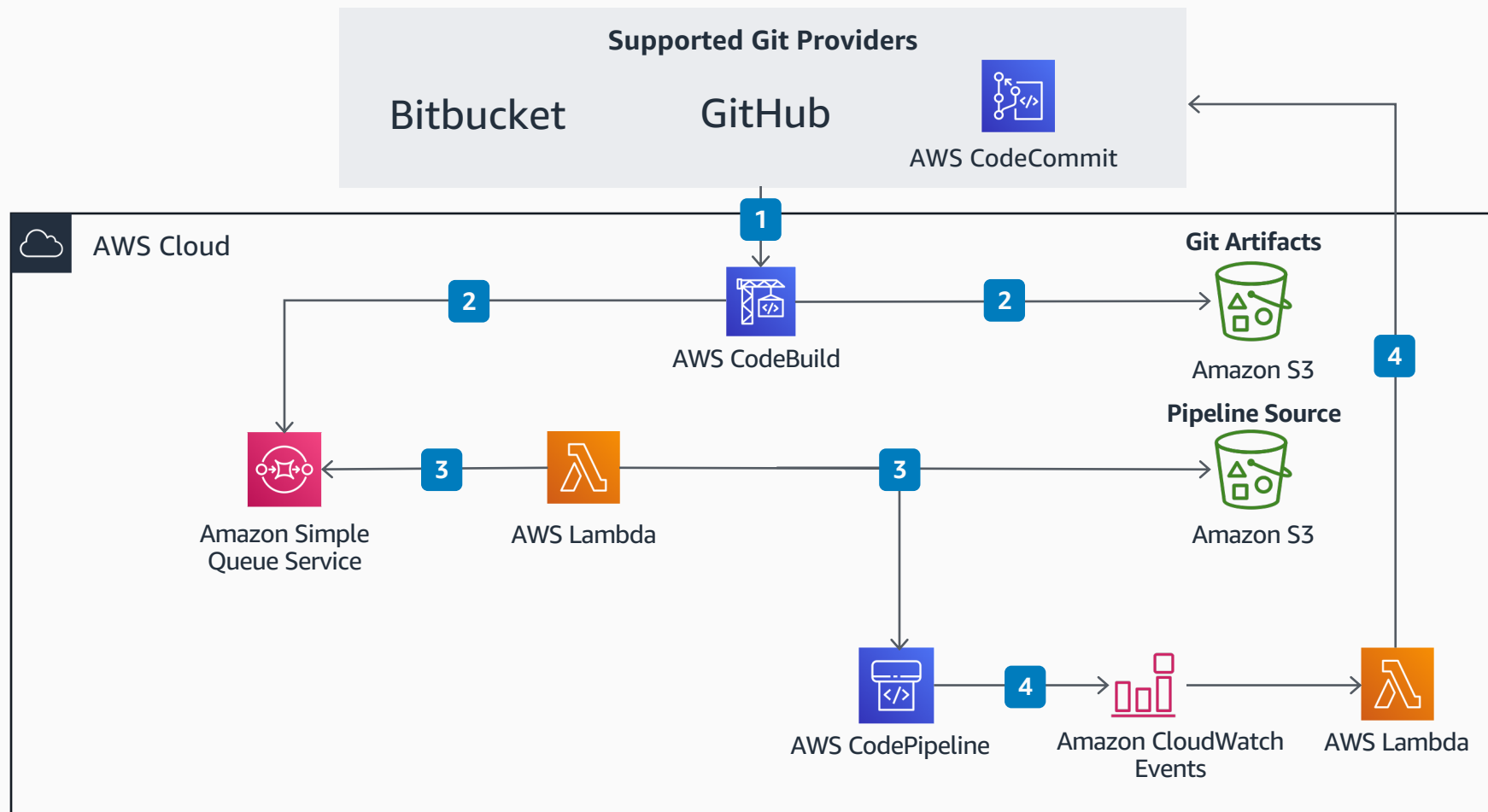# Pull Request Continuous Integration

Build a Continuous Integration/Continuous Delivery (CI\CD) Pipeline for Git Pull Requests

Run CI\CD 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 feedbacks earlier, speed up PR reviews (now enriched by CI\CD outcomes), and minimize impacts of broken builds to development teams.

Supported Git Providers

Bitbucket   GitHub   AWS CodeCommit

AWS Cloud

1 AWS CodeBuild

2 → **Git Artifacts** Amazon S3

2 → Amazon Simple Queue Service

3 ← AWS Lambda

3 → **Pipeline Source** Amazon S3

4

AWS CodePipeline **4** → Amazon CloudWatch Events → AWS Lambda

---

1 Webhooks are configured between a supported Git provider and **AWS CodeBuild**. The Git provider sends events whenever a PR is created or updated.

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 a **Amazon SQS** queue about a fresh codebase ready to be processed.

3 An **AWS Lambda** function is triggered by **Amazon SQS** whenever a new codebase is ready to be processed. The Lambda function starts the CI\CD 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\CD, and removes obsolete codebase from the backlog if updates are submitted against a PR.

4 **AWS CodePipeline** orchestrates CI\CD 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 CloudWatch Events** 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