



Diagram Arsitektur

Strategi Tanpa Server untuk Harga Dinamis



Strategi Tanpa Server untuk Harga Dinamis: Diagram Arsitektur

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Harga Dinamis i

 Strategi Tanpa Server untuk Harga Dinamis 1

 Sumber bacaan lebih lanjut 2

 Sejarah diagram 2

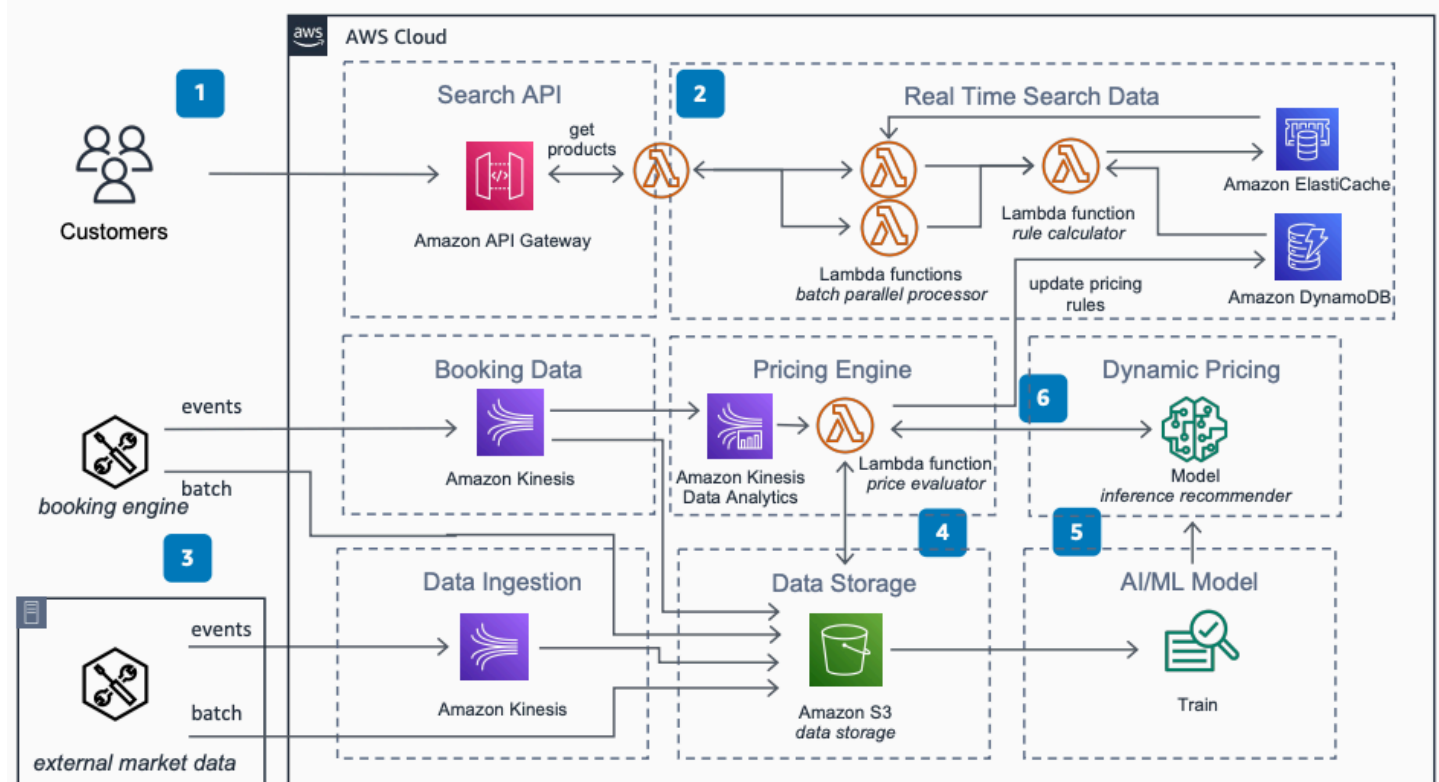
..... iv

Strategi Tanpa Server untuk Harga Dinamis

Tanggal publikasi: Juni 28, 2021 ([Sejarah diagram](#))

Arsitektur ini menunjukkan strategi tanpa server untuk penetapan harga dinamis. Untuk pemesanan berdasarkan rentang tanggal, Anda dapat membuat harga secara paralel dan menghitungnya secara dinamis berdasarkan faktor-faktor seperti tanggal, durasi, dan jumlah orang. Anda kemudian menampilkan produk sebanding yang diurutkan dengan waktu respons di bawah detik, dan memperbarui harga berdasarkan kondisi pasar saat ini.

Strategi Tanpa Server untuk Harga Dinamis



Langkah-langkah berikut menjelaskan arsitektur:

1. Anda memberikan masukan pencarian untuk mendapatkan daftar produk yang diurutkan berdasarkan harga dan atribut lainnya. Data dikirim melalui [Amazon API Gateway](#) yang memanggil beberapa [AWS Lambda](#) fungsi untuk memproses permintaan harga secara paralel untuk pencarian harga real-time.

2. Fungsi Lambda prosesor paralel batch memanggil fungsi kalkulator aturan untuk mendapatkan aturan produk terbaru dari [Amazon DynamoDB](#). Produk dihitung terhadap ID pencarian dan disimpan di [Amazon ElastiCache](#). Setelah fungsi batch selesai, semua hasil untuk ID pencarian dikembalikan dari cache.
3. Data historis dari mesin pemesanan dan data pasar eksternal memberikan informasi sehingga mesin penetapan harga memiliki data yang cukup untuk memperbarui harga secara real time.
4. Peristiwa mesin pemesanan ditangkap dan dipantau di [Amazon Kinesis](#). Amazon Kinesis Data Analytics menanyakan aliran data dan memicu fungsi Lambda evaluator harga jika perubahan terdeteksi. Acara disimpan di [Amazon Simple Storage Service](#) dengan mendukung data batch historis.
5. Model [Amazon SageMaker AI](#) Train menggunakan data pemesanan historis dan data historis tambahan untuk melatih model untuk rekomendasi harga.
6. Menggunakan model rekomendasi inferensi SageMaker AI, mesin penetapan harga mendapatkan rekomendasi harga baru berdasarkan kondisi pasar.

Sumber bacaan lebih lanjut

Untuk informasi tambahan, lihat sumber daya berikut:

- [AWS Ikon Arsitektur](#)
- [AWS Pusat Arsitektur](#)
- [AWS Well-Architected](#)

Sejarah diagram

Untuk diberitahu tentang pembaruan diagram arsitektur referensi ini, berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Publikasi awal	Diagram arsitektur referensi pertama kali diterbitkan.	28 Juni 2021

 Note

Untuk berlangganan pembaruan RSS, Anda harus mengaktifkan plugin RSS untuk browser yang Anda gunakan.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.