



Diagrammi di architettura

Database DevOps su AWS



Database DevOps su AWS: Diagrammi di architettura

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Flusso di lavoro di sviluppo 1

Database su AWS: flusso di lavoro di sviluppo DevOps 1

Approfondimenti 2

Storia del diagramma 2

Workflow di produzione 4

Database su AWS: flusso di lavoro di produzione DevOps 4

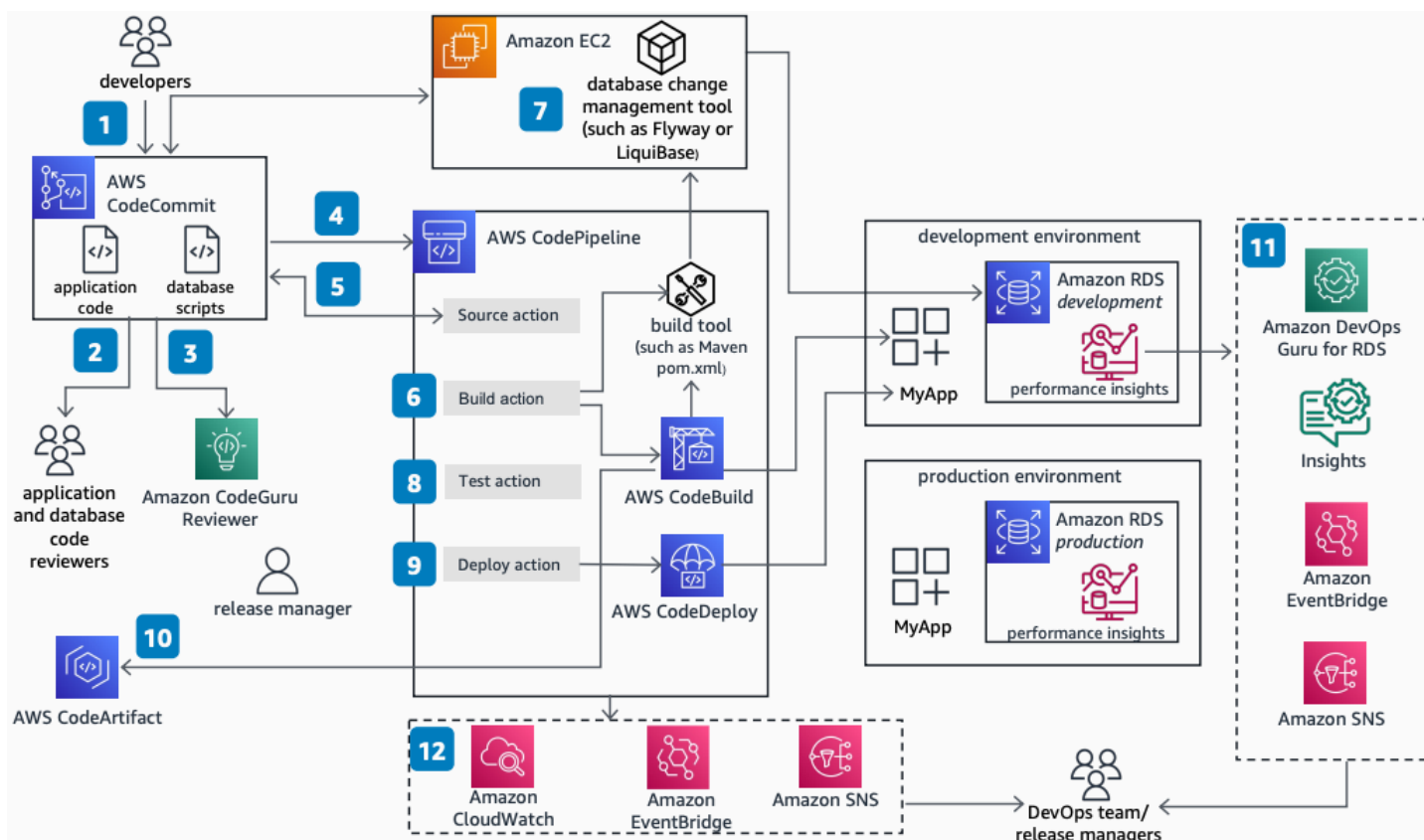
..... vi

Database DevOps su AWS: flusso di lavoro di sviluppo

Data di pubblicazione: 31 marzo 2023 () [Storia del diagramma](#)

Usa questa architettura nel flusso di lavoro di sviluppo per automatizzare le modifiche al database come parte della DevOps pratica. Crea un'orchestrazione per implementare le modifiche al database alla stessa velocità del codice dell'applicazione, applicare revisioni congiunte del codice del database e dell'applicazione, integrare le operazioni del database e del codice dell'applicazione, rilevare comportamenti anomali del sistema e automatizzare le notifiche ai team designati.

Database su AWS: flusso di lavoro di sviluppo DevOps



I passaggi seguenti descrivono l'architettura:

1. Lo sviluppatore analizza sia il codice dell'applicazione che quello del database [AWS CodeCommit](#).
2. Ad ogni richiesta push, AWS CodeCommit applica le revisioni del codice tramite l'uso di regole di approvazione.

3. Amazon CodeGuruReviewer valuta automaticamente le modifiche al codice, rileva gli errori e offre consigli sul codice.
4. AWS CodeCommit inizia a intraprendere azioni sulla [CodePipeline](#) pipeline.
5. L'azione CodePipeline Source verifica il codice dell'applicazione e del database.
6. L'azione CodePipeline Crea richiama uno strumento di gestione della compilazione per avviare la fase di creazione del codice dell'applicazione e del database.
7. Lo strumento di compilazione chiama lo strumento di modifica del database. Scarica gli script del database e li esegue sul database di destinazione.
8. Lo strumento di compilazione richiama gli script di test del database e convalida l'output.
9. L'azione CodePipeline Deploy distribuisce il codice nell'ambiente di destinazione.
10. Gli artefatti di compilazione vengono pubblicati in [AWS CodeArtifact](#) per essere utilizzati da altri utenti.
11. (Facoltativo) Usa Performance Insights su [Amazon RDS](#) e applica automaticamente Amazon DevOps Guru le tecniche di machine learning per rilevare rallentamenti delle prestazioni e problemi operativi.
12. [EventBridge](#) acquisisce gli eventi e li invia a un argomento di [Amazon SNS](#) per le notifiche agli utenti.

Approfondimenti

Per ulteriori informazioni, consulta le seguenti risorse:

- [AWS Icone dell'architettura](#)
- [AWS Centro di architettura](#)
- [AWS Well-Architected](#)

Storia del diagramma

Per ricevere notifiche sugli aggiornamenti di questo diagramma di architettura di riferimento, iscriviti al feed RSS.

Modifica

Descrizione

Data

[Pubblicazione iniziale](#)

Diagramma dell'architettura di riferimento pubblicato per la prima volta. 31 marzo 2023

 Note

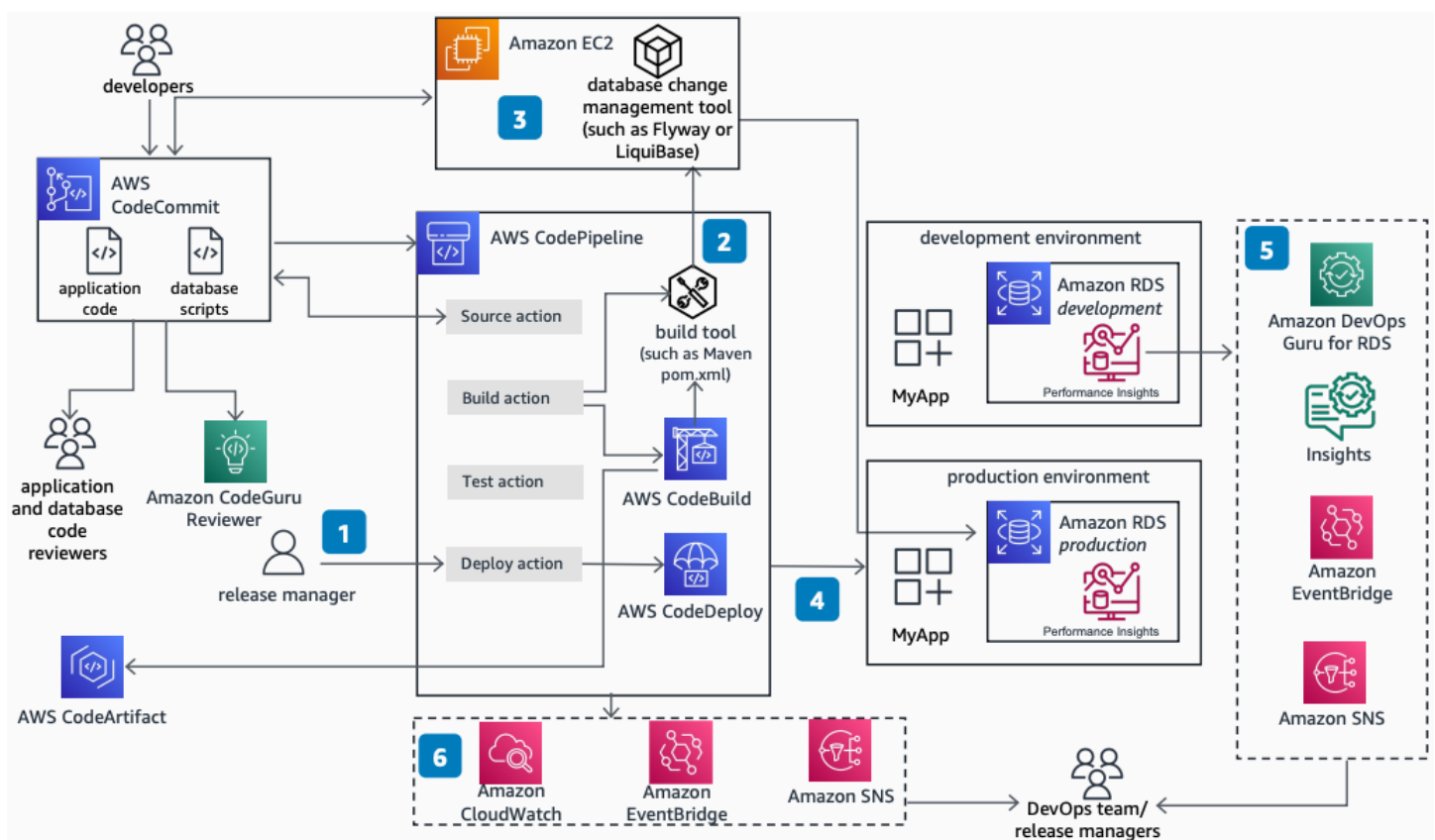
Per iscriverti agli aggiornamenti RSS, devi avere un plugin RSS abilitato per il browser che stai utilizzando.

Database DevOps su AWS: flusso di lavoro di produzione

Data di pubblicazione: 31 marzo 2023 () [Storia del diagramma](#)

Utilizzate questa architettura nel flusso di lavoro di produzione per automatizzare le modifiche al database come parte della DevOps pratica. Crea un'orchestrazione per implementare le modifiche al database alla stessa velocità del codice dell'applicazione, rilevare comportamenti anomali del sistema e automatizzare le notifiche ai team designati affinché intraprendano azioni correttive.

Database su AWS: flusso di lavoro di produzione DevOps



I passaggi seguenti descrivono l'architettura:

1. Il release manager emette una richiesta di distribuzione in produzione.
2. [CodePipeline](#) richiama lo strumento di gestione delle build che richiama lo strumento di gestione delle modifiche al database.
3. Lo strumento di modifica del database scarica gli script di modifica del database e li esegue sul database di produzione [Amazon RDS](#) di destinazione.

4. CodePipeline avvia l'azione Deploy per distribuire il codice nell'ambiente di destinazione.
5. (Facoltativo) Usa Performance Insights su Amazon RDS e applica automaticamente le tecniche di machine learning Amazon DevOps Guru per rilevare problemi operativi e rallentamenti nelle prestazioni.
6. [EventBridge](#) acquisisce gli eventi e li invia a un argomento di [Amazon SNS](#) per le notifiche agli utenti.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.