CI/CD

Continuous integration (CI) and continuous delivery (CD), also known as CI/CD, embodies a culture, operating principles, and a set of practices that application development teams use to deliver code changes more frequently and reliably.

By automating integration and delivery, CI/CD lets software development teams focus on meeting business requirements while ensuring code quality and software security.

Why do we need CI/CD

CI/CD employs tools and practices that facilitate the release of value-generating features quickly and frequently resulting in a significant increase in revenue

Avoids error-prone manual checks done by humans before deployment in favor of automated checks and deployments that helps our product features reach market in less time boosting revenue.

Why do we need CI/CD

Issues that emerge upon integrating new developer code to existing code base force our teams to spend more time resolving these issues instead of focusing on product features.

CI/CD makes identifying such integration issues easier by isolating the source of the issues and referring each issues to the responsible team member resulting in faster issue resolutions cutting down the time and cost spent on identifying and solving merging issues.

Why do we need CI/CD

Unit tests help ensure our code is ready for production and functions as expected. Using CI/CD to automate unit tests enhances the speed and frequency at which unit tests are performed against our production code increasing the quality of our software with less or no bugs. This way CI/CD will help us avoid costs related to fixing bugs on production and manual testing.

Why do we need CI/CD

Ultimately, a CI/CD pipeline gives developers the power to fail fast and recover even faster. It's a simple process to push code into production and, if there are issues, simply roll it back. The ability to easily rollback code saves teams time, energy, and resources and leads to faster fixes of problem code