

Do we need CI/CD?

By: Fady Habib

What we will discuss?

- 1- what is CI/CD?
- 2- what is CI/CD Pipeline?
- 3- Benefits of CI
- 4- Benefits of CD
- 5- Tools

what is CI/CD?

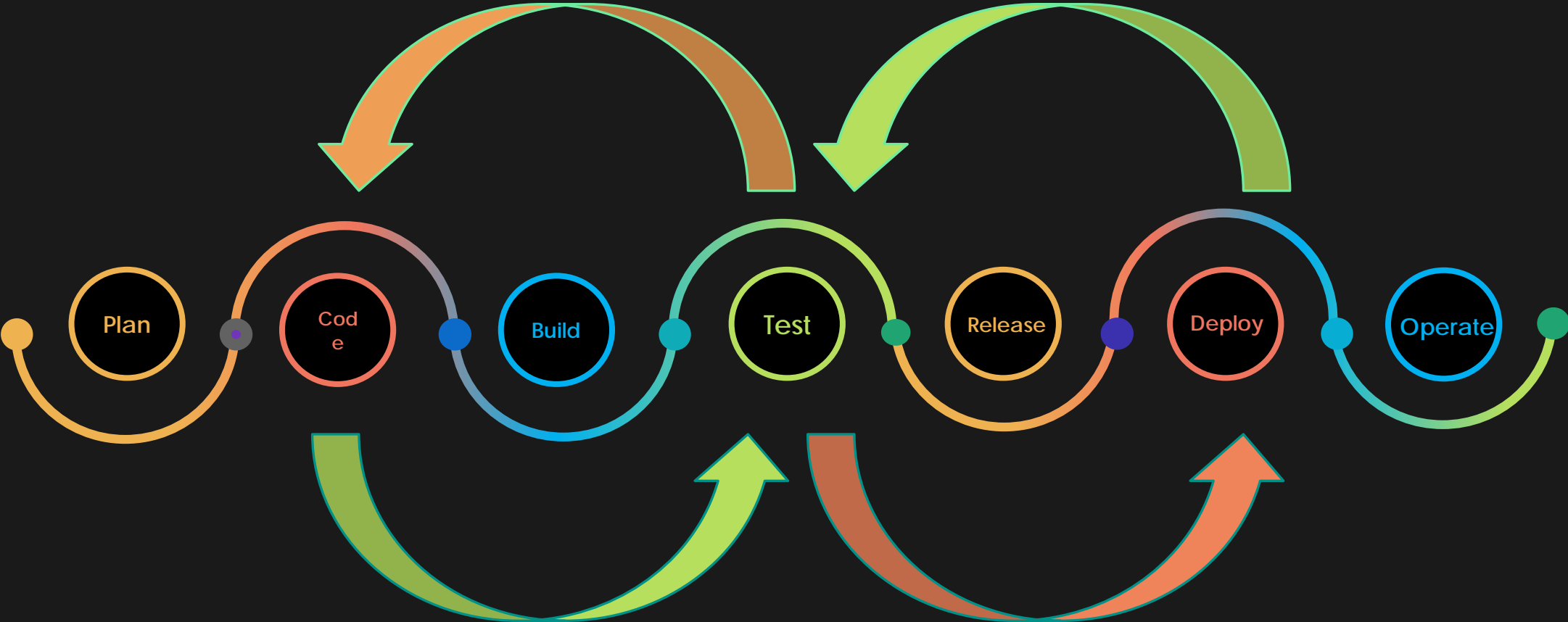
CONTINUOUS INTEGRATION + **Continuous Delivery** = **Continuous Deployment**

is a practice that encourages developers to integrate their code into a main branch of a shared repository early and often. Instead of building out features in isolation and integrating them at the end of a development cycle, code is integrated with the shared repository by each developer multiple times throughout the day

is an extension of continuous integration. It focuses on automating the software delivery process so that teams can easily and confidently deploy their code to production at any time. By ensuring that the codebase is always in a deployable state, releasing software becomes an unremarkable event, without any complicated rituals. Teams can be confident that they can release whenever they need to without complex coordination or late-stage testing. As with continuous integration, continuous delivery is a practice that requires a mixture of technical and organizational improvements to be effective.

is an extension of continuous delivery that automatically deploys each build that passes the full test cycle. Instead of waiting for a human gatekeeper to decide what and when to deploy to production, a continuous deployment system deploys everything that has successfully traversed the deployment pipeline. Keep in mind that when new code is automatically deployed, new features can still be activated conditionally at a later time or for a subset of users. Deploying automatically pushes features and fixes to customers quickly, encourages smaller changes with limited scope, and helps avoid confusion over what is currently deployed to production.

CI/CD PIPELINE



CONTINUOUS INTEGRATION

CONTINUOUS DELEVERY

Continuous Integration Benefits

Minimize the cost of integration

by making it an early consideration. Developers can discover conflicts at the boundaries between new and existing code early, while conflicts are still relatively easy to reconcile. Once the conflict is resolved, work can continue with confidence that the new code honors the requirements of the existing codebase.

Integrating code frequently

integration is costly because manual processes are used to ensure that the code meets standards, does not introduce bugs, and does not break existing functionality. Frequent integration can create friction when an approach to automation does not align with quality assurance measures in place.

Test suites and an automated system

When a developer merges code into the main repository, automated processes kick off a build of the new code. Afterwards, test suites are run against the new build to check whether any integration problems were introduced. If either the build or the test phase fails, the team is alerted so that they can work to fix the build.

Make integration a simple, repeatable process

that is part of the everyday development workflow in order to reduce integration costs and respond to defects early. Working to make sure the system is robust, automated, and fast while cultivating a team culture that encourages frequent iteration and responsiveness to build issues is fundamental to CI success.

Continuous Delivery Benefits

Automate the Software Release Process

Continuous delivery lets your team automatically build, test, and prepare code changes for release to production so that your software delivery is more efficient and rapid.

Improve Developer Productivity

These practices help your team be more productive by freeing developers from manual tasks and encouraging behaviors that help reduce the number of errors and bugs deployed to customers.

Find and Address Bugs Quicker

Your team can discover and address bugs earlier before they grow into larger problems later with more frequent and comprehensive testing. Continuous delivery lets you more easily perform additional types of tests on your code because the entire process has been automated.

Deliver Updates Faster

Continuous delivery helps your team deliver updates to customers faster and more frequently. When continuous delivery is implemented properly, you will always have a deployment-ready build artifact that has passed through a standardized test process.

Most Popular CI/CD Tools



Jenkins



circleci



GitLab



TeamCity



Bamboo



Travis CI