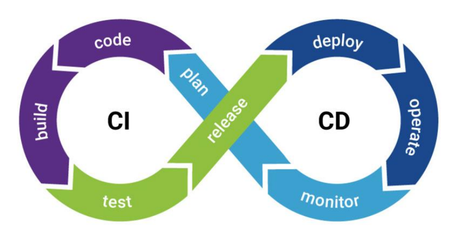
Benefits of CI/CD Implementation



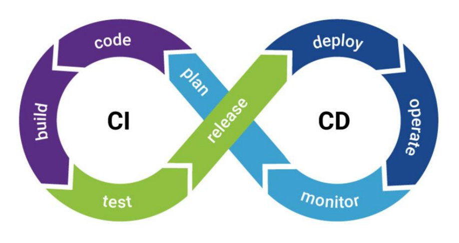
What is CI/CD?

Continuous integration (CI)

It is a software development process where teams integrate code early and often into a central repository where they can run frequent tests and validate changes. The goal is to speed up the release process by enabling teams to identify and fix problems early in the development lifecycle.

Continuous Delivery (CD)

It is the practice of getting all updates, fixes, features, and configuration changes either into production or into the hands of end-users as quickly (and safely) as possible. The goal is to streamline the delivery/deployment process so that predictable tasks can be performed on-demand.



Why should we implement CI/CD?

Improved code quality

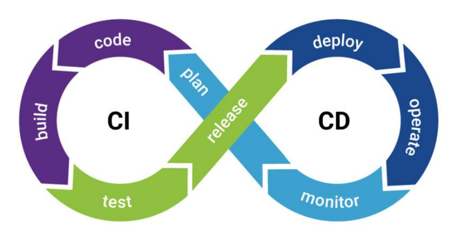
Since code is released in small batches, it is possible to test it thoroughly, detecting and fixing the most serious bugs before software is deployed to production. Gradual changes to the code are also more manageable.

Shorter time-to-market of new features

If code changes are small, we can release software builds faster—essentially, on demand. This improves flexibility and the ability to ship new functionalities. Now when users ask for new features or competitors introduce enhancements, our reaction can be faster.

Automation to reduce costs and labour

Build automation, automated testing and deployment not only make the life of the DevOps team easier, but also reduces costs. Automated actions are also less susceptible to human error and easier to manage.



Why should we implement CI/CD?

Enables rapid feedback

When automated tests detect a bug, it can be fixed rapidly. The same is true for automated deployment: if the metrics being monitored suggest there’s an issue with a new update, it will automatically be rolled back, thus protecting the company’s reputation from any harm it may suffer due to faulty software.

Gain Real-Time Visibility of the Development Process

Stakeholders can easily see where a project stands at any given moment—spot bottlenecks, inefficiencies, etc., and use those insights to optimize the process. You can also see the history of deployments and success rates, which can inform the direction of future projects, help teams plan ahead, and keep everyone focused on the tasks that create the most value.

Improved customer satisfaction

Buggy software can harm the company's reputation, sometimes irreparably. Fast and frequent releases, new features shipped regularly, bugs fixed promptly and immediate reaction to feedback—these are the major factors that will make end-users happy to pay for our software.