



Architektur-Diagramme

Serverlose Strategie für dynamische Preisgestaltung



Serverlose Strategie für dynamische Preisgestaltung: 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

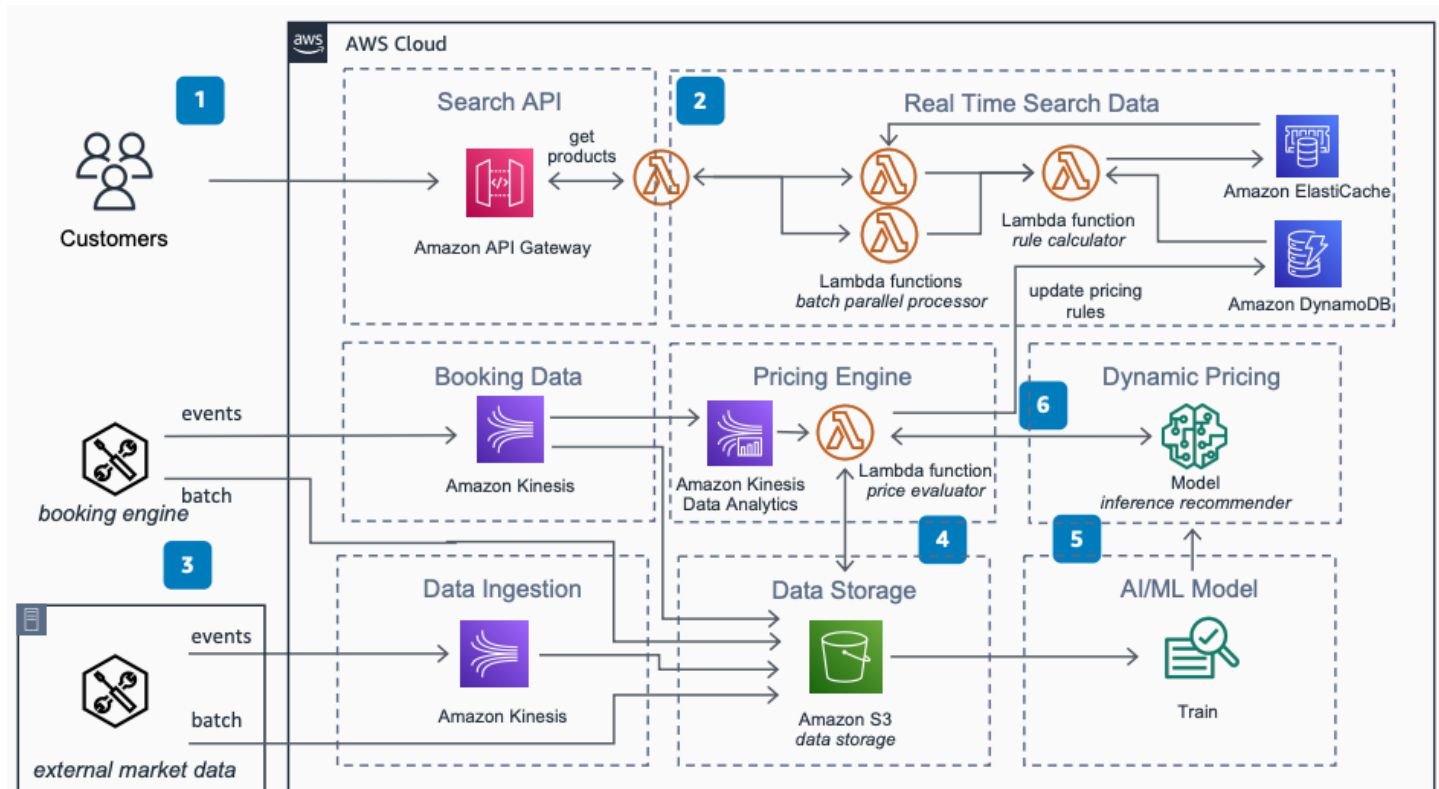
Dynamische Preisgestaltung	i
Serverlose Strategie für dynamische Preisgestaltung	1
Weitere Informationen	2
Verlauf des Diagramms	2
.....	iv

Serverlose Strategie für dynamische Preisgestaltung

Veröffentlichungsdatum: 28. Juni 2021 () [Geschichte des Diagramms](#)

Diese Architektur zeigt eine serverlose Strategie für dynamische Preisgestaltung. Für Buchungen, die auf Datumsbereichen basieren, können Sie Preise parallel erstellen und diese anhand von Faktoren wie Datum, Dauer und Anzahl der Personen dynamisch berechnen. Anschließend werden vergleichbare Produkte sortiert mit Antwortzeiten unter einer Sekunde angezeigt und die Preise werden auf der Grundlage der aktuellen Marktbedingungen aktualisiert.

Serverlose Strategie für dynamische Preisgestaltung



Die folgenden Schritte beschreiben die Architektur:

1. Sie geben Sucheingaben ein, um eine sortierte Liste von Produkten zu erhalten, die nach Preis und anderen Attributen gefiltert sind. Die Daten werden über das [Amazon API Gateway](#) [gesendet](#), das mehrere [AWS Lambda](#) Funktionen zur parallelen Bearbeitung von Preisanfragen für Preisabfragen in Echtzeit aufruft.

2. Die Lambda-Funktionen des Batch-Parallelprozessors rufen die Regelrechnerfunktion auf, um die neuesten Produktregeln von [Amazon DynamoDB abzurufen](#). Das Produkt wird anhand der Such-ID berechnet und bei Amazon gespeichert. [ElastiCache](#) Nach Abschluss der Batch-Funktionen werden alle Ergebnisse für die Such-ID aus dem Cache zurückgegeben.
3. Historische Daten aus der Buchungsmaschine und externe Marktdaten liefern Informationen, sodass die Preisfindungsmaschine über ausreichend Daten verfügt, um die Preise in Echtzeit zu aktualisieren.
4. Die Ereignisse der Buchungsmaschine werden auf [Amazon Kinesis](#) erfasst und überwacht. Amazon Kinesis Data Analytics fragt den Datenstrom ab und löst die Lambda-Funktion des Preisbewerters aus, wenn eine Änderung festgestellt wird. Ereignisse werden in [Amazon Simple Storage Service](#) mit unterstützenden historischen Batchdaten gespeichert.
5. Ein [Amazon SageMaker AI](#) Train-Modell verwendet historische Buchungsdaten und erweiterte historische Daten, um das Modell im Hinblick auf Preisempfehlungen zu trainieren.
6. Mithilfe eines SageMaker KI-Inferenz-Empfehlungsmodells erhält die Pricing-Engine neue Preisempfehlungen, die auf den Marktbedingungen basieren.

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.	28. Juni 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.