



Diagram Arsitektur

# Server-Side Rendering Micro-Frontends di AWS



# Server-Side Rendering Micro-Frontends di AWS: 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

SSR Micro-Frontends ..... i

Server-Side Rendering Micro-Frontends di AWS ..... 1

Sumber bacaan lebih lanjut ..... 2

Sejarah diagram ..... 2

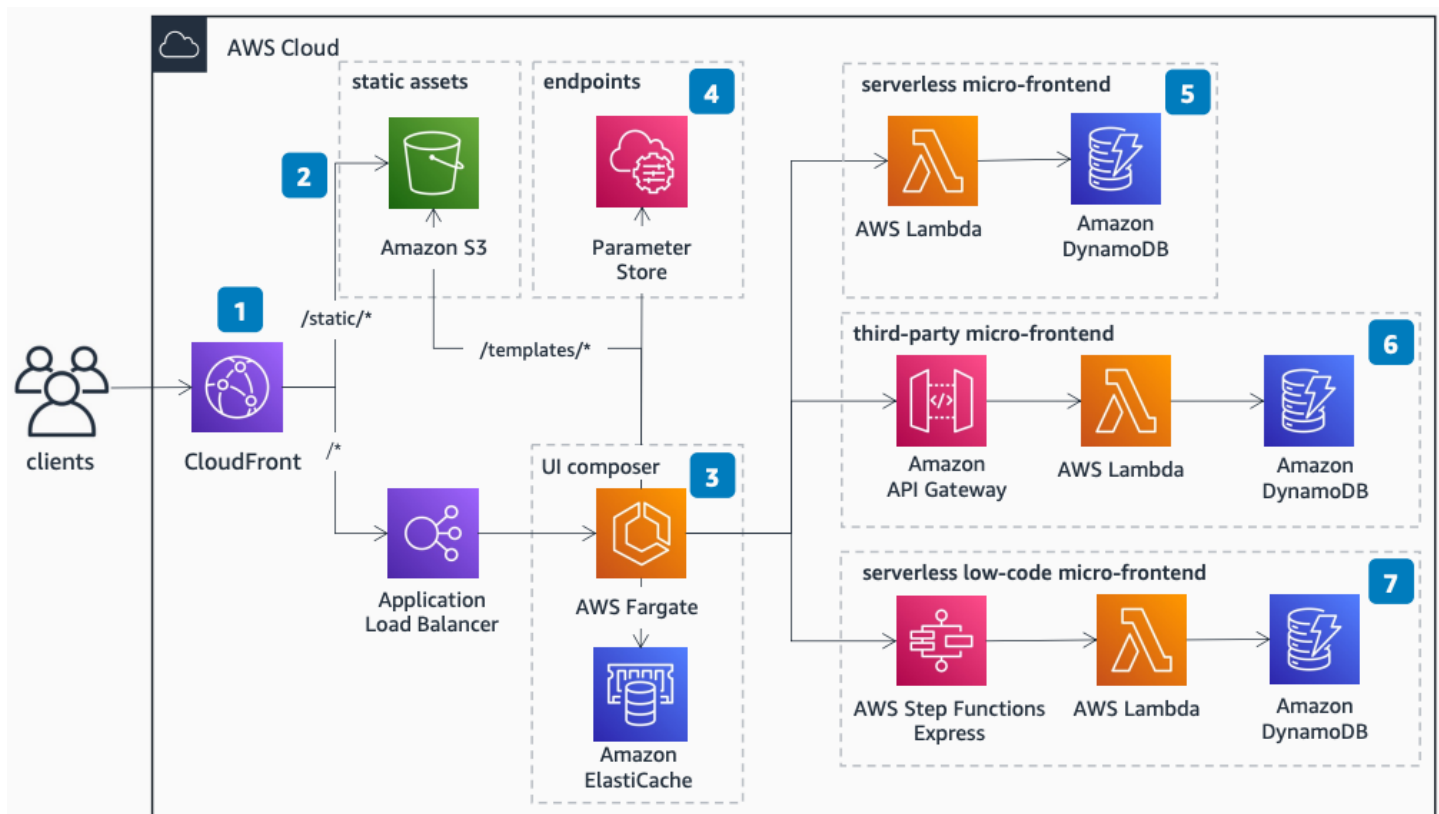
..... iv

# Server-Side Rendering Micro-Frontends di AWS

Tanggal publikasi: 9 September 2022 ([Sejarah diagram](#))

Arsitektur ini menunjukkan cara mengimplementasikan micro-frontend rendering sisi server dalam AWS menggunakan pendekatan tanpa server. Setiap micro-frontend tanpa server mengembalikan fragmen HTML (HTML-on-the-wire), dan komposer UI menyatukan bagian-bagian independen ini untuk menciptakan pengalaman yang mulus bagi pengguna.

## Server-Side Rendering Micro-Frontends di AWS



Langkah-langkah berikut menjelaskan arsitektur:

1. [Amazon CloudFront](#) berfungsi sebagai titik masuk dengan dua asal: [bucket](#) Amazon Simple Storage Service dan Penyeimbang Beban Aplikasi publik.
2. Bucket Amazon S3 berisi semua file statis untuk browser seperti dependensi micro-frontend umum, gambar, dan file CSS. Ini juga berisi template yang digunakan komposer UI untuk menempatkan setiap mikro-frontend pada halaman HTML.

3. UI komposer berjalan di [cluster](#) AWS Fargate dan menyatukan frontend mikro yang berbeda, streaming respons untuk meningkatkan kinerja. Gunakan [Amazon ElastiCache](#) untuk menyimpan output micro-frontend atau seluruh halaman untuk peningkatan kinerja tambahan.
4. Gunakan [AWS Systems Manager](#) Parameter Store untuk mengumpulkan semua titik akhir layanan mikro. Mereka dapat berupa titik akhir HTTP atau Nama Sumber Daya Amazon (ARN), memisahkan dependensi tingkat tim.
5. Micro-frontend tanpa server menggunakan [AWS Lambda](#) dan [Amazon DynamoDB](#) untuk menyimpan data dan membuat fragmen HTML siap untuk disematkan dalam template komposer UI.
6. Untuk mikrofrontend pihak ketiga, gunakan [Amazon API Gateway](#) untuk memvalidasi token atau kunci API, memastikan hanya aplikasi Anda yang dapat mengakses titik akhir tersebut.
7. Gunakan [AWS Step Functions](#) Express sebagai solusi kode rendah untuk menghasilkan frontend mikro. Step Functions terintegrasi dengan lebih dari 200 layanan untuk mengurangi komputasi, mengambil data dari DynamoDB secara native, dan mendelegasikan rendering ke fungsi Lambda.

## 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          |
|--------------------------------|--------------------------------------------------------|------------------|
| <a href="#">Publikasi awal</a> | Diagram arsitektur referensi pertama kali diterbitkan. | 9 September 2022 |

 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.