



Architektur-Diagramme

Stellen Sie Ihren Data Lake als GraphQL-API zur Verfügung



Stellen Sie Ihren Data Lake als GraphQL-API zur Verfügung: Architektur-Diagramme

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

Die Handelsmarken und die Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Data Lake GraphQL-API i

 Stellen Sie Ihren Data Lake als GraphQL-API zur Verfügung 1

 Weitere Informationen 2

 Verlauf des Diagramms 2

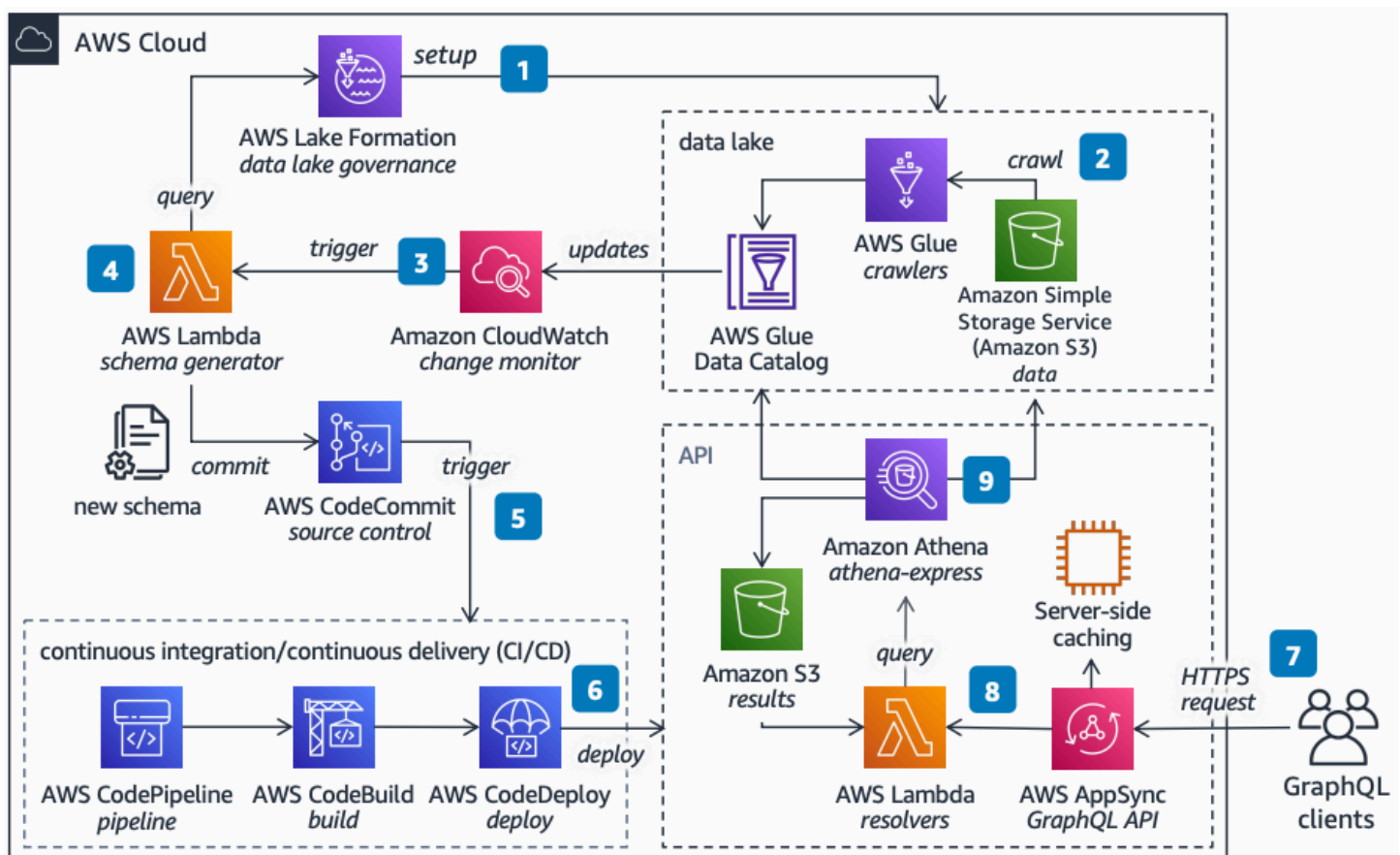
..... iv

Stellen Sie Ihren Data Lake als GraphQL-API zur Verfügung

Veröffentlichungsdatum: 25. April 2022 () [Geschichte des Diagramms](#)

Diese Architektur zeigt, wie ein Data Lake mithilfe von [Lake Formation erstellt](#) und als selbstdokumentierte GraphQL API geteilt wird. Sie können die API automatisch mit einer reaktiven CI/CD Pipeline erstellen und autorisierten externen Clients ermöglichen, Ihren Data Lake mithilfe der GraphQL Syntax über HTTPS abzufragen.

Stellen Sie Ihren Data Lake als GraphQL-API zur Verfügung



Die folgenden Schritte beschreiben die Architektur:

1. Richten Sie mithilfe von Lake Formation und [Amazon Simple Storage Service](#) einen Data Lake ein.
2. Durchforsten Sie die Daten in den Amazon S3-Buckets mit [AWS Glue](#), um einen Datenkatalog zu erstellen.

3. Überwachen Sie alle Änderungen im AWS Glue-Datenkatalog mit [Amazon und lösen Sie bei jeder Katalogänderung eine AWS Lambda Funktion aus CloudWatch](#), um das Schema der externen API zu aktualisieren.
4. Fragen Sie in der Lambda-Funktion den neuesten Datenkatalog und die Sicherheitseinstellungen ab, generieren Sie automatisch den Quellcode, um eine neue Version des Schemas und der Resolver der externen API zu erstellen und bereitzustellen, und übertragen Sie den Code an AWS. CodeCommit
5. Richten Sie CodeCommit es so ein, dass CodePipeline nach jedem Code-Commit automatisch eine neue CI/CD Pipeline auf AWS gestartet wird.
6. Verwenden Sie diese Option CodePipeline , um CodeBuild mithilfe des neu generierten Quellcodes eine neue Version der API mit AWS zu erstellen, und verwenden Sie AWS, CodeDeploy um sowohl die neue [AppSync](#) GraphQL Amazon-API als auch die neuen Lambda-Abfrage-Resolver bereitzustellen.
7. Ermöglichen Sie Ihren GraphQL Kunden, die externe API sicher über HTTPS mit dem von Ihnen gewählten Authentifizierungsmechanismus aufzurufen. Aktivieren Sie das AppSync serverseitige Caching.
8. Verwenden Sie die AppSync Lambda-Abfrage-Resolver, um die GraphQL Abfrage in SQL-Abfragen zu übersetzen, die von Athena unterstützt werden. <https://docs.aws.amazon.com/athena/latest/ug/what-is.html>
9. Führen Sie Athena-Jobs synchron mit der Athena-Express-Bibliothek aus, wobei das Timeout angemessen ist und die Ergebnismenge begrenzt wird.

Weitere Informationen

Weitere Informationen finden Sie in den folgenden Ressourcen:

- [AWS Architektur-Ikonen](#)
- [AWS Zentrum für Architektur](#)
- [AWS Well-Architected](#)

Geschichte des Diagramms

Abonnieren Sie den RSS-Feed, um über Aktualisierungen dieses Referenzarchitekturdiagramms informiert zu werden.

Änderung	Beschreibung	Datum
Erste Veröffentlichung	Das Referenzarchitekturdiagramm wurde zuerst veröffentlicht.	25. April 2022

 Note

Um RSS-Updates zu abonnieren, muss ein RSS-Plugin für den von Ihnen verwendeten Browser aktiviert sein.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.