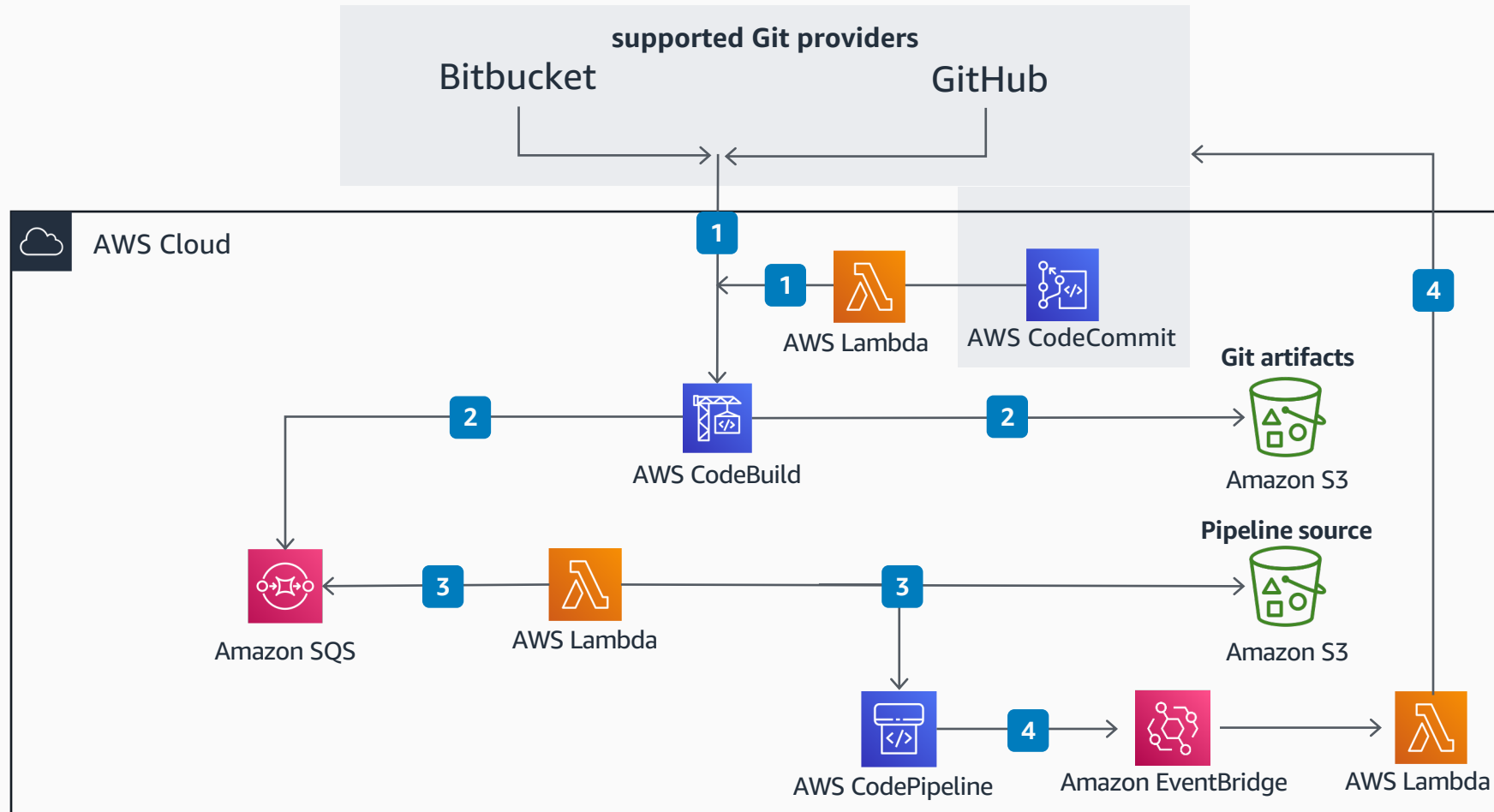# Continuous Integration for Pull Requests

Build a Continuous Integration (CI) Pipeline for Git Pull Requests

Run CI at pull request (PR) time to move from validating a build to validating a whole environment before merging to mainline. This approach allows developers to get higher quality feedback earlier, speed up PR reviews (now enriched by CI outcomes), and minimize impacts of broken builds to development teams.

supported Git providers

Bitbucket   GitHub

1

AWS Cloud

1   AWS Lambda   AWS CodeCommit

2   AWS CodeBuild

Git artifacts
Amazon S3

4

Pipeline source

Amazon SQS   **3**   AWS Lambda   **3**   Amazon S3

AWS CodePipeline   **4**   Amazon EventBridge   AWS Lambda

1. Webhooks are configured between supported Git providers such as GitHub/Bitbucket and **AWS CodeBuild**. The Git provider sends events whenever a PR is created or updated. In the case of **AWS CodeCommit**, it's set up to trigger a Lambda function on those events, which in turn kicks off the build within **AWS CodeBuild**.

2. **AWS CodeBuild** accepts Git PR events and compresses the PR codebase in an archive. This archive is stored in the *Git Artifacts* **Amazon S3** bucket; **AWS CodeBuild** posts a message on an Amazon Simple Queue Service (**Amazon SQS**) queue about a fresh codebase ready to be processed.

3. An **AWS Lambda** function polling the **Amazon SQS** queue consumes the message. The Lambda function starts the CI process and moves the new codebase to the *Pipeline Source* **Amazon S3** bucket location. This S3 bucket acts as the source of **AWS CodePipeline**. The Lambda function also checks any automation prerequisite, like status of third-party resources or environments consumed by CI, and removes obsolete codebase from the backlog if updates are submitted against a PR.

4. **AWS CodePipeline** orchestrates CI workflow for the specific project under development by statically validating the codebase, running unit tests, building artifacts, creating a preview environment where the application is deployed, and executing integration and end-to-end (E2E) tests. **Amazon EventBridge** monitors the changes in pipeline stages; a **Lambda function** updates the PR based on status of events generated by **AWS CodePipeline** (success, in progress, failed, completed).

AWS Reference Architecture