

Understanding CI/CD Pipelines



By Edwin Choi

What is CI/CD?

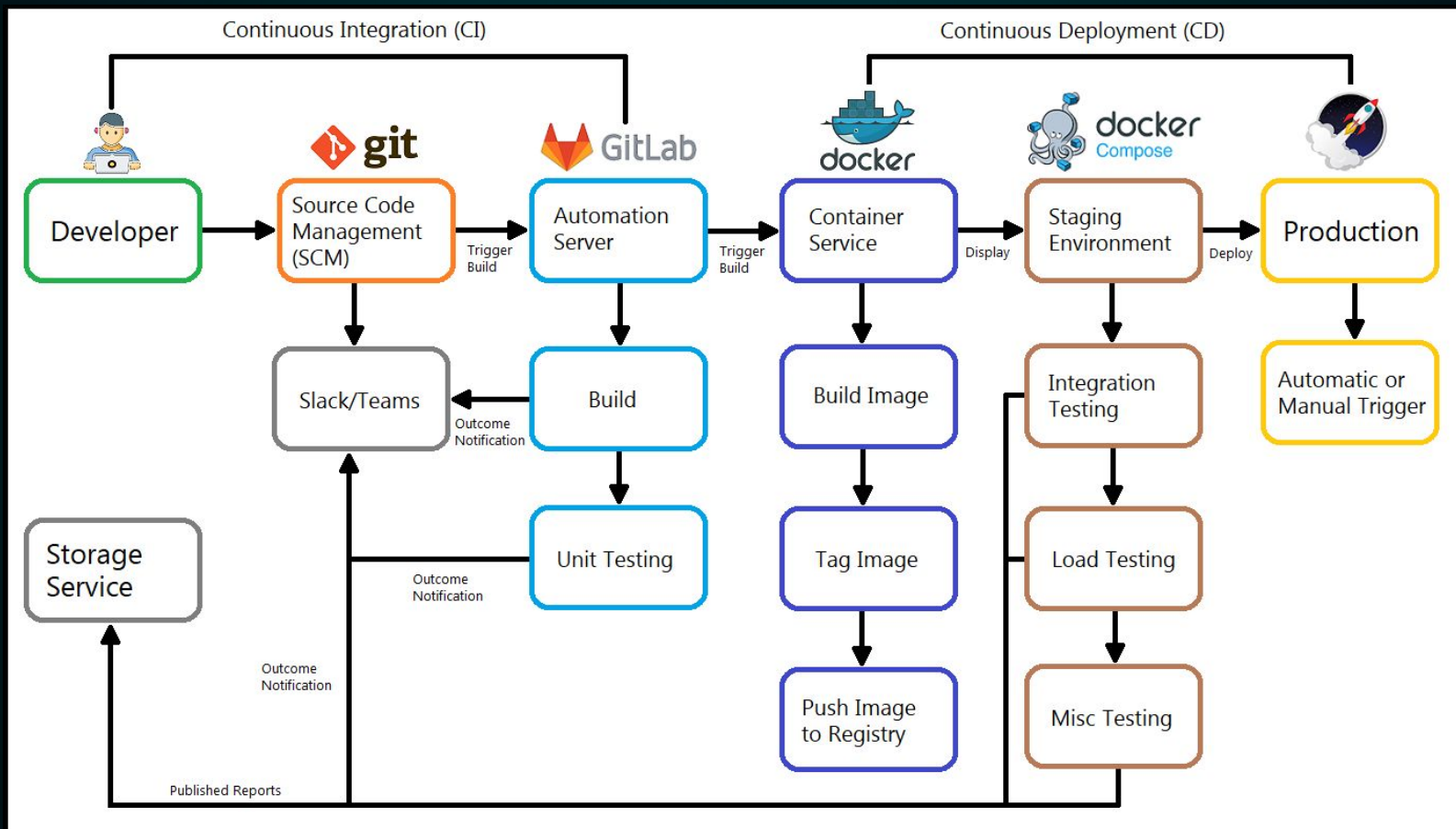
- CI/CD is an abbreviation for **Continuous Integration** and **Continuous Deployment**. It is a set of processes used to automate the building, testing, and deployment of software applications.
- The primary goal of CI/CD pipelines is to:
 - Streamline the software delivery process
 - Improve collaboration
 - Accelerate the time needed to deliver software changes to production

Breakdown of Continuous Integration (CI)

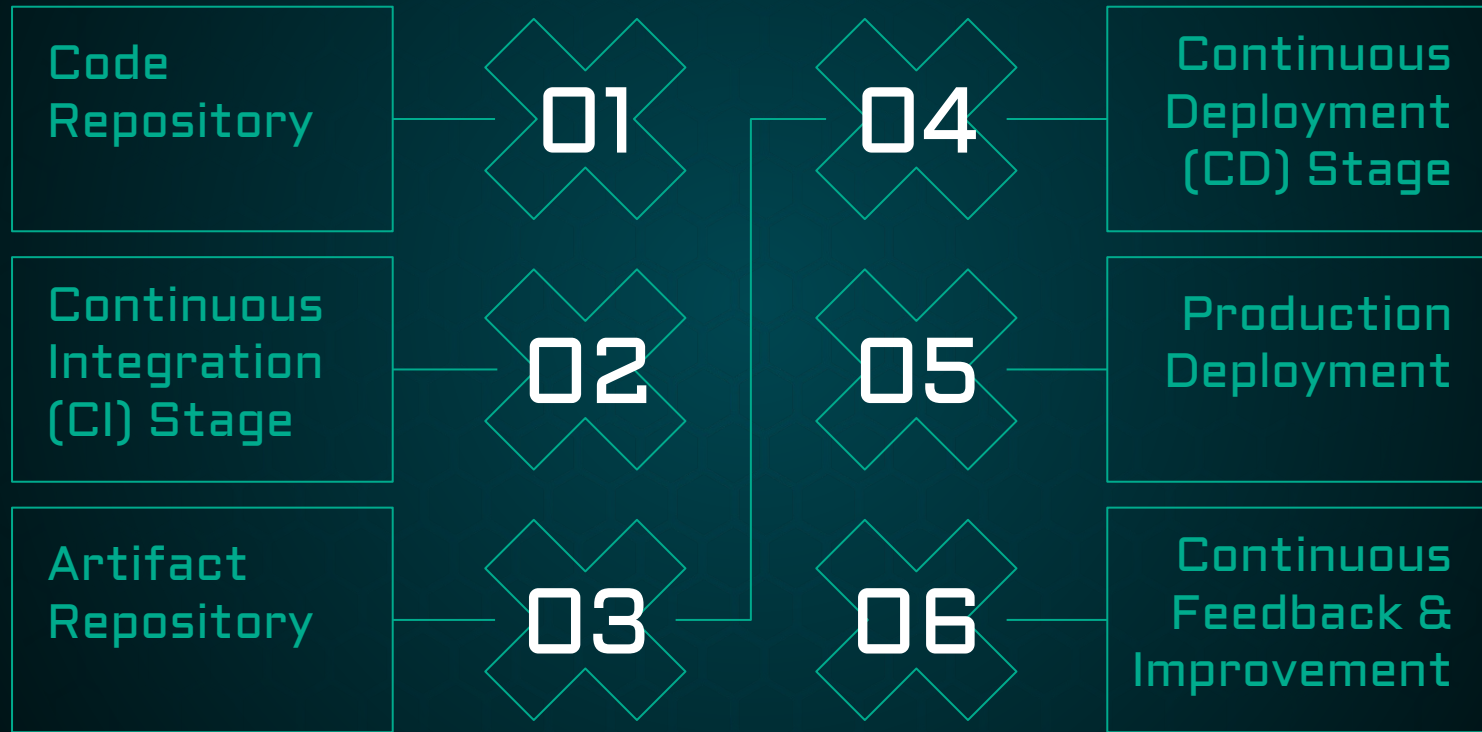
- CI involves integrating code changes from multiple developers into a shared repository, such as Git, regularly. This ensures that code changes are validated and merged together as frequently as possible.
- CI pipelines usually include steps such as compiling code, running automated tests, and checking for code quality and style.
- Through the practice of CI, teams can identify issues or conflicts early on, thereby enabling faster feedback loops and mitigating the risk of integration problems.

Breakdown of Continuous Deployment (CD)

- CD focuses on automating the deployment and delivery of software changes to production environments.
- CD pipelines include the additional stages that follow the CI process, such as deploying the app to staging environments, running additional tests, and then deploying the app to production if all tests are successfully passed.
- CD pipelines also help ensure that software changes are delivered quickly, reliably, and with minimal manual intervention.



CI/CD Pipeline Process



Code Repository

The CI/CD pipeline begins with a code repository, such as Git, Subversion, and Mercurial, where developers commit their code changes.



Continuous Integration (CI)

Code Build

Upon code changes being pushed to the repo, the CI pipeline triggers a code build.

ie. compilation,
dependency resolution,
executable artifact
generation

Unit Testing

Automated tests are then run to ensure that individual components function correctly and meet requirements.

Static Code Analysis

SCA tools scan the codebase for potential bugs, security vulnerabilities, and to ensure the code meets predefined standards.

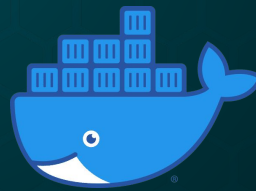
Code Coverage

The CI pipeline collects code coverage metrics to measure the efficiency of unit tests.

Artifact Repository

The successfully built and tested artifacts are stored in an artifact repository, such as Nexus, JFrog, or Docker Registry, for future use.

Artifacts can be classified as compiled code, libraries, binaries, executables, or any other deliverable produced during the building and testing phases.



Continuous Deployment (CD)

Staging Environment

The CD pipeline deploys the built artifacts to a staging environment, which is similar to the production environment.

It is here that additional tests are performed. (ie. integration, load, etc)

Security Scanning

Security tools like SAST and DAST are used to identify vulnerabilities and ensure the app meets security needs.

(S)(D)AST: application security testing, **static** or **dynamic**

UAT

User application testing, or UAT, is where the staging environment may become accessible to end-users for further validation.

Approval Gates

In some cases, manual approvals may be required at specific stages before the app can progress into production.

Production Deployment

Once all the necessary tests and approvals have been successfully completed, the CD pipeline deploys the tested and validated artifacts to the production environment.

Monitoring & Alerting

Continuous monitoring tools are then employed to track the app's performance, availability, and security in the production environment.

Continuous Feedback & Improvement

Throughout the CI/CD pipeline, various metrics and logs are collected to provide continuous feedback on the code quality, test results, security vulnerabilities, and deployment success rates.

The DevSecOps team analyzes these metrics and collaborates with development, operations, and security teams in order to:

- Identify areas for improvement.
- Optimize the pipeline.
- Enhance the overall delivery process.

Overview

- The CI/CD pipeline ensures that code changes go through rigorous testing, security checks, and a controlled deployment process before reaching the production stage.
- This practice emphasizes collaboration, automation, and continuous improvement, which coincide with the principles of DevSecOps to deliver secure, reliable, and high-quality software solutions.

Cited Works

GitLab - What is CI/CD?

<https://about.gitlab.com/topics/ci-cd/>

Red Hat - What is a CI/CD pipeline?

<https://www.redhat.com/en/topics/devops/what-cicd-pipeline>

IBM - What Are CI/CD and the CI/CD Pipeline?

<https://www.ibm.com/cloud/blog/ci-cd-pipeline>

Slidesgo - Slide template

<https://slidesgo.com/theme/software-testing-company-infographics#search-futuristic&position-12&results-236&rs=search>