Architecture Diagrams

Serverless Strategy for Dynamic Pricing

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

Serverless Strategy for Dynamic Pricing: Architecture Diagrams

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

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

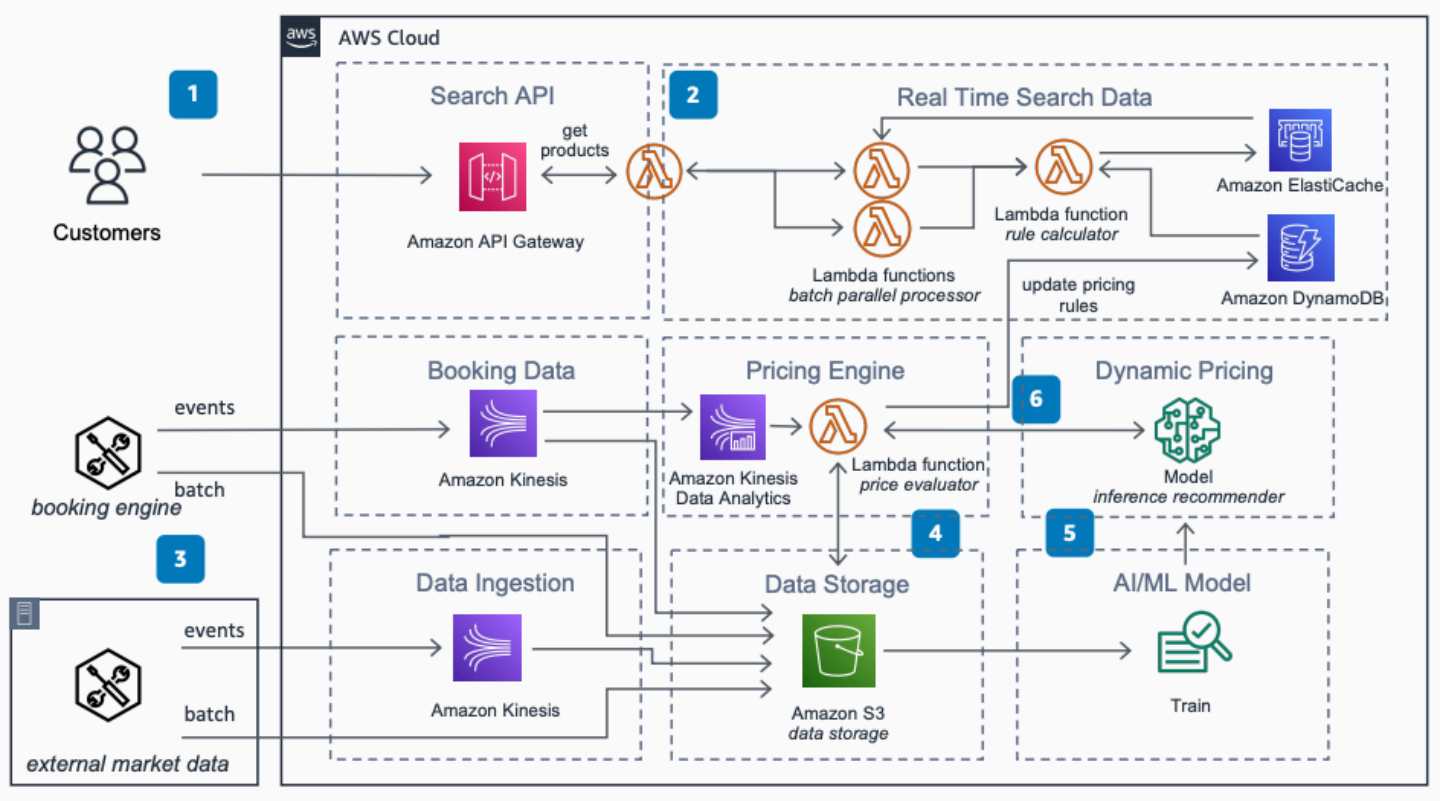
Table of Contents

Serverless Strategy for Dynamic Pricing

Publication date: **June 28, 2021 (Diagram history)**

This architecture shows a serverless strategy for dynamic pricing. For bookings based on date ranges, you can build prices in parallel and dynamically calculate them based on factors such as date, duration, and number of people. You then display sorted comparable products with sub-second response times, and update prices based on current market conditions.

Serverless Strategy for Dynamic Pricing



The following steps describe the architecture:

1. You provide search input to obtain a sorted list of products filtered by price and other attributes. Data is sent through Amazon API Gateway which calls multiple AWS Lambda functions to process pricing requests in parallel for real-time price lookups.
2. The batch parallel processor Lambda functions call the rule calculator function to get the latest product rules from Amazon DynamoDB. The product is calculated against the search ID and

stored in Amazon ElastiCache. After the batch functions complete, all results for the search ID are returned from the cache.

3. Historical data from the booking engine and external market data provide information so the pricing engine has sufficient data to update pricing in real time.
4. The booking engine events are captured and monitored on Amazon Kinesis. Amazon Kinesis Data Analytics queries the data stream and triggers the price evaluator Lambda function if a change is detected. Events are stored in Amazon Simple Storage Service with supporting historical batch data.
5. An Amazon SageMaker AI Train model uses historic booking data and augmented historic data to train the model for pricing recommendations.
6. Using an SageMaker AI inference recommender model, the pricing engine gets new price recommendations based on market conditions.

Further reading

For additional information, refer to the following resources:

- AWS Architecture Icons
- AWS Architecture Center
- AWS Well-Architected

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	June 28, 2021

> **Note**
>
> To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.