



## Architektur-Diagramme

# Verwaltung von Inserts und Upserts in einem serverlosen Data Lake



# Verwaltung von Inserts und Upserts in einem serverlosen Data Lake:

## 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

Inserts und Upserts in Data Lake ..... i

Verwaltung von Inserts und Upserts in einem serverlosen Data Lake ..... 1

Weitere Informationen ..... 2

Verlauf des Diagramms ..... 2

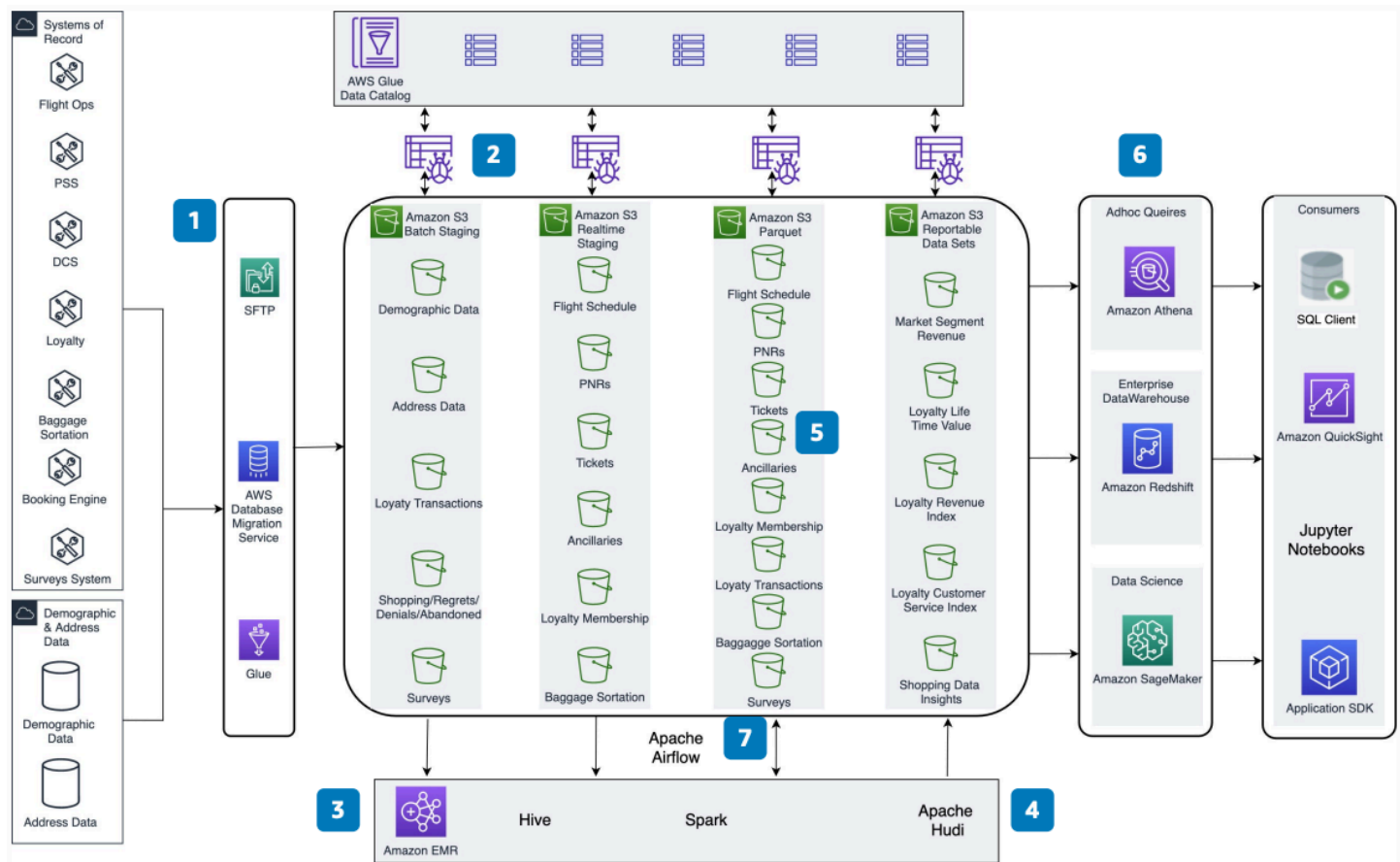
..... iv

# Verwaltung von Inserts und Upserts in einem serverlosen Data Lake

Veröffentlichungsdatum: 15. Juni 2021 () [Geschichte des Diagramms](#)

Diese Architektur zeigt, wie Sie Apache Hudi Running on Amazon EMR verwenden können, um Einfügungen und Aktualisierungen von Datensätzen in [Amazon Simple Storage Service zu verarbeiten](#). Sie können einen kostengünstigen und skalierbaren Data Lake aufbauen, der On-Demand-Analysen bereitstellt und persistente Data Marts erstellt.

## Verwaltung von Inserts und Upserts in einem serverlosen Data Lake



Die folgenden Schritte beschreiben die Architektur:

1. Erfassen Sie Daten aus Quellsystemen per Batch, Change Data Capture (CDC) oder per Streaming in die Rohschicht in Amazon S3.

2. Nachdem die Daten im Rohdatensee auf Amazon S3 gespeichert sind, crawlen Sie die Daten und füllen Sie sie mithilfe eines Crawlers in den [AWS](#) Glue-Datenkatalog ein.
3. Ziehen Sie Rohdaten in einen Amazon EMR-Cluster und lesen Sie sie mit Hive und Spark zur Bereinigung und Transformation ein.
4. Apache HudiDie Ausführung auf Amazon EMR liest die Daten mithilfe von Spark-APIs und führt Einfügungen und Upserts für die erforderlichen Datensätze durch.
5. Speichern Sie die bereinigten und transformierten Daten wieder in den von Amazon S3 verarbeiteten und berichtspflichtigen Buckets.
6. Verwenden Sie die meldepflichtigen Daten bei Bedarf mit [Athena](#) oder laden Sie sie in Amazon Redshift. <https://docs.aws.amazon.com/redshift/latest/dg/welcome.html> Verschiedene Benutzer, Tools und Ressourcen können diese Daten nutzen.
7. Übernehmen Sie die komplette Datenverlagerung, erstellen Sie Amazon EMR-Cluster auf Abruf für Batch-Daten und laden Sie die Daten mithilfe der Workflow-Orchestrierung mit Amazon Managed Workflows for Apache Airflow (MWAA).

## 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         |
|----------------------------------------|--------------------------------------------------------------|---------------|
| <a href="#">Erste Veröffentlichung</a> | Das Referenzarchitekturdiagramm wurde zuerst veröffentlicht. | 15. Juni 2021 |

 **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.