



Architektur-Diagramme

Architektur von Webanwendungen auf AWS



Architektur von Webanwendungen auf AWS: Architektur-Diagramme

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und die Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Architektur der Web-App i

Architektur der Webanwendung auf AWS 1

Weitere Informationen 2

Verlauf des Diagramms 2

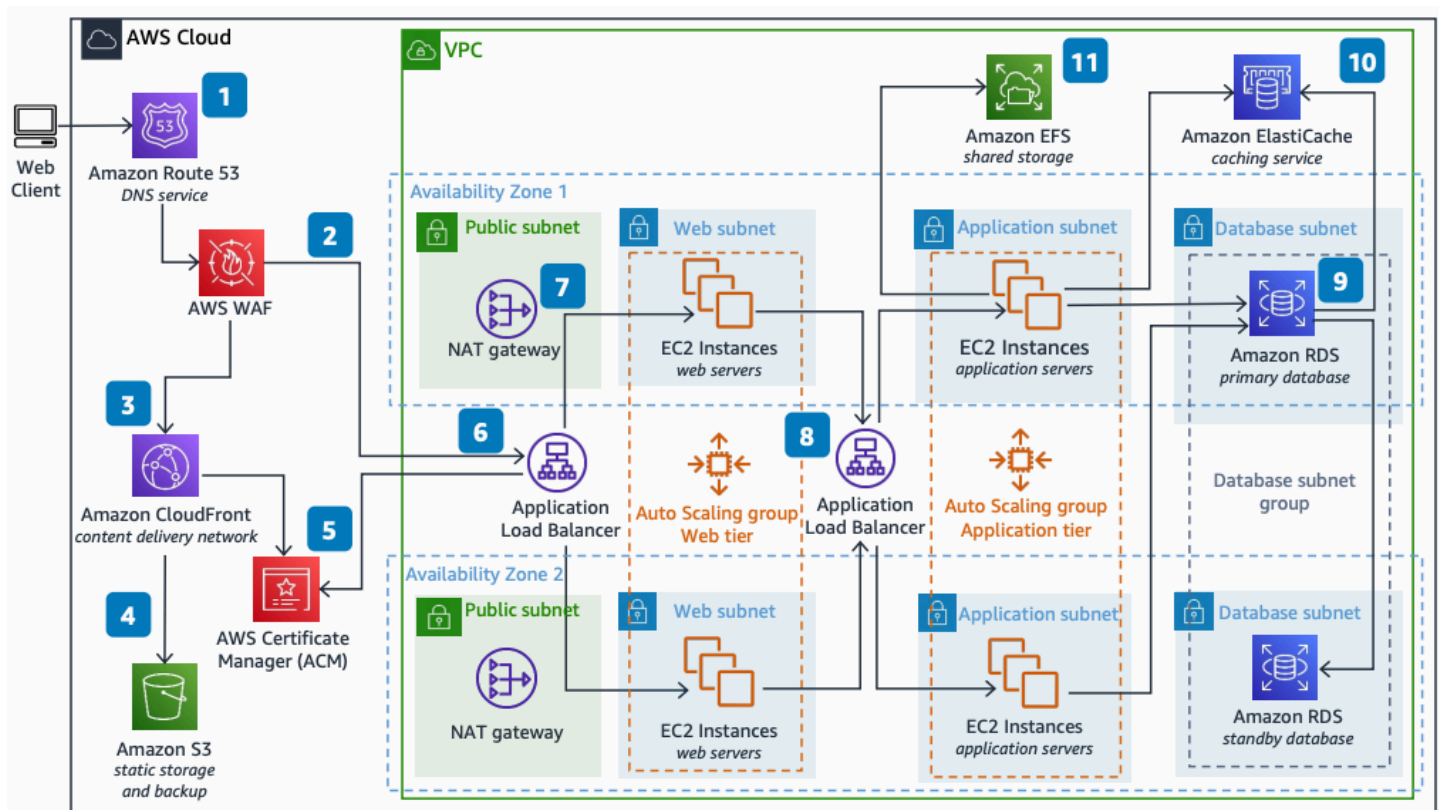
..... iv

Architektur von Webanwendungen auf AWS

Veröffentlichungsdatum: 19. November 2021 ([Geschichte des Diagramms](#))

Diese Architektur zeigt, wie Sie eine klassische Webanwendung auf hosten können AWS. Sie verwenden mehrere Availability Zones für hohe Verfügbarkeit, Application Load Balancer für die Verteilung des Datenverkehrs und Managed Services für Datenbank, Caching und gemeinsam genutzten Speicher.

Architektur der Webanwendung auf AWS



Die folgenden Schritte beschreiben die Architektur:

1. Leiten Sie den Datenverkehr vom Webclient auf der Grundlage des Anforderungspaths für statische und dynamische Inhalte mithilfe von [Route 53 weiter](#).
2. Verwenden Sie ein Content Delivery Network (CDN) wie [Amazon CloudFront](#), um die Latenz bei der Bereitstellung Ihrer statischen Inhalte zu reduzieren.
3. Verwenden Sie [Amazon Simple Storage Service](#), um statische Inhalte und Backups zu speichern.

4. Schützen Sie Ihre Webanwendung mit einer Webanwendungs-Firewall wie AWS WAF vor gängigen Web-Exploits.
5. Vereinfachen Sie die Verwaltung Ihrer SSL-Zertifikate mit dem AWS Certificate Manager (ACM).
6. Verwenden Sie einen mit dem Internet verbundenen Application Load Balancer, um den Webdatenverkehr auf Ihre Webserver zu verteilen, die über mehrere Availability Zones verteilt sind.
7. Verwenden Sie NAT-Gateways in jedem öffentlichen Subnetz, um [Amazon Elastic Compute](#) Cloud-Instances in privaten Subnetzen den Zugriff auf das Internet zu ermöglichen.
8. Verwenden Sie einen internen Application Load Balancer, um den Datenverkehr auf Ihre Anwendungsserver zu verteilen, die über mehrere Availability Zones verteilt sind.
9. Vereinfachen Sie Ihre Datenbankverwaltung, indem Sie Ihre Datenbankebene in [Amazon RDS](#) ausführen.
10. Wenn die Datenbankzugriffsmuster leseintensiv sind, nutzen Sie eine Caching-Ebene wie Amazon. [ElastiCache](#)
11. Erwägen Sie, einen Dienst für gemeinsam genutzten Speicher wie Amazon Elastic File System (Amazon EFS) zu verwenden, wenn Ihre Server auf gemeinsam genutzte Dateien zugreifen.

Weitere Informationen

Weitere Informationen finden Sie in den folgenden Ressourcen:

- [AWS Architektur-Ikonen](#)
- [AWS Zentrum für Architektur](#)
- [AWS Well-Architected](#)

Geschichte des Diagramms

Abonnieren Sie den RSS-Feed, um über Aktualisierungen dieses Referenzarchitekturdiagramms informiert zu werden.

Änderung

Beschreibung

Datum

Erste Veröffentlichung

Das Referenzarchitekturdiagramm wurde zuerst veröffentlicht.

19. November 2021

 Note

Um RSS-Updates zu abonnieren, muss ein RSS-Plugin für den von Ihnen verwendeten Browser aktiviert sein.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.