



Schémas d'architecture

Stratégie sans serveur pour une tarification dynamique



Stratégie sans serveur pour une tarification dynamique: Schémas d'architecture

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Tarification dynamique i

Stratégie sans serveur pour une tarification dynamique 1

Suggestions de lecture 2

Historique du diagramme 2

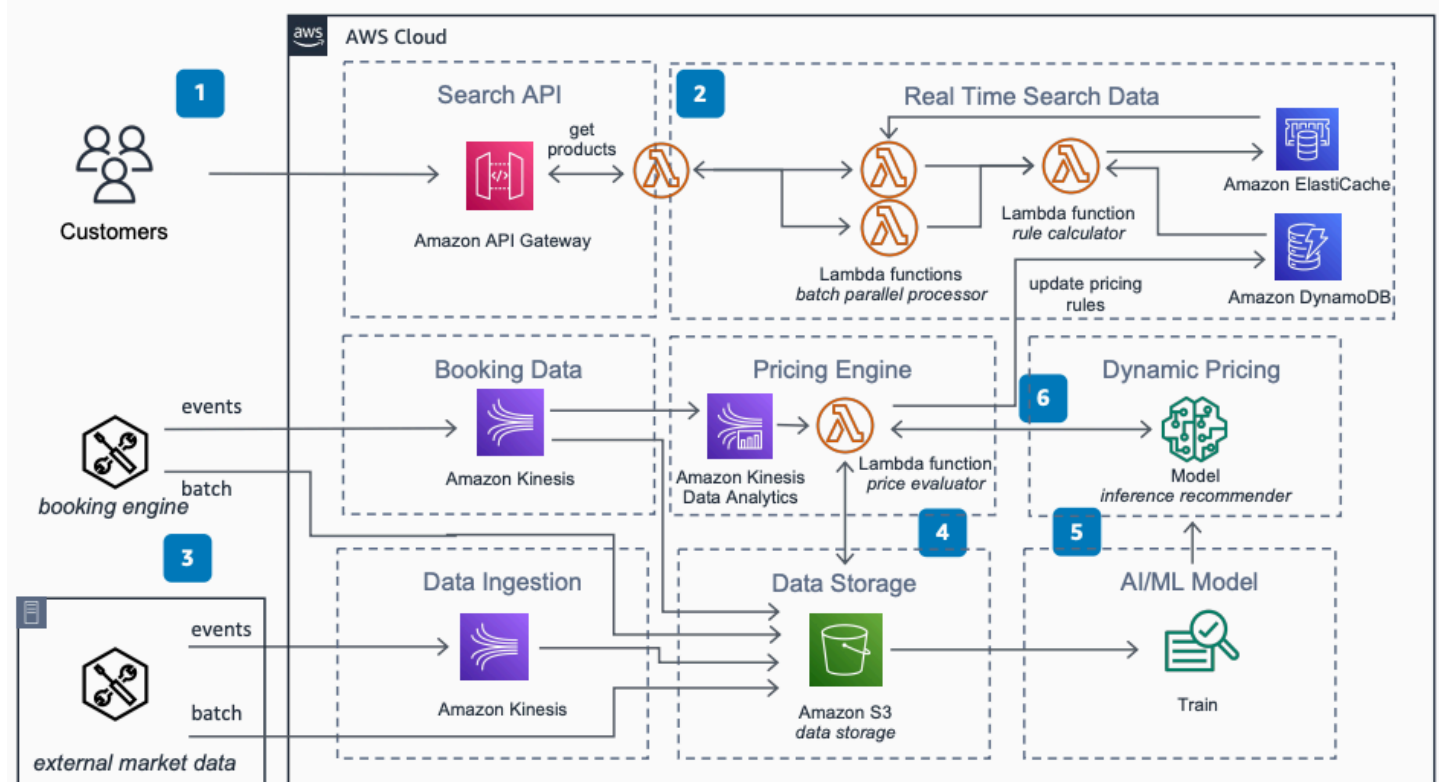
..... iv

Stratégie sans serveur pour une tarification dynamique

Date de publication : 28 juin 2021 ([Historique du diagramme](#))

Cette architecture présente une stratégie sans serveur pour une tarification dynamique. Pour les réservations basées sur des plages de dates, vous pouvez créer des prix en parallèle et les calculer dynamiquement en fonction de facteurs tels que la date, la durée et le nombre de personnes. Vous affichez ensuite des produits comparables triés avec des temps de réponse inférieurs à la seconde, et vous mettez à jour les prix en fonction des conditions actuelles du marché.

Stratégie sans serveur pour une tarification dynamique



Les étapes suivantes décrivent l'architecture :

1. Vous fournissez des données de recherche pour obtenir une liste triée de produits filtrés par prix et par d'autres attributs. Les données sont envoyées via [Amazon API Gateway](#), qui fait appel à plusieurs [AWS Lambda](#) fonctions pour traiter les demandes de prix en parallèle afin de consulter les prix en temps réel.

2. Les fonctions Lambda du processeur parallèle par lots appellent la fonction de calcul de règles pour obtenir les dernières règles de produit auprès d'[Amazon DynamoDB](#). Le produit est calculé en fonction de l'identifiant de recherche et stocké [sur Amazon ElastiCache](#). Une fois les fonctions de traitement par lots terminées, tous les résultats de l'ID de recherche sont renvoyés depuis le cache.
3. Les données historiques du moteur de réservation et les données de marché externes fournissent des informations permettant au moteur de tarification de disposer de suffisamment de données pour mettre à jour les prix en temps réel.
4. Les événements du moteur de réservation sont capturés et surveillés sur [Amazon Kinesis](#). Amazon Kinesis Data Analytics interroge le flux de données et déclenche la fonction Lambda de l'évaluateur de prix si un changement est détecté. Les événements sont stockés dans [Amazon Simple Storage Service](#) et sont accompagnés de données historiques sur les lots.
5. Un modèle [Amazon SageMaker AI](#) Train utilise des données de réservation historiques et des données historiques augmentées pour entraîner le modèle à des recommandations de prix.
6. À l'aide d'un modèle de recommandation par inférence basé sur l' [SageMaker IA](#), le moteur de tarification obtient de nouvelles recommandations de prix en fonction des conditions du marché.

Suggestions de lecture

Pour plus d'informations, consultez les ressources suivantes :

- [AWS Icônes de l'architecture](#)
- [AWS Centre d'architecture](#)
- [AWS Well-Architected](#)

Historique du diagramme

Pour être informé des mises à jour de ce schéma d'architecture de référence, abonnez-vous au fil RSS.

Modification	Description	Date
Publication initiale	Schéma d'architecture de référence publié pour la première fois.	28 juin 2021

 Note

Pour vous abonner aux mises à jour RSS, un plug-in RSS doit être activé pour le navigateur que vous utilisez.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.