



Diagramas de arquitectura

Exponga su lago de datos como una API de GraphQL



Exponga su lago de datos como una API de GraphQL: 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

API GraphQL de Data Lake i

Exponga su lago de datos como una API de GraphQL 1

Documentación adicional 2

Historial del diagrama 2

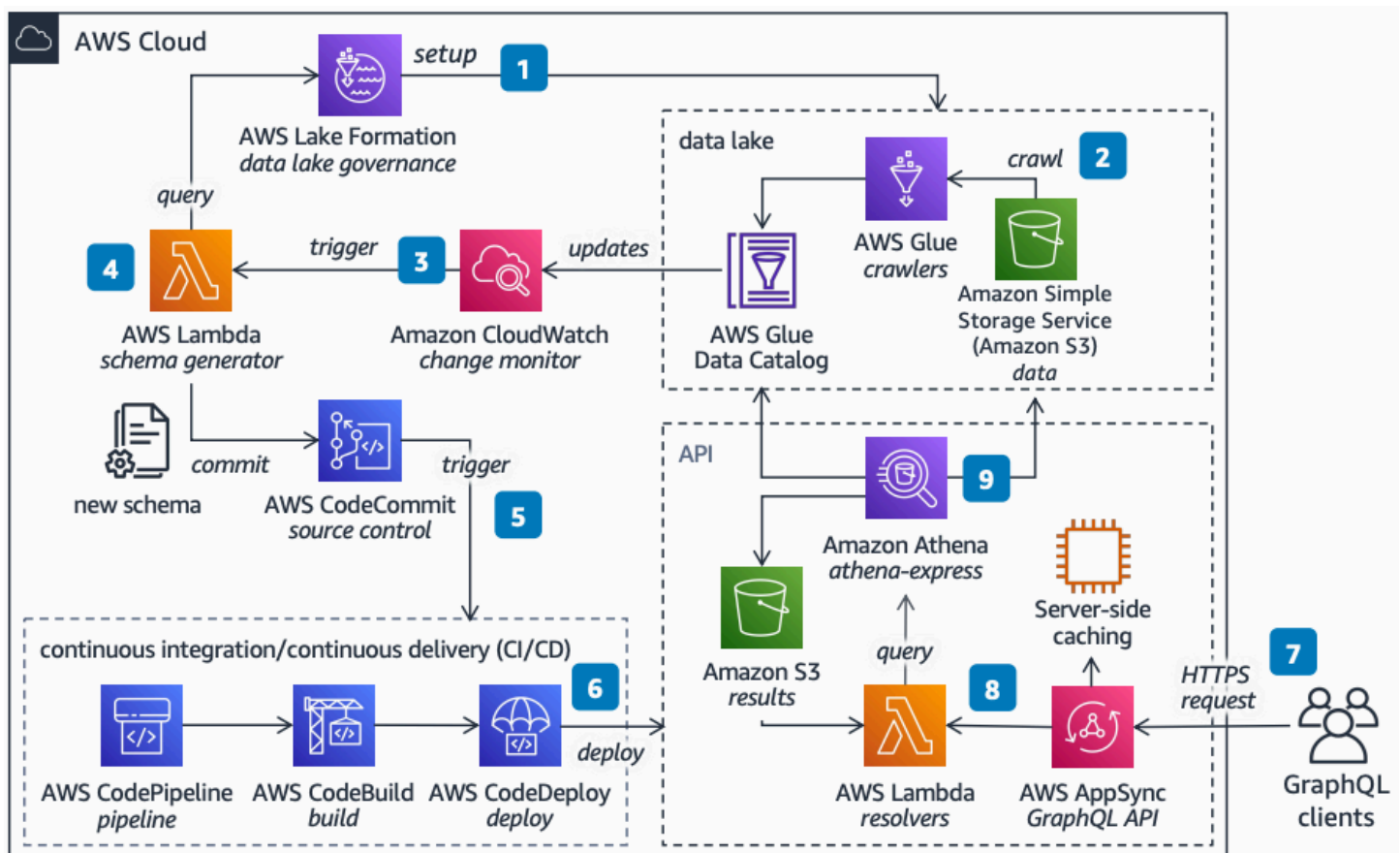
..... iv

Exponga su lago de datos como una API de GraphQL

Fecha de publicación: 25 de abril de 2022 () [Historial del diagrama](#)

Esta arquitectura muestra cómo crear un lago de datos con [Lake Formation](#) y compartirlo como una GraphQL API autodocumentada. Puedes crear automáticamente la API con una CI/CD canalización reactiva y permitir que los clientes externos autorizados consulten tu lago de datos mediante la GraphQL sintaxis a través de HTTPS.

Exponga su lago de datos como una API de GraphQL



Los siguientes pasos describen la arquitectura:

1. Configure un lago de datos con Lake Formation y [Amazon Simple Storage Service](#).
2. Rastree los datos de los depósitos de Amazon S3 con [AWS Glue](#) para crear un catálogo de datos.

3. Supervise cualquier cambio en el catálogo de datos de AWS Glue con [Amazon CloudWatch](#) y active una [AWS Lambda](#) función en cada cambio en el catálogo para actualizar el esquema de la API externa.
4. En la función Lambda, consulte el catálogo de datos y la configuración de seguridad más recientes, genere automáticamente el código fuente para crear e implementar una nueva versión del esquema y los solucionadores de la API externa y envíe el código a AWS. CodeCommit
5. Configure CodeCommit para iniciar automáticamente una nueva CI/CD canalización en AWS CodePipeline después de cualquier confirmación de código.
6. CodePipeline Utilícela para crear una nueva versión de la API con AWS CodeBuild utilizando el código fuente recién generado y utilice AWS CodeDeploy para implementar tanto la nueva AppSync [GraphQL API de](#) Amazon como los nuevos solucionadores de consultas de Lambda.
7. Permita que sus GraphQL clientes invoquen de forma segura la API externa a través de HTTPS con el mecanismo de autenticación que elija. Activa el almacenamiento en caché del AppSync lado del servidor.
8. Utilice los solucionadores de consultas de AppSync Lambda para convertir la GraphQL consulta en consultas de SQL compatibles con Athena. <https://docs.aws.amazon.com/athena/latest/ug/what-is.html>
9. Ejecute los trabajos de Athena de forma sincrónica con la biblioteca athena-express, agotando el tiempo de espera de forma adecuada y limitando el conjunto de resultados.

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.

25 de abril 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.