



Diagrammi di architettura

Server-Side Rendering Micro-Frontends in AWS



Server-Side Rendering Micro-Frontends in AWS: Diagrammi di architettura

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

SSR Micro-Frontends i

Server-Side Micro-Frontends Rendering in AWS 1

Approfondimenti 2

Storia del diagramma 2

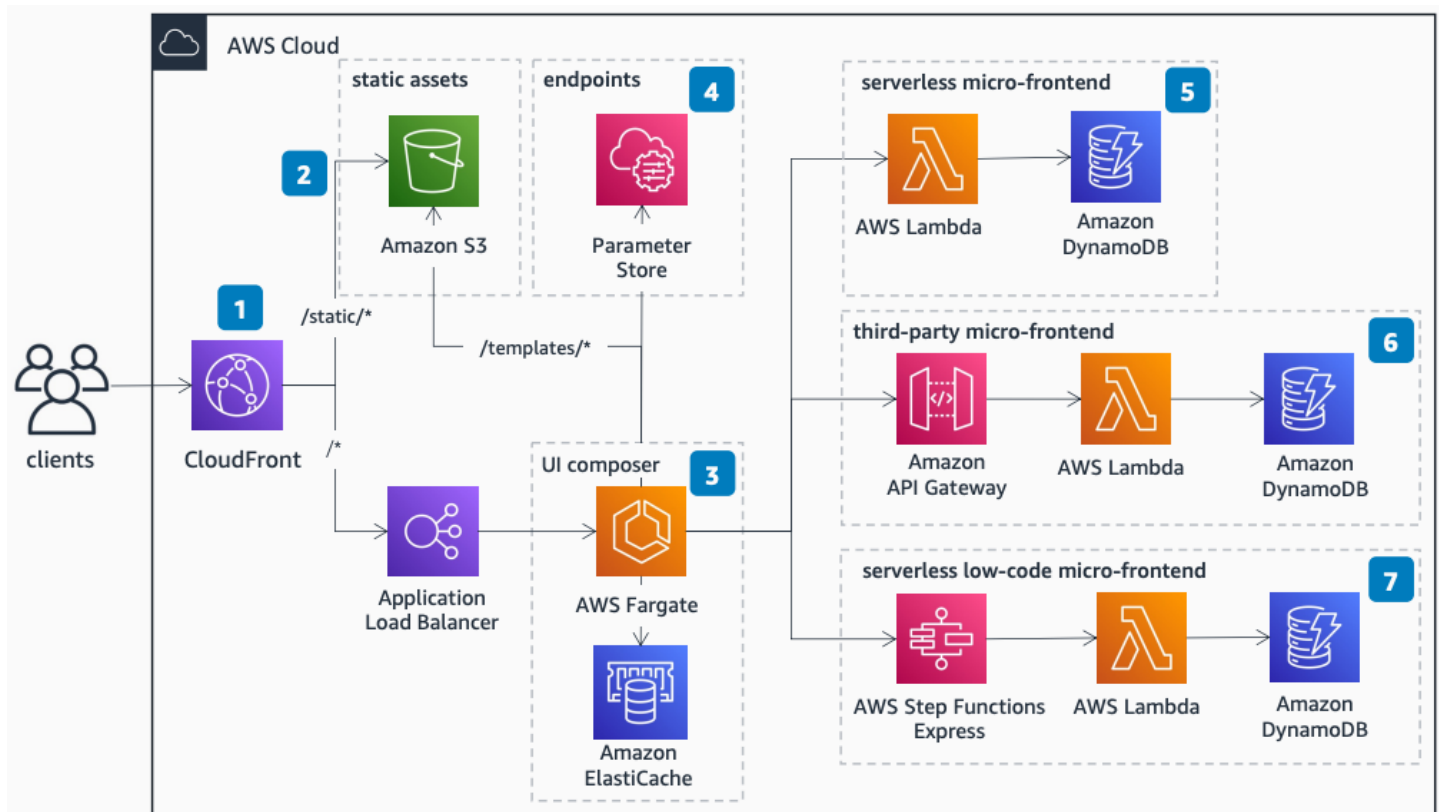
..... iv

Server-Side Rendering Micro-Frontends in AWS

Data di pubblicazione: [Storia del diagramma](#) 9 settembre 2022 ()

Questa architettura mostra come implementare i micro-frontend di rendering lato server AWS utilizzando un approccio serverless. Ogni microfrontend serverless restituisce un frammento HTML (HTML-on-the-wire) e un compositore dell'interfaccia utente unisce queste parti indipendenti per creare un'esperienza perfetta per gli utenti.

Server-Side Micro-Frontends Rendering in AWS



I passaggi seguenti descrivono l'architettura:

1. [Amazon CloudFront](#) funge da punto di ingresso con due origini: un [bucket](#) Amazon Simple Storage Service e un Application Load Balancer pubblico.
2. Il bucket Amazon S3 contiene tutti i file statici per il browser, come le comuni dipendenze di micro-frontend, immagini e file CSS. Contiene anche i modelli utilizzati dal compositore dell'interfaccia utente per posizionare ogni micro-frontend in una pagina HTML.

3. Il compositore dell'interfaccia utente viene eseguito su un [cluster](#) AWS Fargate e unisce diversi micro-frontend, trasmettendo in streaming la risposta per migliorare le prestazioni. Usa [Amazon ElastiCache](#) per memorizzare nella cache l'output del micro-frontend o intere pagine per aumentare ulteriormente le prestazioni.
4. Usa [AWS Systems Manager](#) Parameter Store per raccogliere tutti gli endpoint dei microservizi. Possono essere endpoint HTTP o Amazon Resource Names (ARN), che separano le dipendenze a livello di team.
5. Un micro-frontend serverless utilizza [Amazon DynamoDB per archiviare i dati AWS Lambda e renderizzare un frammento HTML pronto](#) per l'incorporamento nel modello di composizione dell'interfaccia utente.
6. Per i microfrontend di terze parti, utilizza [Amazon API Gateway per](#) convalidare token o chiavi API, assicurandoti che solo la tua applicazione possa accedere a quell'endpoint.
7. Usa [AWS Step Functions](#) Express come soluzione low-code per generare micro-frontend. Step Functions si integra con oltre 200 servizi per ridurre il calcolo, recupera i dati da DynamoDB in modo nativo e delega il rendering a una funzione Lambda.

Approfondimenti

Per ulteriori informazioni, consulta le seguenti risorse:

- [AWS Icone dell'architettura](#)
- [AWS Centro di architettura](#)
- [AWS Well-Architected](#)

Storia del diagramma

Per ricevere notifiche sugli aggiornamenti di questo diagramma di architettura di riferimento, iscriviti al feed RSS.

Modifica	Descrizione	Data
Pubblicazione iniziale	Diagramma dell'architettura di riferimento pubblicato per la prima volta.	9 settembre 2022

 Note

Per iscriverti agli aggiornamenti RSS, devi avere un plugin RSS abilitato per il browser che stai utilizzando.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.