. Skalierbarkeit wird durch Hinzufügen von Hosts auf diesen Ebenen gewährleistet. Die Architektur verfügt außerdem über integrierte Leistungs-, Failover- und Verfügbarkeitsfunktionen. Die herkömmliche Webhosting-Architektur lässt sich mit nur wenigen Änderungen problemlos in die AWS Cloud portieren.




Eine herkömmliche Webhosting-Architektur

In den folgenden Abschnitten wird erläutert, warum und wie eine solche Architektur in der AWS Cloud bereitgestellt werden sollte und könnte.

# Webanwendungshosten in der Cloud mithilfe von AWS

Die erste Frage, die Sie stellen sollten, betrifft den Wert der Verlagerung einer klassischen Lösung des Webanwendungshostings in die AWS Cloud. Wenn Sie entscheiden, dass die Cloud für Sie geeignet ist, benötigen Sie eine geeignete Architektur. In diesem Abschnitt können Sie eine AWS Cloud-Lösung bewerten. Es vergleicht die Bereitstellung Ihrer Webanwendung in der Cloud mit einer On-Premises-Bereitstellung, stellt eine AWS Cloud-Architektur zum Hosten Ihrer Anwendung dar und erörtert die wichtigsten Komponenten der AWS Cloud Architecture-Lösung.

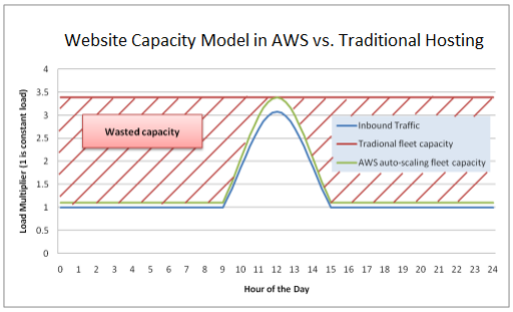
## Wie AWS allgemeine Hostingprobleme von Webanwendungen lösen kann

Wenn Sie für die Ausführung einer Webanwendung verantwortlich sind, könnten Sie auf eine Reihe von Infrastruktur- und Architekturproblemen stoßen, für die AWS einfache, nahtlose und kostengünstige Lösungen bietet. Nachstehend finden Sie einige Vorteile angeführt, die die Nutzung von AWS im Vergleich zu einem herkömmlichen Hostingmodell bietet:

### Eine kostengünstige Alternative zu übergroßen Flotten, die zur Handhabung von Spitzenkapazitäten benötigt werden.

Beim herkömmlichen Hostingmodell müssen Sie Server bereitstellen, um die Spitzenkapazität zu bewältigen. Nicht genutzte Zyklen werden außerhalb der Spitzenzeiten verschwendet. Von AWS gehostete Webanwendungen können die On-Demand-Bereitstellung zusätzlicher Server nutzen, sodass Sie Kapazität und Kosten ständig an die tatsächlichen Verkehrsmuster anpassen können.

Das folgende Diagramm zeigt beispielsweise eine Webanwendung mit einer Spitzenauslastung von 9 bis 15 Uhr und weniger Nutzung für den Rest des Tages. Ein automatischer Skalierungsansatz, der auf tatsächlichen Verkehrstrends basiert und Ressourcen nur bei Bedarf bereitstellt, würde zu weniger verschwendeter Kapazität und zu einer Reduzierung der Kosten um mehr als 50 Prozent führen.




Ein Beispiel verschwendeter Kapazität in einem klassischen Hostingmodell

## Eine skalierbare Lösung zur Handhabung unerwarteter Verkehrsspitzen

Eine sogar noch schwerwiegendere Folge der langsamen Bereitstellung, die mit dem herkömmlichen Hostingmodell verbunden ist, ist die Unfähigkeit zeitgerecht auf unerwartete Verkehrsspitzen zu reagieren. Es gibt eine Reihe von Berichten darüber, dass Webanwendungen aufgrund eines unerwarteten Anstiegs des Datenverkehrs nicht mehr verfügbar sind, nachdem die Website in populären Medien erwähnt wurde. In der AWS Cloud kann dieselbe On-Demand-Funktion, mit der Webanwendungen an reguläre Datenverkehrsspitzen skaliert werden können, auch eine unerwartete Last bewältigen. Neue Hosts können gestartet werden und sind in wenigen Minuten verfügbar. Sie können genauso schnell offline geschaltet werden, wenn sich der Verkehr wieder normalisiert.

## Als On-Demand-Lösung für Test-, Load-, Beta- und Vorproduktionsumgebungen

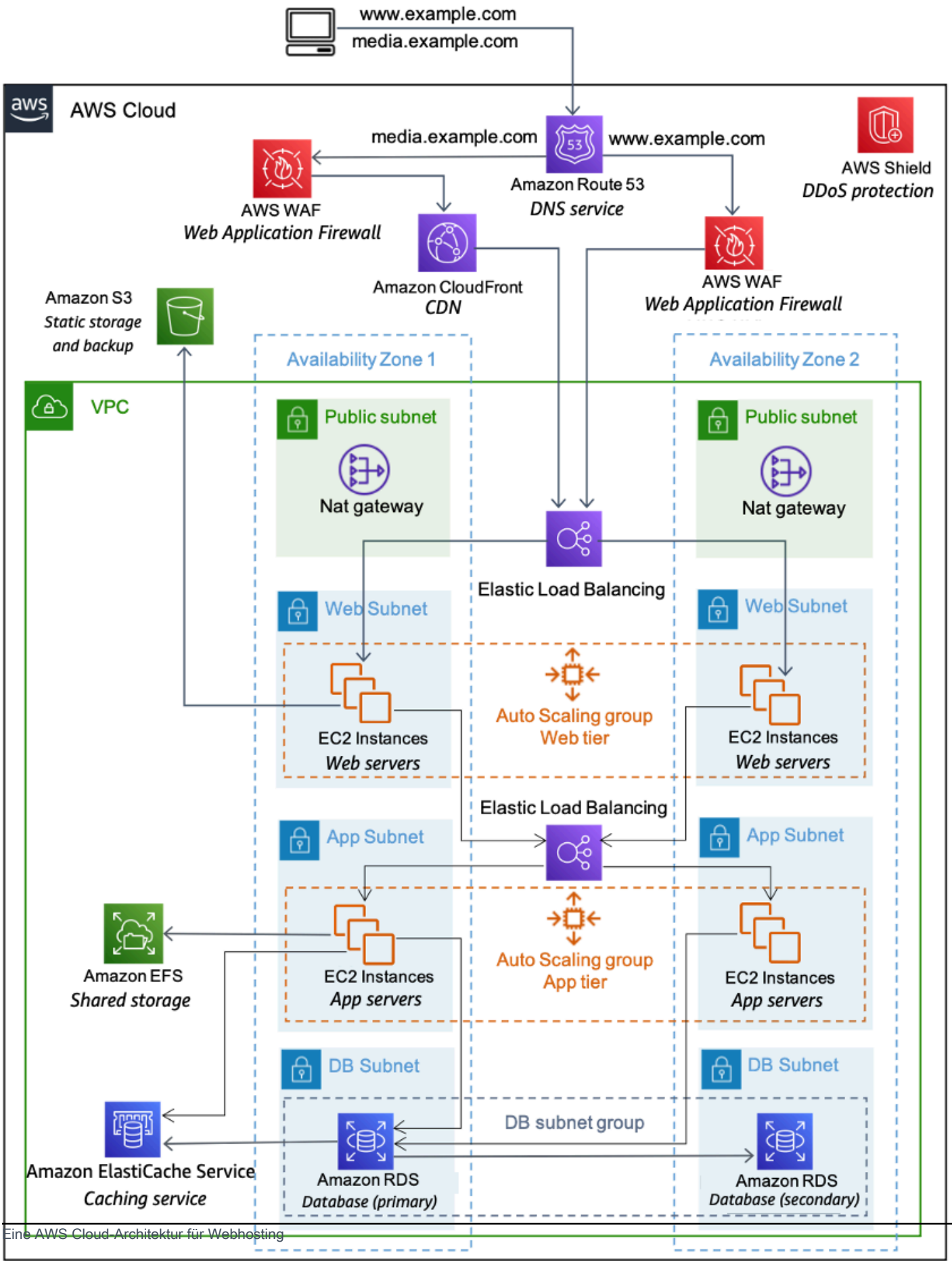
Die Hardwarekosten für den Ausbau einer herkömmlichen Hostingumgebung für die Produktions-Webanwendung enden nicht mit dem Gerätepark für die Produktion. Oft müssen Sie Vorproduktions-, Beta- und Testflotten erstellen, um die Qualität der Webanwendung in jeder Phase des Entwicklungslebenszyklus sicherzustellen. Während Sie verschiedene Optimierungen vornehmen können, um die höchstmögliche Nutzung dieser Testhardware zu gewährleisten, werden diese parallelen Flotten nicht immer optimal genutzt, und eine Menge teurer Hardware bleibt für längere Zeit ungenutzt.

In der AWS Cloud können Sie Testflotten nach Bedarf bereitstellen. Dadurch entfällt nicht nur die Notwendigkeit, Ressourcen Tage oder Monate vor der tatsächlichen Nutzung vorab bereitzustellen, sondern gibt Ihnen auch die Flexibilität, die Infrastrukturkomponenten abzubauen, wenn Sie sie nicht benötigen. Darüber hinaus können Sie während des Lasttests den Benutzerverkehr in der AWS Cloud simulieren. Sie können diese parallelen Flotten auch als Staging-Umgebung für eine

neue Produktionsversion verwenden. Dies ermöglicht eine schnelle Umstellung von der aktuellen Produktion auf eine neue Anwendungsversion mit geringen oder keinen Serviceausfällen.

# Eine AWS Cloud-Architektur für Webhosting

Die nachstehende Abbildung zeigt diese klassische Webanwendungs-Architektur und wie sie die AWS Cloud Computing-Infrastruktur nutzen kann.


www.example.com
media.example.com
AWS Cloud
media.example.com
www.example.com
AWS Shield
DDoS protection
AWS WAF
Web Application Firewall
Amazon Route 53
DNS service
Amazon CloudFront
CDN
AWS WAF
Web Application Firewall
Amazon S3
Static storage and backup
Availability Zone 1
Availability Zone 2
VPC
Public subnet
Nat gateway
Public subnet
Nat gateway
Elastic Load Balancing
Web Subnet
Web Subnet
EC2 Instances
Web servers
Auto Scaling group
Web tier
EC2 Instances
Web servers
Elastic Load Balancing
App Subnet
App Subnet
Amazon EFS
Shared storage
EC2 Instances
App servers
Auto Scaling group
App tier
EC2 Instances
App servers
DB Subnet
DB Subnet
DB subnet group
Amazon ElastiCache Service
Caching service
Amazon RDS
Database (primary)
Amazon RDS
Database (secondary)

Beispiel einer Webhosting-Architektur auf AWS

1. DNS-Dienste mit Amazon Route 53 — Bietet DNS-Dienste zur Vereinfachung der Domainverwaltung.
2. Edge-Caching mit Amazon CloudFront — Edge speichert Inhalte mit hohem Volumen im Cache, um die Latenz für Kunden zu verringern.
3. Edge-Sicherheit für Amazon CloudFront mitAWS WAF — Filtert bösartigen Datenverkehr, einschließlich Cross-Site-Scripting (XSS) und SQL-Injection über kundenspezifische Regeln.
4. Load Balancing mit Elastic Load Balancing (ELB) — Ermöglicht die Lastenverteilung auf mehrere Availability Zones und AWS Auto-Scaling-Gruppen für Redundanz und Entkopplung von Services.
5. DDoS-Schutz mitAWS Shield — Schützt Ihre Infrastruktur automatisch vor den häufigsten DDoS-Angriffen auf Netzwerk- und Transportebene.
6. Firewalls mit Sicherheitsgruppen — Verschiebt die Sicherheit auf die Instance, um eine statusbehaftete Firewall auf Host-Ebene für Web- und Anwendungsserver bereitzustellen.
7. Caching mit Amazon ElastiCache — Bietet Caching-Dienste mit Redis oder Memcached, um Last aus der App und Datenbank zu entfernen und die Latenz bei häufigen Anfragen zu verringern.
8. Verwaltete Datenbank mit Amazon Relational Database Service (Amazon RDS) — Erstellt eine hochverfügbare Multi-AZ-Datenbankarchitektur mit sechs möglichen DB-Engines.
9. Statischer Speicher und Backups mit Amazon Simple Storage Service (Amazon S3) — Ermöglicht einfachen HTTP-basierten Objektspeicher für Backups und statische Assets wie Bilder und Videos.

# Schlüsselkomponenten einer AWS Webhosting-Architektur

In den folgenden Abschnitten werden einige der wichtigsten Komponenten einer in der AWS Cloud bereitgestellten Webhosting-Architektur beschrieben und erläutert, wie sie sich von einer herkömmlichen Webhosting-Architektur unterscheiden.

## Netzwerkmanagement

In der AWS Cloud ermöglicht die Möglichkeit, Ihr Netzwerk von dem anderer Kunden zu segmentieren, eine sicherere und skalierbarere Architektur. Während Sicherheitsgruppen Sicherheit auf Host-Ebene bieten (siehe Abschnitt Host Security), ermöglicht Amazon Virtual Private Cloud (Amazon VPC) Ihnen, Ressourcen in einem logisch isolierten und virtuellen Netzwerk zu starten, das Sie definieren.

Amazon VPC ist ein Service, mit dem Sie die volle Kontrolle über die Details Ihrer Netzwerkeinrichtung in AWS haben. Beispiele für diese Steuerung sind das Erstellen öffentlich zugänglicher Subnetze für Webserver und private Subnetze ohne Internetzugang für Ihre Datenbanken. Darüber hinaus ermöglicht Amazon VPC Ihnen, Hybridarchitekturen mithilfe von virtuellen privaten Hardware-Netzwerken (VPNs) zu erstellen und die AWS Cloud als Erweiterung Ihres eigenen Rechenzentrums zu verwenden.

Amazon VPC umfasst zusätzlich zur herkömmlichen IPv4-Unterstützung für Ihr Netzwerk auch IPv6-Unterstützung.

## Bereitstellung von Inhalten

Wenn Ihr Web-Datenverkehr geo-dispersiert ist, ist es nicht immer machbar und schon gar nicht kosteneffektiv, Ihre gesamte Infrastruktur über den gesamten Globus zu replizieren. Ein Content Delivery Network (CDN) bietet Ihnen die Möglichkeit, sein globales Netzwerk von Edge-Standorten zu nutzen, um Ihren Kunden eine zwischengespeicherte Kopie von Webinhalten wie Videos, Webseiten, Bildern usw. bereitzustellen. Um die Antwortzeit zu verkürzen, nutzt das CDN den dem Kunden oder dem Ort der ursprünglichen Anfrage nächstgelegenen Edge-Standort. Der Durchsatz wird drastisch erhöht, da die Web-Assets aus dem Cache geliefert werden. Für dynamische Daten können viele CDNs so konfiguriert werden, dass sie die Daten von den Ursprungsservern abrufen.

Sie können CloudFront zur Bereitstellung Ihrer Website, einschließlich dynamischer, statischer und Streaming-Inhalte, mithilfe eines globalen Netzwerks von Edge-Standorten verwenden. CloudFront leitet Ihre Inhalte automatisch an den nächsten Edge-Standort weiter, sodass die Bereitstellung der Inhalte mit der bestmöglichen Leistung erfolgt. CloudFront ist für die Arbeit mit anderen AWS-Services wie Amazon S3 und Amazon Elastic Compute Cloud (Amazon EC2) optimiert. CloudFront funktioniert außerdem reibungslos mit allen nicht zu AWS gehörenden Ursprungsservern, auf denen die maßgeblichen Originalversionen Ihrer Dateien gespeichert sind.

Wie bei anderen AWS-Services gibt es keine vertraglichen oder sonstigen monatlichen Verpflichtungen zur Nutzung von Amazon CloudFront – Sie zahlen nur für die Inhalte, die Sie auch tatsächlich bereitstellen.

Darüber hinaus sollten alle vorhandenen Lösungen für Edge-Caching in Ihrer Webanwendungsinfrastruktur in der AWS Cloud gut funktionieren.

## Verwaltung des öffentlichen DNS

Das Verschieben einer Webanwendung in die AWS Cloud erfordert einige Änderungen des Domain Name System (DNS). Um Ihnen bei der Verwaltung des DNS-Routings zu helfen, bietet AWS

Amazon Route 53, einen hochverfügbaren und skalierbaren Cloud-DNS-Webservice. Entwicklern und Unternehmen bietet Route 53 eine äußerst zuverlässige und kostengünstige Möglichkeit, Endbenutzer zu Internetanwendungen zu routen. Dazu werden Namen wie „www.beispiel.de“ in numerische IP-Adressen wie 192.0.2.1 übersetzt, die Computer untereinander für den Verbindungsaufbau verwenden. Route 53 ist uneingeschränkt mit IPv6 kompatibel.

## Host-Sicherheit

Zusätzlich zur Filterung des eingehenden Netzwerkverkehrs am Edge empfiehlt AWS auch, dass Webanwendungen die Filterung des Netzwerkverkehrs auf Host-Ebene anwenden. Amazon EC2 bietet eine Funktion mit dem Namen Sicherheitsgruppen. Eine Sicherheitsgruppe entspricht einer Firewall für eingehende Netzwerke, für die Sie die Protokolle, Ports und Quell-IP-Bereiche angeben können, die Ihre EC2-Instances erreichen dürfen.

Sie können jeder EC2-Instance eine oder mehrere Sicherheitsgruppen zuweisen. Jede Sicherheitsgruppe ermöglicht entsprechenden Datenverkehr zu jeder Instanz. Sicherheitsgruppen können so konfiguriert werden, dass nur bestimmte Subnetze, IP-Adressen und Ressourcen Zugriff auf eine EC2-Instance haben. Alternativ können sie auf andere Sicherheitsgruppen verweisen, um den Zugriff auf EC2-Instances zu beschränken, die sich in bestimmten Gruppen befinden.

In der AWS-Webhosting-Architektur in Abbildung 3 erlaubt die Sicherheitsgruppe für den Webservercluster möglicherweise nur den Zugriff von Web-Layer-Lastenverteilung und nur über TCP an den Ports 80 und 443 (HTTP und HTTPS). Die Sicherheitsgruppe des Anwendungsservers hingegen erlaubt möglicherweise nur den Zugriff von Lastenverteilung auf Anwendungsebene. Bei diesem Modell müssten Ihre Support-Techniker auch auf die EC2-Instances zugreifen, was mit AWS Systems Manager Session Manager erreicht werden kann. Eine eingehendere Diskussion zur Sicherheit finden Sie in AWS Cloud Security, das Sicherheitsbulletins, Zertifizierungsinformationen und Sicherheits-Whitepaper enthält, in denen die Sicherheitsfunktionen von AWS erläutert werden.

## Lastenverteilung über Cluster hinweg

Hardware-Lastenverteilungen sind gebräuchliche Netzwerkgeräte, die in herkömmlichen Webanwendungs-Architekturen zum Einsatz kommen. AWS stellt diese Funktion über den Elastic Load Balancing (ELB)-Service bereit. ELB verteilt den eingehenden Anwendungsverkehr automatisch auf mehrere Ziele, wie Amazon EC2 Instances, Container, IP-Adressen und AWS Lambda-Funktionen. Es kann die variable Last Ihres Anwendungsdatenverkehrs in eine einzelne Availability Zone oder in mehrere Availability Zones leiten. Elastic Load Balancing bietet vier Arten von Lastenverteilung, die alle die hohe Verfügbarkeit, die automatische Skalierung und die robuste Sicherheit bieten, die notwendig sind, um Ihre Anwendungen fehlertolerant zu machen.

## Suchen anderer Hosts und Services

In der traditionellen Webhosting-Architektur haben die meisten Ihrer Hosts statische IP-Adressen. In der AWS Cloud haben die meisten Ihrer Hosts dynamische IP-Adressen. Obwohl jede EC2-Instance sowohl öffentliche als auch private DNS-Einträge haben kann und über das Internet adressierbar ist, werden die DNS-Einträge und die IP-Adressen dynamisch zugewiesen, wenn Sie die Instance starten. Sie können nicht manuell zugewiesen werden. Statische IP-Adressen (elastische IP-Adressen in der AWS-Terminologie) können laufenden Instances zugewiesen werden, nachdem sie gestartet wurden. Sie sollten Elastic IP-Adressen für Instanzen und Services verwendet werden, die konsistente Endpunkte erfordern, z.B. primäre Datenbanken, zentrale Dateiserver und EC2-gehostete Lastenverteilungen.

## Caching innerhalb der Webanwendung

In-Memory-Anwendungscaches können die Belastung von Diensten reduzieren und die Leistung und Skalierbarkeit auf der Datenbankebene verbessern, indem häufig verwendete Informationen zwischengespeichert werden. Amazon ElastiCache ist ein Web-Service, mit dem ein In-Memory-Cache auf einfache Weise in der Cloud bereitgestellt, betrieben und skaliert werden kann. Sie können den von Ihnen erstellten In-Memory-Cache so konfigurieren, dass er automatisch mit Last skaliert und ausgefallene Knoten automatisch ersetzt. ElastiCache ist protokollkonform mit Memcached und Redis, was die Migration von Ihrer aktuellen lokalen Lösung vereinfacht.

## Datenbankkonfiguration, Sicherung und Ausfallsicherung

Viele Webanwendungen enthalten irgendeine Form von Persistenz, normalerweise in Form einer relationalen oder nicht relationalen Datenbank. AWS bietet sowohl relationale als auch nicht relationale Datenbankservices. Alternativ können Sie Ihre eigene Datenbanksoftware auf einer EC2-Instance bereitstellen. In der folgenden Tabelle sind diese Optionen zusammengefasst, die in diesem Abschnitt näher erläutert werden.

Tabelle 1 — Relationale und nicht relationale Datenbanklösungen

| | Lösungen für relationale Datenbanken | NoSQL-Lösungen |
|---|---|---|
| Verwalteter Datenbankdienst | Amazon RDS for MySQL, Oracle, SQL Server, MariaDB, PostgreSQL, Amazon Aurora | Amazon DynamoDB, Amazon Keyspaces, Amazon Neptune, |

| | Lösungen für relationale Datenbanken | NoSQL-Lösungen |
|---|---|---|
| | | Amazon QLDB, Amazon Timestream |
| Selbstverwaltet | Hosten eines relationalen Datenbankverwaltungssystems (DBMS) auf einer Amazon EC2-Instance | Hosten einer nicht relationalen Datenbanklösung auf einer EC2-Instance |

## Amazon RDS

Amazon Relational Database Service (Amazon RDS) bietet Ihnen Zugriff auf die Funktionen einer vertrauten MySQL-, PostgreSQL-, Oracle- und Microsoft SQL Server-Datenbank-Engine. Der Code, die Anwendungen und Tools, die Sie bereits verwenden, können mit Amazon RDS verwendet werden. Amazon RDS führt automatisch Patches der Datenbank-Software durch und erstellt Backups Ihrer Datenbank. Diese Backups werden für einen vom Benutzer definierten Zeitraum aufbewahrt. Es unterstützt auch die zeitpunktbezogene Wiederherstellung. Ein weiterer Vorteil für Sie ist die flexible Skalierung der Rechenressourcen oder der Speicherkapazität für die Instance Ihrer relationalen Datenbank über einen einzigen API-Aufruf.

Amazon RDS Multi-AZ-Bereitstellungen erhöhen Ihre Datenbankverfügbarkeit und schützen Ihre Datenbank vor ungeplanten Ausfällen. Amazon RDS Read Replicas bieten schreibgeschützte Replikate Ihrer Datenbank, sodass Sie über die Kapazität einer einzelnen Datenbankbereitstellung hinaus für leseintensive Datenbank-Workloads skalieren können. Wie bei allen AWS-Services sind keine Vorabinvestitionen erforderlich, und Sie zahlen nur für die Ressourcen, die Sie nutzen.

## Hosten eines relationalen Datenbankverwaltungssystems (RDBMS) auf einer Amazon EC2-Instance

Zusätzlich zum verwalteten Amazon RDS-Angebot können Sie RDBMS Ihrer Wahl (wie MySQL, Oracle, SQL Server oder DB2) auf einer EC2-Instance installieren und selbst verwalten. AWS-Kunden, die eine Datenbank auf Amazon EC2 hosten, verwenden erfolgreich eine Vielzahl von Primär-/Standby- und Replikationsmodellen, einschließlich Spiegelung für schreibgeschützte Kopien und Protokollversand für immer einsatzbereite passive Slaves.

Wenn Sie Ihre eigene Datenbanksoftware direkt in Amazon EC2 verwalten, sollten Sie auch die Verfügbarkeit von fehlertolerantem und beständigem Speicher in Betracht ziehen. Zu diesem Zweck empfehlen wir, dass Datenbanken, die auf Amazon EC2 ausgeführt werden, Amazon Elastic Block Store (Amazon EBS) -Volumes verwenden, die dem netzwerkgebundenen Speicher ähneln.

Für EC2-Instances, die eine Datenbank ausführen, sollten Sie alle Datenbankdaten und Protokolle auf EBS-Volumes platzieren. Diese bleiben auch dann verfügbar, wenn der Datenbank-Host ausfällt. Diese Konfiguration ermöglicht ein einfaches Failover-Szenario, in dem eine neue EC2-Instance gestartet werden kann, wenn ein Host ausfällt, und die vorhandenen EBS-Volumes an die neue Instance angehängt werden können. Die Datenbank kann dann dort weitermachen, wo sie aufgehört hat.

EBS-Volumes bieten automatisch Redundanz innerhalb der Availability Zone. Wenn die Leistung eines einzelnen EBS-Volumes für die Anforderungen Ihrer Datenbanken nicht ausreicht, kann Striping auf Volumes angewendet werden, um die Leistung von Eingabe-/Ausgabevorgängen pro Sekunde (IOPS) für Ihre Datenbank zu erhöhen.

Für anspruchsvolle Workloads können Sie auch EBS-bereitgestellte IOPS verwenden, in denen Sie die erforderlichen IOPS angeben. Wenn Sie Amazon RDS verwenden, verwaltet der Dienst seinen eigenen Speicher, sodass Sie sich auf die Verwaltung Ihrer Daten konzentrieren können.

## Nicht-relationale Datenbanken

AWS unterstützt nicht nur relationale Datenbanken, sondern bietet auch eine Reihe verwalteter, nicht relationaler Datenbanken:

- Amazon DynamoDB ist ein vollständig verwalteter NoSQL-Datenbank-Service, der schnelle und planbare Leistung mit nahtloser Skalierbarkeit bereitstellt. Mit derAWS Management Console oder der DynamoDB-API können Sie die Kapazität ohne Ausfallzeiten oder Leistungseinbußen nach oben oder unten skalieren. Mit Amazon DynamoDB lagern Sie die administrativen Aufgaben in Bezug auf Betrieb und Skalierung verteilter Datenbanken an AWS aus. So müssen Sie sich keine Gedanken um die Bereitstellung von Hardware, um deren Einrichtung und Konfiguration, um die Replikation, um Software-Patches oder um die Cluster-Skalierung machen.
- Amazon DocumentDB (mit MongoDB-Kompatibilität) ist ein Datenbankservice, der speziell für die Verwaltung von JSON-Daten in großem Umfang entwickelt wurde, vollständig verwaltet und in AWS integriert ist und eine hohe Beständigkeit für Unternehmen bietet.
- Amazon Keyspaces (für Apache Cassandra) ist ein skalierbarer, hochverfügbarer und verwalteter Apache Cassandra-kompatibler Datenbankservice. Mit Amazon Keyspaces können Sie Ihre

Cassandra-Workloads in AWS mit demselben Anwendungscode und denselben Entwicklertools ausführen, die Sie heute verwenden.

- Amazon Neptune ist ein schneller, zuverlässiger, vollständig verwalteter Graph-Datenbankservice, mit dem es ganz einfach ist, Anwendungen zu erstellen und auszuführen, die mit stark verbundenen Datensätzen arbeiten. Der Kern von Amazon Neptune ist eine speziell entwickelte, hochleistungsfähige Graph-Datenbankengine, die für die Speicherung von Milliarden von Beziehungen und die Abfrage des Graphen mit einer Latenzzeit im Millisekundenbereich optimiert ist.
- Amazon Quantum Ledger Database (Amazon QLDB) (QLDB) ist eine vollständig verwaltete Ledger-Datenbank, die ein transparentes, unveränderliches und kryptografisch überprüfbares Transaktionsprotokoll im Besitz eines zentralen vertrauenswürdigen Ausstellers bereitstellt. QLDB kann verwendet werden, um jede einzelne Änderung der Anwendungsdaten zu verfolgen, und führt eine vollständige und überprüfbare Historie der Änderungen im Laufe der Zeit.
- Amazon Timestream ist ein schneller, skalierbarer und serverloser Zeitreihen-Datenbankservice für IoT und Betriebsanwendungen, der die Speicherung und Analyse von Billionen Ereignissen pro Tag ermöglicht – bis zu tausendfach schneller und bis zu einem Zehntel der Kosten im Vergleich mit relationalen Datenbanken.

Darüber hinaus können Sie Amazon EC2 verwenden, um andere nicht relationale Datenbanktechnologien zu hosten, mit denen Sie möglicherweise arbeiten.

## Speicherung und Sicherung von Daten und Komponenten

Innerhalb der AWS Cloud gibt es zahlreiche Optionen für Speicherung, Zugriff und Sicherung von Webanwendungs-Daten und -Komponenten. Amazon S3 bietet einen hochverfügbaren und redundanten Objektspeicher. Amazon S3 ist eine hervorragende Speicherlösung für eher statische und sich langsam ändernde Objekte, wie Bilder, Videos und andere statische Medien. Amazon S3 unterstützt auch Edge-Caching und Streaming dieser Komponenten, indem es mit CloudFront interagiert.

Für angehängten Dateisystem-ähnlichen Speicher können EC2-Instances EBS-Volumes angehängt haben. Diese verhalten sich wie installierbare Festplatten zum Ausführen von EC2-Instances. Amazon EBS eignet sich hervorragend für Daten, auf die als Blockspeicher zugegriffen werden muss und erfordert Persistenz über die Betriebsdauer der ausgeführten Instanz hinaus, beispielsweise Datenbankpartitionen und Anwendungsprotokolle.

Sie haben nicht nur eine von der EC2-Instance unabhängige Lebensdauer, sondern können auch Snapshots von EBS-Volumes erstellen und diese in Amazon S3 speichern. Da EBS-Snapshots nur Änderungen seit dem vorherigen Snapshot sichern, können häufigere Snapshots die Snapshot-Zeiten verkürzen. Sie können auch einen EBS-Snapshot als Grundlage für die Replikation von Daten über mehrere EBS-Volumes hinweg verwenden und diese Volumes an andere laufende Instances anhängen.

EBS-Volumes können bis zu 16 TB groß sein, und mehrere EBS-Volumes können für noch größere Volumes oder für eine höhere Eingangs-/Ausgangsleistung (I/O) aufgeteilt werden. Um die Leistung Ihrer I/O-intensiven Anwendungen zu maximieren, können Sie bereitgestellte IOPS-Volumes verwenden. Volumes mit bereitgestellten IOPS sind darauf ausgelegt, die Anforderungen E/A-intensiver Workloads zu erfüllen, insbesondere von Datenbank-Workloads, die bei E/A-Durchsätzen mit zufälligen Zugriffen empfindlich auf die Speicherleistung und Konsistenz reagieren.

Sie geben eine IOPS-Rate an, wenn Sie das Volume erstellen, und Amazon EBS-Rückstellungen, die für die Lebensdauer des Volumens gelten. Amazon EBS unterstützt derzeit IOPS pro Volume von maximal 16000 (für alle Instance-Typen) bis zu 64.000 (für Instances, die auf Nitro System aufbauen). Sie können mehrere Volumes per Stripe-Verfahren zusammenbringen, um Ihrer Anwendung Tausende von E/A-Vorgängen pro Sekunde pro Instance zu bieten. Abgesehen davon können Sie für einen höheren Durchsatz und geschäftskritische Workloads, die eine Latenz von weniger als einer Millisekunde erfordern, den io2 Block Express-Volume-Typ verwenden, der bis zu 256.000 IOPS bei einer maximalen Speicherkapazität von 64 TB unterstützt.

## Automatische Skalierung der Flotte

Einer der Hauptunterschiede zwischen der AWS Cloud-Architektur und dem traditionellen Hosting-Modell besteht darin, dass AWS die Webanwendungsflotte bei Bedarf automatisch skalieren kann, um Änderungen des Datenverkehrs zu bewältigen. Beim herkömmlichen Hostingmodell werden im Allgemeinen Verkehrsprognosemodelle verwendet, um die Hosts vor dem erwarteten Verkehr einzurichten. In AWS können Instances gemäß einer Reihe von Auslösern für die Hin- und Rückwärtsskalierung der Flotte spontan bereitgestellt werden.

Der Auto Scaling-Service kann verwendet werden, um Kapazitätsgruppen von Servern zu erstellen, die auf Anforderung wachsen oder schrumpfen können. Auto Scaling arbeitet auch direkt mit CloudWatch für Metrikdaten und mit Elastic Load Balancing zusammen, um Hosts zur Lastverteilung hinzuzufügen und zu entfernen. Wenn die Webserver beispielsweise während einer bestimmten Zeitperiode mehr als 80% CPU-Nutzung melden, könnte schnell ein zusätzlicher Webserver

implementiert und dann zur Lastenverteilung hinzugefügt werden, um sofort in die Drehung der Lastenverteilung einbezogen zu werden.

Wie im AWS Webhosting-Architekturmodell gezeigt, können Sie mehrere Auto-Scaling-Gruppen für verschiedene Ebenen der Architektur erstellen, damit jede Ebene der Architektur unabhängig skaliert werden kann. Beispielsweise kann die Auto-Scaling-Gruppe des Webservers als Reaktion auf Änderungen der Netzwerk-E/A ein Skalieren auslösen, während die Auto-Scaling-Gruppe des Anwendungsservers entsprechend der CPU-Auslastung ein- und ausskaliert. Sie können Mindest- und Höchstwerte festlegen, um die Verfügbarkeit rund um die Uhr sicherzustellen und die Nutzung innerhalb einer Gruppe zu begrenzen.

Auto Scaling-Auslöser können sowohl so eingestellt werden, dass sie auf einer bestimmten Ebene wachsen als auch die gesamte Flotte verkleinern, um die Ressourcenauslastung an den tatsächlichen Bedarf anzupassen. Zusätzlich zum Auto Scaling-Service können Sie Amazon EC2-Flotten direkt über die Amazon EC2-API skalieren, die das Starten, Beenden und Prüfen von Instances ermöglicht.

## Zusätzliche Sicherheitsfunktionen

Die Zahl und Komplexität von Distributed Denial of Service (DDoS)-Angriffen nimmt zu. Traditionell sind diese Angriffe schwer abzuwehren. Sie sind häufig sowohl in Bezug auf die Minderung als auch auf den Stromverbrauch sowie in Bezug auf die Opportunitätskosten durch verlorene Besuche auf Ihrer Website während des Angriffs kostspielig. Es gibt eine Reihe von AWS-Faktoren und -Services, die Ihnen helfen können, sich vor solchen Angriffen zu schützen. Eine davon ist die Größe des AWS-Netzwerks. Die AWS-Infrastruktur ist ziemlich groß und ermöglicht es Ihnen, unsere Größe zu nutzen, um Ihren Schutz zu optimieren. Verschiedene Services, darunter Elastic Load Balancing, Amazon CloudFront und Amazon Route 53, können Ihre Webanwendung als Reaktion auf einen starken Anstieg des Datenverkehrs effektiv skalieren.

Insbesondere die Infrastrukturschutzdienste helfen Ihnen bei Ihrer Verteidigungsstrategie:

- AWS Shield ist ein verwalteter DDoS-Schutzdienst, der zum Schutz vor verschiedenen Formen von DDoS-Angriffsvektoren beiträgt. Das Standardangebot von AWS Shield ist kostenlos und automatisch in Ihrem gesamten Konto aktiv. Dieses Standardangebot hilft bei der Abwehr der häufigsten Netzwerk- und Transportebenenangriffe. Darüber hinaus bietet das erweiterte Angebot einen höheren Schutz vor Ihrer Webanwendung, indem es Ihnen nahezu in Echtzeit Einblick in einen laufenden Angriff bietet und sich auf höheren Ebenen in die zuvor genannten Dienste integriert. Außerdem erhalten Sie Zugriff auf das AWS DDoS Response Team (DRT), um umfangreiche und ausgeklügelte Angriffe auf Ihre Ressourcen abzuwehren.

- AWS WAF (Web Application Firewall) wurde entwickelt, um Ihre Webanwendungen vor Angriffen zu schützen, die die Verfügbarkeit oder Sicherheit gefährden oder auf andere Weise übermäßige Ressourcen verbrauchen können. AWS WAF arbeitet im Einklang mit CloudFront oder Application Load Balancer zusammen mit Ihren benutzerdefinierten Regeln, um Angriffe wie Cross-Site-Scripting, SQL-Injection und DDoS abzuwehren. Wie bei den meisten AWS-Services wird eine API mit vollem FunktionsumfangAWS WAF geliefert, mit der Sie die Erstellung und Bearbeitung von Regeln für Ihre AWS WAF-Instanz automatisieren können, wenn sich Ihre Sicherheitsanforderungen ändern.
- AWS Firewall Manager ist ein Sicherheitsverwaltungsdienst, mit dem Sie Firewall-Regeln für Ihre Konten und Anwendungen zentral konfigurieren und verwalten können AWS Organizations. Wenn neue Anwendungen erstellt werden, ist es einfach durch AWS Firewall Manager neue Anwendungen und Ressourcen in Übereinstimmung mit den Vorschriften zu bringen, indem ein gemeinsamer Satz von Sicherheitsregeln durchgesetzt wird.

## Failover mit AWS

Ein weiterer wichtiger Vorteil von AWS gegenüber herkömmlichem Webhosting sind die Availability Zones, die Ihnen einfachen Zugriff auf redundante Bereitstellungsorte ermöglichen. Availability Zones sind physisch unabhängige Standorte, die so aufgebaut sind, dass sie von Fehlern in anderen Availability Zones nicht betroffen sind. Sie stellen anderen Availability Zones in der gleichen AWS-Region kostengünstige Netzwerkkonnektivität mit geringer Latenz zur Verfügung. Wie das Diagramm der AWS-Webhosting-Architektur zeigt, empfiehlt AWS, EC2-Hosts über mehrere Availability Zones hinweg bereitzustellen, um Ihre Webanwendung fehlertoleranter zu gestalten.

Es ist wichtig sicherzustellen, dass Vorkehrungen zur Migration einzelner Zugangspunkte auf Availability Zones im Störungsfall getroffen werden. Sie sollten beispielsweise einen Datenbank-Standby in einer zweiten Availability Zone einrichten, damit die Persistenz der Daten auch in einem unwahrscheinlichen Ausfallszenario konsistent und hochverfügbar bleibt. Sie können dies auf Amazon EC2 oder Amazon RDS mit einem Klick auf eine Schaltfläche tun.

Während beim Verschieben einer vorhandenen Webanwendung in die AWS Cloud häufig einige architektonische Änderungen erforderlich sind, gibt es erhebliche Verbesserungen der Skalierbarkeit, Zuverlässigkeit und Kosteneffizienz, sodass sich die Verwendung der AWS Cloud lohnt. Im nächsten Abschnitt werden diese Verbesserungen erörtert.

# Wichtige Überlegungen bei der Verwendung von AWS für Webhosting

Es gibt einige wichtige Unterschiede zwischen der AWS Cloud und einem herkömmlichen Webanwendungshosting-Modell. Der vorherige Abschnitt wies auf zahlreiche wichtige Bereiche hin, die Sie bei der Implentierung einer Webanwendung in der Cloud berücksichtigen sollten. In diesem Abschnitt werden einige der wichtigsten architektonischen Veränderungen aufgezeigt, die Sie berücksichtigen müssen, wenn Sie eine Anwendung in die Cloud bringen.

## Keine physischen Netzwerkgeräte mehr

Sie können keine physischen Netzwerkgeräte in AWS bereitstellen. Firewalls, Router und Lastenverteilung für Ihre AWS-Anwendungen können sich nicht mehr auf physischen Geräten befinden, sondern müssen durch Softwarelösungen ersetzt werden. Es gibt eine Vielzahl von Softwarelösungen in Unternehmensqualität, sei es für die Lastenverteilung oder den Aufbau einer VPN-Verbindung. Dies stellt keine Beschränkung für Anwendungen dar, die in der AWS Cloud ausgeführt werden können. Die Architektur Ihrer Anwendung wird sich aber ändern, wenn Sie diese Geräte heute nutzen.

## Überall Firewalls

Wo es früher eine einfache demilitarisierte Zone (DMZ) und dann offene Kommunikation zwischen Ihren Hosts in einem herkömmlichen Hostingmodell gab, setzt AWS ein Modell mit höherer Sicherheit um, in dem jeder Host gesperrt ist. Einer der Schritte bei der Planung einer AWS-Bereitstellung ist die Analyse des Datenverkehrs zwischen Hosts. Diese Analyse wird Entscheidungen darüber leiten, welche Ports genau geöffnet werden müssen. Sie können Sicherheitsgruppen für jeden Hosttyp in Ihrer Architektur erstellen. Sie können auch eine Vielzahl einfacher und gestufter Sicherheitsmodelle erstellen, um den minimalen Zugriff zwischen Hosts innerhalb Ihrer Architektur zu ermöglichen. Die Verwendung von Netzwerkzugriffskontrolllisten in Amazon VPC kann dazu beitragen, Ihr Netzwerk auf Subnetzebene zu sperren.

## Bedenken Sie die Verfügbarkeit mehrerer Rechenzentren

Stellen Sie sich Availability Zones innerhalb einer AWS-Region als mehrere Rechenzentren vor. EC2-Instances in unterschiedlichen Availability Zones sind logisch wie physisch getrennt und bieten

ein einfaches Modell für hohe Verfügbarkeit und Zuverlässigkeit Ihrer Anwendung auf mehreren Rechenzentren. Amazon VPC als regionaler Service ermöglicht es Ihnen, Availability Zones zu nutzen und gleichzeitig alle Ihre Ressourcen im selben logischen Netzwerk zu halten.

## Hosts werden als ephemer und dynamisch behandelt

Die wahrscheinlich wichtigste Veränderung beim Aufbau der Architektur Ihrer AWS-Anwendung ist die Tatsache, dass EC2-Hosts als ephemer und dynamisch betrachtet werden sollen. Jede für die AWS Cloud erstellte Anwendung sollte nicht davon ausgehen, dass ein Host immer verfügbar sein wird, und sollte mit dem Wissen entworfen werden, dass alle Daten in den EC2-Instant Stores verloren gehen, wenn eine EC2-Instance ausfällt.

Wenn ein neuer Host hochgefahren wird, sollten Sie keine Annahmen über die IP-Adresse oder den Standort innerhalb einer Availability Zone des Hosts treffen. Ihr Konfigurationsmodell muss flexibel sein, und Ihr Ansatz zum Bootstrapping eines Hosts muss die Dynamik der Cloud berücksichtigen. Diese Techniken sind für den Aufbau und Betrieb einer hoch skalierbaren und fehlertoleranten Anwendung von entscheidender Bedeutung.

## Bedenken Sie Container und Serverless

Dieses Whitepaper konzentriert sich hauptsächlich auf eine herkömmlichere Webarchitektur. Erwägen Sie jedoch, Ihre Webanwendungen zu modernisieren, indem Sie auf Container und Serverless-Technologien umsteigen und Dienste wie AWS Fargate und AWS Lambda nutzen, damit Sie die Verwendung von virtuelle Maschinen zur Ausführung von Rechenaufgaben einsetzen können. Beim Serverless Computing werden Infrastrukturverwaltungsaufgaben wie Kapazitätsbereitstellung und Patching von AWS übernommen, sodass Sie flexiblere Anwendungen erstellen können, mit denen Sie schneller innovativ sein und auf Änderungen reagieren können.

## Bedenken Sie automatisierte Bereitstellung

- Amazon Lightsail ist ein einfach zu bedienender virtueller privater Server (VPS), der Ihnen alles bietet, was Sie für die Erstellung einer Anwendung oder Website benötigen, sowie einen kostengünstigen, monatlichen Plan. Lightsail ist ideal für einfachere Workloads, schnelle Implementierungen und den Einstieg in AWS. Es ist konzipiert Ihnen zu helfen, klein anzufangen und dann mit zunehmendem Wachstum zu skalieren.
- AWS Elastic Beanstalk ist ein benutzerfreundlicher Service für die Bereitstellung und Skalierung von Webanwendungen und -Services, die mit Java, .NET, PHP, Node.js, Python, Ruby, Go und

Docker auf vertrauten Servern wie Apache, NGINX, Passenger und IIS entwickelt werden. Sie laden Ihren Code einfach hoch und Elastic Beanstalk übernimmt automatisch die Bereitstellung, Kapazitätsbereitstellung, Lastenverteilung, automatische Skalierung und Statusüberwachung der Anwendung. Gleichzeitig erhalten Sie mit Elastic Beanstalk vollständige Kontrolle über die AWS-Ressourcen hinter Ihrer Anwendung und können jederzeit auf die zugrunde liegenden Ressourcen zugreifen.

- AWS App Runner ist ein vollständig verwalteter Service, mit dem Ihre Entwickler schnell und skalierbar containerisierte Webanwendungen und APIs bereitstellen können, ohne Erfahrungen im Infrastrukturbereich zu besitzen. Sie beginnen mit Ihrem Quellcode oder einem Container-Image. App Runner führt die Entwicklung und Bereitstellung der Webanwendung automatisch aus und nimmt eine Lastenverteilung des Datenverkehrs mit Verschlüsselung vor. App Runner skaliert darüber hinaus automatisch aufwärts oder abwärts, um den Anforderungen Ihres Datenverkehrs zu entsprechen.
- AWS Amplify ist eine Reihe von Tools und Services, die zusammen oder einzeln verwendet werden können, um Front-End-Web- und Mobilentwickler beim Aufbau skalierbarer Full-Stack-Anwendungen Powered by AWS zu unterstützen. Mit Amplify können Sie App-Backends konfigurieren und Ihre App in Minutenschnelle verbinden, statische Web-Apps mit wenigen Klicks bereitstellen und App-Inhalte problemlos außerhalb der AWS-Konsole verwaltenAWS Management Console.

# Fazit und Mitwirkende

## Abschluss

Es gibt zahlreiche architektonische und konzeptionelle Überlegungen, wenn Sie erwägen, Ihre Webanwendung zur AWS Cloud zu migrieren. Die Nutzen einer kostengünstigen, hoch skalierbaren und fehlertoleranten Infrastruktur, die mit Ihrem Unternehmen wächst, übertreffen die Bemühungen bei der Migration zur AWS Cloud bei weitem.

## Mitwirkende

Dieses Dokument ist unter der Mitarbeit folgender Personen und Unternehmen entstanden:

- Amir Khairalomoum, Leitender Lösungsarchitekt, AWS
- Dinesh Subramani, Leitender Lösungsarchitekt, AWS
- Jack Hemion, Leitender Lösungsarchitekt, AWS
- Jatin Joshi, Cloud-Support-Techniker, AWS
- Jorge Fonseca, Leitender Lösungsarchitekt, AWS
- Shinduri K S, Lösungsarchitekt, AWS

# Weitere Informationen

- Bereitstellen einer Django-basierten Anwendung in Amazon Lightsail
- Bereitstellung einer Drupal-Website mit hoher Verfügbarkeit für Elastic Beanstalk
- Bereitstellen einer PHP-Anwendung mit hoher Verfügbarkeit für Elastic Beanstalk
- Bereitstellen einer Node.js-Anwendung mit DynamoDB in Elastic Beanstalk
- Erste Schritte mit Linux-Webanwendungen in der AWS Cloud
- Hosten einer statischen Website
- Hosten einer statischen Website mit Amazon S3
- Tutorial: Bereitstellen einer ASP.NET Core-Anwendung mit Elastic Beanstalk
- Tutorial: Erstellen und Bereitstellen einer .NET-Beispielanwendung mit Elastic Beanstalk

# Dokumentversionen

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

| Update-Historie-Änderung | Update-Historie-Beschreibung | Update-Historie-Datum |
|---|---|---|
| Whitepaper aktualisiert | Mehrere Abschnitte und Diagramme wurden mit neuen Diensten, Funktionen und aktualisierten Service-Limits aktualisiert. | 20. August 2021 |
| Whitepaper aktualisiert | Die Symbolbezeichnung für „Caching with ElastiCache“ (Caching mit ElastiCache) wurde in Abbildung 3 aktualisiert. | 29. September 2019 |
| Whitepaper aktualisiert | Es wurden mehrere Abschnitte hinzugefügt und für neue Dienste aktualisiert. Aktualisierte Diagramme zur zusätzlichen Übersichtlichkeit und Dienste. Hinzufügen von VPC als Standardnetzwerkmethode in AWS in „Network Management“ (Netzwerk-Management). Abschnitt über DDoS-Schutz und -Minderung in „Additional Security Features“ (Zusätzliche Sicherheitsfunktionen) hinzugefügt. Es wurde ein kleiner Abschnitt über Serverless-Architekturen für das Webhosting hinzugefügt. | 1. Juli 2017 |

| | | |
|---|---|---|
| Whitepaper aktualisiert | Mehrere Abschnitte wurden zur Verbesserung der Übersichtlichkeit aktualisiert. Diagramme zur Verwendung von AWS-Symbolen aktualisiert. Abschnitt „Managing Public DNS“ (Öffentliche DNS verwalten) für Details zu Amazon Route 53 hinzugefügt. Der Abschnitt „Finding Other Hosts and Services“ (Suche nach anderen Hosts und Services) wurde zur besseren Übersicht aktualisiert. Der Abschnitt „Database Configuration, Backup, and Failover“ (Datenbankkonfiguration, Sicherung und Failover) wurde aus Gründen der Übersichtlichkeit und DynamoDB aktualisiert. Der Abschnitt „Storage and Backup of Data and Assets“ (Speicherung und Sicherung von Daten und Assets) wurde erweitert, um von EBS bereitgestellte IOPS-Volumes abzudecken. | 1. September 2012 |
| Erste Veröffentlichung | Whitepaper veröffentlicht. | 1. Mai 2010 |

# Hinweise

Dieses Dokument wird nur zu Informationszwecken zur Verfügung gestellt. Es stellt das aktuelle Produktangebot und die Praktiken von AWS zum Erstellungsdatum dieses Dokuments dar. Änderungen vorbehalten. Kunden sind verantwortlich für ihre eigene Interpretation der in diesem Dokument zur Verfügung gestellten Informationen und für die Nutzung der AWS-Produkte oder -Services. Diese werden alle ohne Mängelgewähr und ohne jegliche Garantie, weder ausdrücklich noch stillschweigend, bereitgestellt. Dieses Dokument gibt keine Garantien, Gewährleistungen, vertragliche Verpflichtungen, Bedingungen oder Zusicherungen von AWS, seinen Partnern, Zulieferern oder Lizenzgebern. Die Verantwortung und Haftung von AWS gegenüber seinen Kunden werden durch AWS-Vereinbarungen geregelt. Dieses Dokument gehört, weder ganz noch teilweise, nicht zu den Vereinbarung von AWS mit seinen Kunden und ändert diese Vereinbarungen auch nicht.

© 2019 Amazon Web Services, Inc. bzw. Tochtergesellschaften des Unternehmens. Alle Rechte vorbehalten.