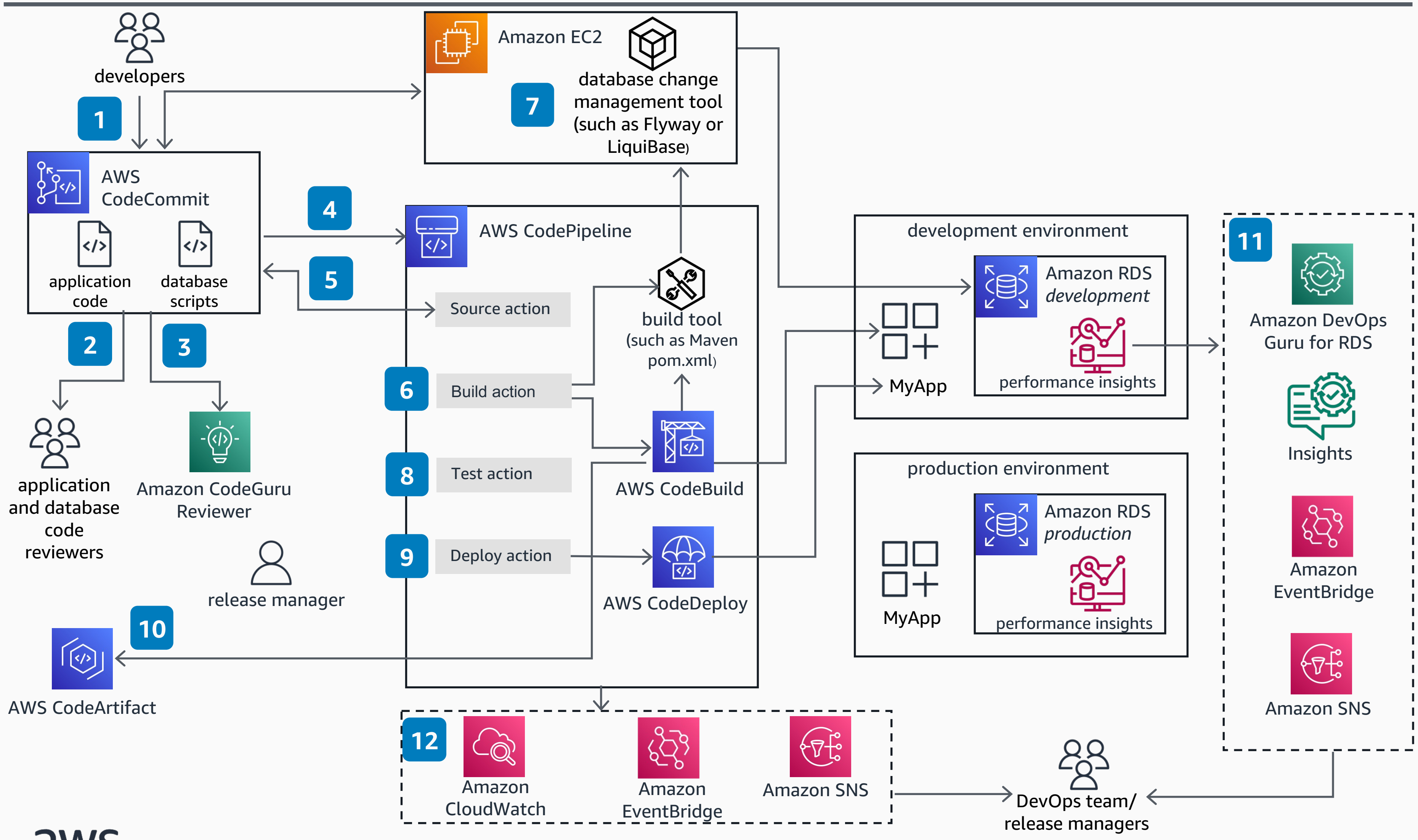
Database DevOps on AWS

Use this architecture in the **development** workflow to automate database changes as part of the DevOps practice. Build an orchestration to deploy database changes at the same rate as the application code, enforce database-application-joint code reviews, integrate database and application code operations, detect abnormal system behavior, and automate notifications to appointed teams to take corrective actions.



1 The developer checks both application and database code into **AWS CodeCommit**.

2 At every push request, **AWS CodeCommit** enforces code reviews through the use of approval rules.

3 **Amazon CodeGuru Reviewer** automatically evaluates code changes, detects errors, and offers code recommendations.

4 **AWS CodeCommit** initiates **AWS CodePipeline** to take pipeline actions.

5 The **AWS CodePipeline** *Source* action checks out application and database code.

6 The **AWS CodePipeline** *Build* action invokes a build management tool to start the application and database code build phase.

7 The build tool calls the database change tool. It downloads the database scripts, and runs them against the target database.

8 The build tool calls database test scripts and validates output.

9 The **AWS CodePipeline** *Deploy* action deploys code to the target environment.

10 Build artifacts are published to **AWS CodeArtifact** for others to consume.

11 (Optional) You can use Performance Insights on **Amazon Relational Database Service** (Amazon RDS) and **Amazon DevOps Guru** to automatically apply machine learning techniques to detect performance bottlenecks and operational issues.

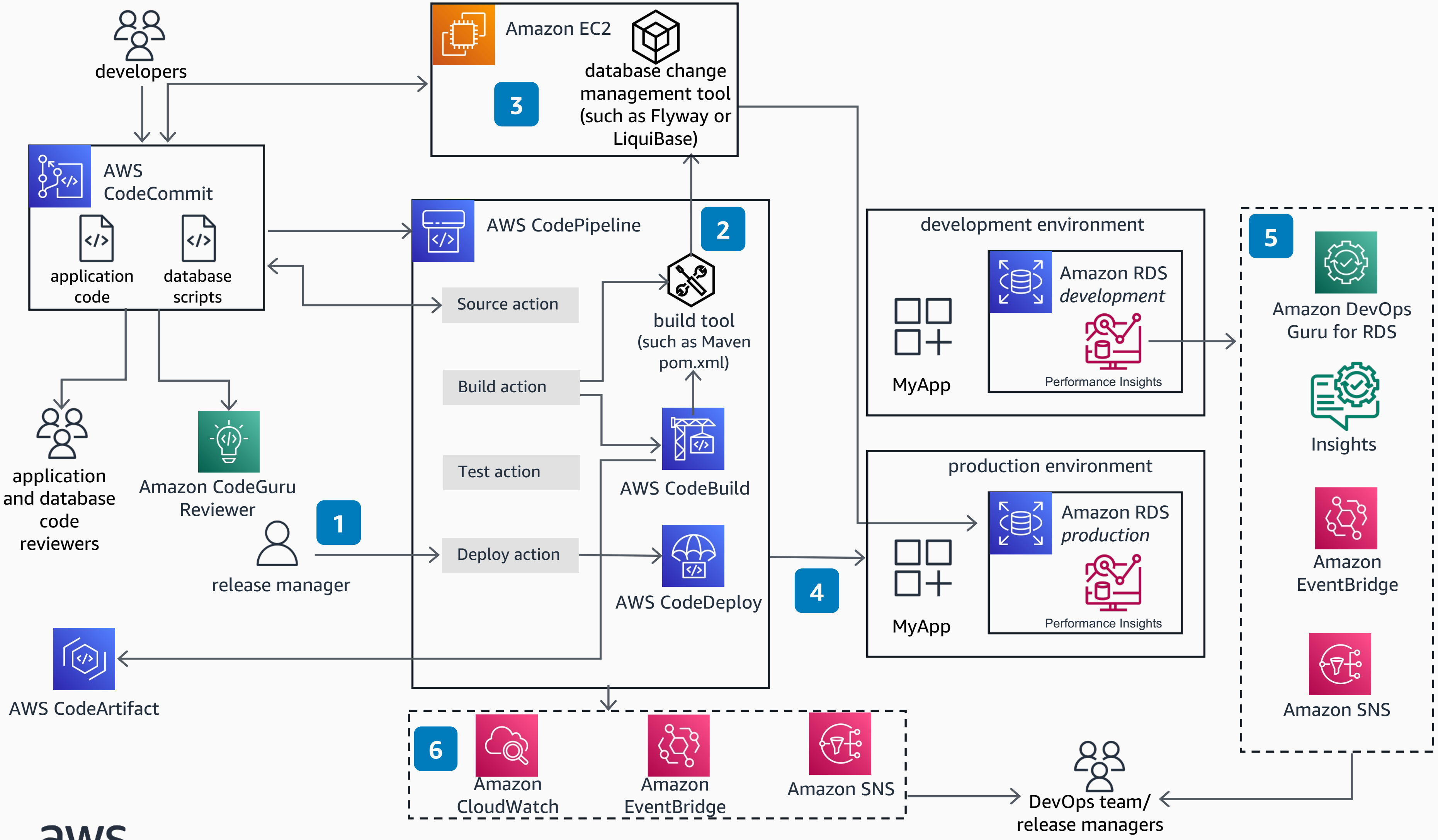
12 **AWS EventBridge** captures events and sends them to an **Amazon Simple Notification Service** (Amazon SNS) topic for user notifications.

Reviewed for technical accuracy March 31, 2023

AWS Reference Architecture

Database DevOps on AWS

Use this architecture in the **production** workflow to automate database changes as part of the DevOps practice. Build an orchestration to deploy database changes at the same rate as the application code, enforce database-application-joint code reviews, integrate database and application code operations, detect abnormal system behavior, and automate notifications to appointed teams to take corrective actions.



1. The release manager issues a production deployment request.
2. **AWS CodePipeline** invokes the build management tool which calls the database change management tool.
3. The database change tool downloads the database change scripts and runs them against the target database.
4. **AWS CodePipeline** initiates the Deploy action to deploy code to the target environment.
5. (Optional) You can use performance Insights on RDS and **Amazon DevOps Guru** to automatically apply machine learning techniques to detect performance bottlenecks and operational issues.
6. **AWS EventBridge** captures events and sends to an **Amazon SNS** topic for user notifications.

Reviewed for technical accuracy March 31, 2023

AWS Reference Architecture