



Schémas d'architecture

Exposez votre lac de données en tant qu'API GraphQL



Exposez votre lac de données en tant qu'API GraphQL: 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

API GraphQL de Data Lake i

Exposez votre lac de données en tant qu'API GraphQL 1

Suggestions de lecture 2

Historique du diagramme 2

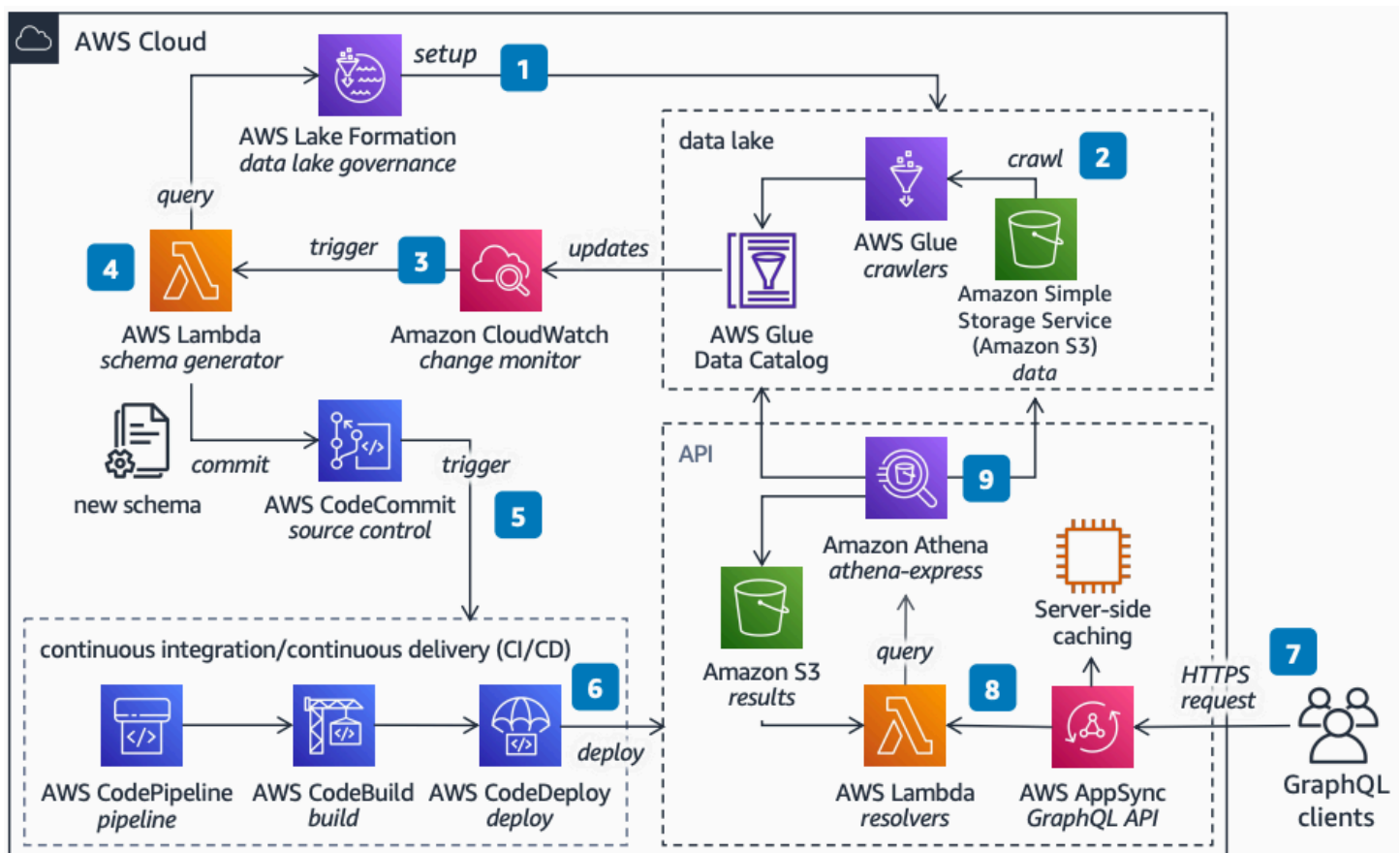
..... iv

Exposez votre lac de données en tant qu'API GraphQL

Date de publication : 25 avril 2022 ([Historique du diagramme](#))

Cette architecture montre comment créer un lac de données à l'aide de [Lake Formation](#) et le partager en tant qu'GraphQLAPI auto-documentée. Vous pouvez créer automatiquement l'API à l'aide d'un CI/CD pipeline réactif et autoriser les clients externes autorisés à interroger votre lac de données en utilisant une GraphQL syntaxe via HTTPS.

Exposez votre lac de données en tant qu'API GraphQL



Les étapes suivantes décrivent l'architecture :

1. Configurez un lac de données à l'aide de Lake Formation et [Amazon Simple Storage Service](#).
2. Explorez les données des compartiments Amazon S3 avec [AWS Glue](#) pour créer un catalogue de données.

3. Surveillez les modifications apportées au catalogue de données AWS Glue avec [Amazon CloudWatch](#) et déclenchez une [AWS Lambda](#) fonction à chaque modification du catalogue afin de mettre à jour le schéma de l'API externe.
4. Dans la fonction Lambda, interrogez le catalogue de données et les paramètres de sécurité les plus récents, générez automatiquement le code source pour créer et déployer une nouvelle version du schéma et des résolveurs de l'API externe, puis validez le code dans AWS CodeCommit
5. Configurez CodeCommit pour démarrer automatiquement un nouveau CI/CD pipeline sur AWS CodePipeline après toute validation de code.
6. CodePipeline Utilisez-la pour créer une nouvelle version de l'API avec AWS à CodeBuild l'aide du code source nouvellement généré, et utilisez AWS CodeDeploy pour déployer à la fois la nouvelle AppSync [GraphQL API](#) Amazon et les nouveaux résolveurs de requêtes Lambda.
7. Permettez à vos GraphQL clients d'invoquer en toute sécurité l'API externe via HTTPS avec le mécanisme d'authentification de votre choix. Activez la mise en cache AppSync côté serveur.
8. Utilisez les résolveurs de requêtes AppSync Lambda pour traduire la GraphQL requête en requêtes SQL prises en charge par [Athena](#).
9. Exécutez les tâches Athena de manière synchrone avec la bibliothèque athena-express, en expirant de manière appropriée et en limitant le jeu de résultats.

Suggestions de lecture

Pour plus d'informations, consultez les ressources suivantes :

- [AWS Icônes de l'architecture](#)
- [AWS Centre d'architecture](#)
- [AWS Well-Architected](#)

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

Le schéma d'architecture de
référence a été publié pour la
première fois

25 avril 2022

 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.