



Diagramas de arquitectura

Server-Side Renderización Micro-Frontends en AWS



Server-Side Renderización Micro-Frontends en AWS: Diagramas de arquitectura

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

SSR Micro-Frontends i

Server-Side Micro-Frontends Renderización en AWS 1

Documentación adicional 2

Historial del diagrama 2

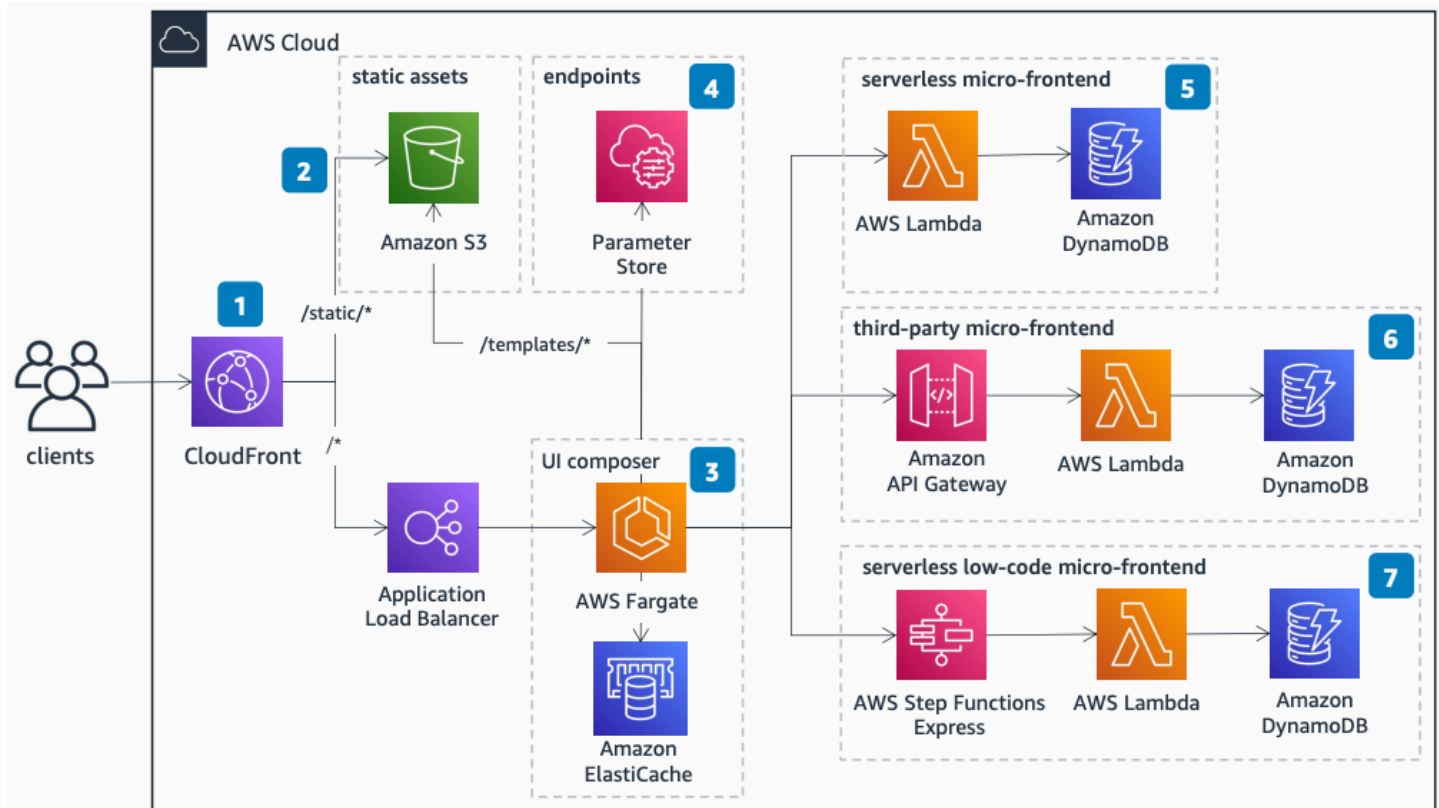
..... iv

Server-Side Renderización Micro-Frontends en AWS

Fecha de publicación: 9 de septiembre de 2022 () [Historial del diagrama](#)

Esta arquitectura muestra cómo implementar microinterfaces de renderizado en el lado del servidor AWS mediante un enfoque sin servidor. Cada microfrontend sin servidor devuelve un fragmento HTML (HTML-on-the-wire) y un compositor de interfaz de usuario une estas partes independientes para crear una experiencia perfecta para los usuarios.

Server-Side Micro-Frontends Renderización en AWS



Los siguientes pasos describen la arquitectura:

1. [Amazon CloudFront](#) sirve como punto de entrada con dos orígenes: un [depósito de](#) Amazon Simple Storage Service y un balanceador de carga de aplicaciones público.
2. El bucket de Amazon S3 contiene todos los archivos estáticos del navegador, como las dependencias comunes de microfrontend, las imágenes y los archivos CSS. También contiene las plantillas que usa el compositor de la interfaz de usuario para colocar cada microfrontend en una página HTML.

3. El compositor de la interfaz de usuario se ejecuta en un [clúster de](#) AWS Fargate y combina diferentes microinterfaces para transmitir la respuesta para mejorar el rendimiento. Utilice [Amazon ElastiCache](#) para almacenar en caché la salida de la microinterfaz o páginas enteras para aumentar aún más el rendimiento.
4. Utilice el almacén de [parámetros de](#) AWS Systems Manager para recopilar todos los puntos de enlace de los microservicios. Pueden ser puntos de enlace HTTP o nombres de recursos de Amazon (ARN), lo que desvincula las dependencias a nivel de equipo.
5. Una microinterfaz sin servidor utiliza [Amazon DynamoDB para almacenar datos AWS Lambda y procesar un fragmento HTML listo](#) para su incrustación en la plantilla del compositor de la interfaz de usuario.
6. En el caso de las microinterfaces de terceros, utilice [Amazon API Gateway](#) para validar los tokens o las claves de API y asegurarse de que solo su aplicación pueda acceder a ese punto final.
7. Utilice [AWS Step Functions](#) Express como una solución con poco código para generar microinterfaces. Step Functions se integra con más de 200 servicios para reducir la computación, recupera datos de DynamoDB de forma nativa y delega el procesamiento en una función de Lambda.

Documentación adicional

Para obtener información adicional, consulte los siguientes recursos:

- [AWS Iconos de arquitectura](#)
- [AWS Centro de arquitectura](#)
- [AWS Well-Architected](#)

Historial del diagrama

Para recibir notificaciones sobre las actualizaciones de este diagrama de arquitectura de referencia, suscríbase a la fuente RSS.

Cambio

Descripción

Fecha

[Publicación inicial](#)

El diagrama de arquitectura de referencia se publicó por primera vez.

9 de septiembre de 2022

 Note

Para suscribirse a las actualizaciones de RSS, debe tener un complemento de RSS habilitado para el navegador que esté utilizando.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.