



Schémas d'architecture

Gestion des insertions et des modifications dans un lac de données sans serveur



Gestion des insertions et des modifications dans un lac de données 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

Insertions et modifications de Data Lake i

Gestion des insertions et des modifications dans un lac de données sans serveur 1

Suggestions de lecture 2

Historique du diagramme 2

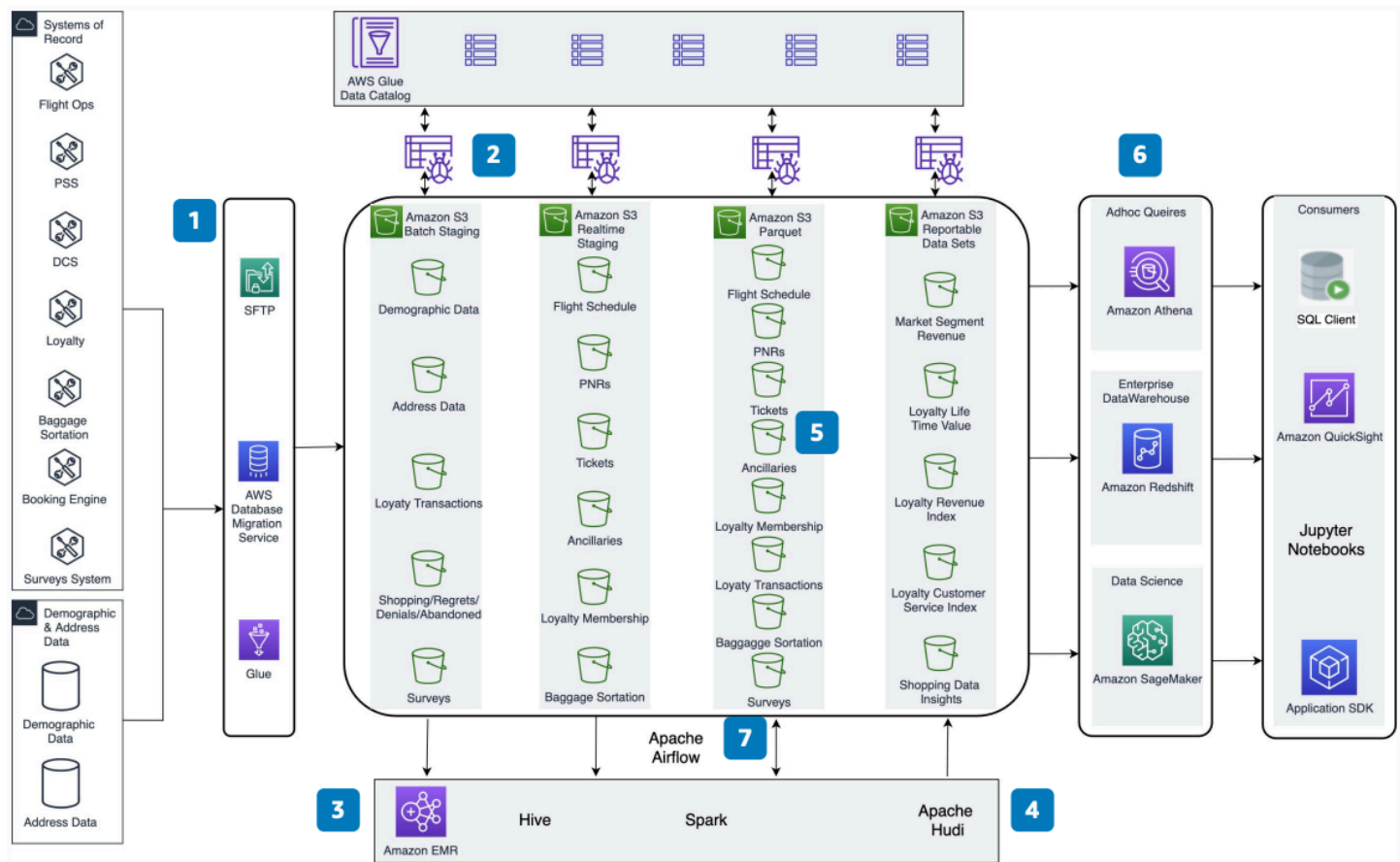
..... iv

Gestion des insertions et des modifications dans un lac de données sans serveur

Date de publication : 15 juin 2021 ([Historique du diagramme](#))

Cette architecture montre comment utiliser l'Apache Hudi exécuté sur Amazon EMR pour traiter les insertions et les mises à jour de jeux de données dans [Amazon Simple Storage Service](#). Vous pouvez créer un lac de données rentable et évolutif qui fournit des analyses à la demande et crée des data marts persistants.

Gestion des insertions et des modifications dans un lac de données sans serveur



Les étapes suivantes décrivent l'architecture :

1. Ingérez des données à partir de systèmes sources par lots, par capture de données modifiées (CDC) ou par diffusion dans la couche brute d'Amazon S3.

2. Une fois que les données persistent dans le lac de données brutes d'Amazon S3, analysez-les et ajoutez-les au catalogue de données [AWS Glue](#) à l'aide d'un robot d'exploration.
3. Extrayez les données brutes dans un cluster Amazon EMR et lisez-les à l'aide de Hive et Spark pour les nettoyer et les transformer.
4. Apache Hudi exécuté sur Amazon EMR lit les données à l'aide des API Spark et effectue des insertions et des modifications sur les ensembles de données requis.
5. Conservez les données nettoyées et transformées dans les compartiments traités et déclarables Amazon S3.
6. Utilisez les données déclarables à la demande à l'aide d'[Athena](#) ou chargez-les dans [Amazon Redshift](#). Différents utilisateurs, outils et ressources peuvent consommer ces données.
7. Gérez le mouvement complet des données, créez des clusters Amazon EMR à la demande pour les données par lots et chargez les données à l'aide de l'orchestration des flux de travail avec Amazon Managed Workflows for Apache Airflow (MWAA).

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	Schéma d'architecture de référence publié pour la première fois.	15 juin 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.