



Schémas d'architecture

Diffusion de Real-Time données d'applications Web sans serveur



Diffusion de Real-Time données d'applications Web sans serveur:

Schémas d'architecture

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Real-Time Radiodiffusion i

Diffusion de Real-Time données d'applications Web sans serveur 1

Suggestions de lecture 2

Historique du diagramme 2

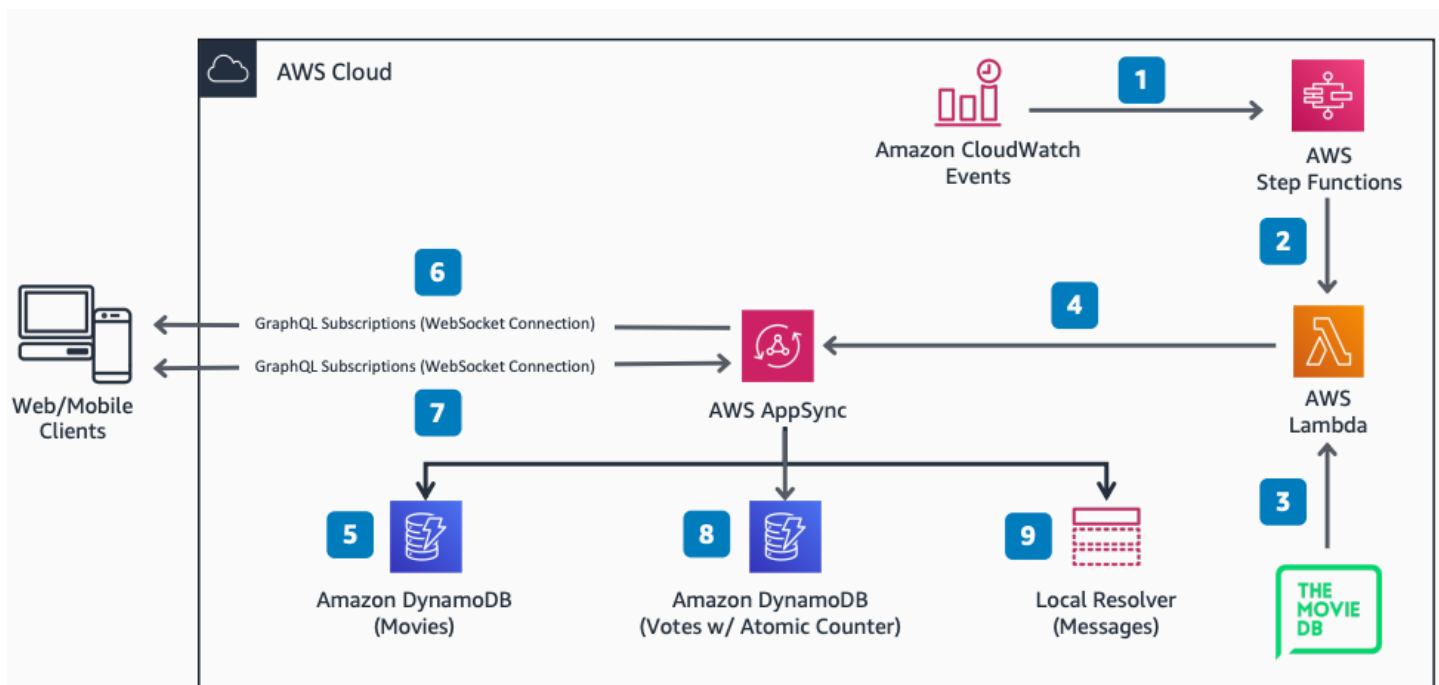
..... iv

Diffusion de Real-Time données d'applications Web sans serveur

Date de publication : 24 mars 2021 ([Historique du diagramme](#))

Cette architecture montre comment créer une application de vote pour les films à l'aide d'[AWS AppSync](#), [AWS Lambda](#), [AWS Step Functions](#), et d'[Amazon DynamoDB](#). Vous pouvez utiliser la diffusion en temps réel vers le backend et le client avec des abonnements GraphQL gérés via WebSockets

Diffusion de Real-Time données d'applications Web sans serveur



Les étapes suivantes décrivent l'architecture :

1. [Amazon CloudWatch](#) Events lance un flux de travail dans Step Functions toutes les 60 secondes.
2. Step Functions déclenche Lambda toutes les 10 secondes.
3. Lambda appelle The Movie DB l'API pour récupérer les métadonnées d'un seul film aléatoire à partir de la liste des films les plus populaires.
4. Lambda met à jour le tableau des films, met à zéro les votes actuels et augmente le classement dans le tableau des votes grâce à des mutations GraphQL vers AppSync

5. AppSync met à jour la table des vidéos dans DynamoDB avec la seule vidéo en cours extraite de Lambda.
6. Tous les clients connectés abonnés à la mutation du backend voient la même affiche du film et le synopsis actuels à l'écran pendant la diffusion.
7. Les clients votent pour le film en cours pendant une fenêtre de 10 secondes et peuvent envoyer et recevoir des messages de discussion dans un salon de discussion public.
8. Lambda met à jour le classement et les votes des clients sur les films grâce à AppSync des mutations.
9. Le salon de discussion public affiche les messages actuels d'une pub/sub chaîne via Local Resolver. Les messages ne sont pas conservés sur le stockage principal et seuls les nouveaux messages sont affichés.

Suggestions de lecture

Pour plus d'informations, consultez les ressources suivantes :

- [AWS Icônes de l'architecture](#)
- [AWS Centre d'architecture](#)
- [AWS Well-Architected](#)
- [Architecture AppSync Real-Time de référence AWS sur GitHub](#)

Historique du diagramme

Pour être informé des mises à jour de ce schéma d'architecture de référence, abonnez-vous au fil RSS.

Modification	Description	Date
Publication initiale	Schéma d'architecture de référence publié pour la première fois.	24 mars 2021

 Note

Pour vous abonner aux mises à jour RSS, un plug-in RSS doit être activé pour le navigateur que vous utilisez.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.