CI CD

Mohamed Mesbah

WHAT IS CI / CD ?

CI and CD stand for continuous integration and continuous delivery/continuous deployment. allows organizations to ship software quickly and efficiently. CI/CD facilitates an effective process for getting products to market faster than ever before, continuously delivering code into production, and ensuring an ongoing flow of new features and bug fixes via the most efficient delivery method.

WHAT IS CI ?

The CI in CI/CD stands for continuous integration. Continuous integration means that developers frequently merge their code changes to a shared repository. It’s an automated process that allows multiple developers to contribute software components to the same project without integration conflicts. CI involves automated testing whenever a software change is integrated into the repository.

WHAT IS CD ?

CD can stand for either continuous delivery or continuous deployment. Both involve taking the code continuously integrated and getting it able to deploy to an environment either QA or production. Continuous deployment takes the process one step further and performs the actual deployment to an environment.

BENEFITS

CI/CD facilitates a faster time to market.

The primary goal of a CI/CD pipeline is to deliver working software to users quickly and frequently. Tech giants may have led the way, adopting Agile and DevOps techniques to transform their development processes and deliver constant improvements to their users

Reduced risk

Having a shorter time to market doesn't just help you keep up with the competition. Rapid releases provide an opportunity for product managers and marketing professionals to engage more closely with the development process.

Better code quality

Testing your code's behavior is an essential step in the software release process but doing it thoroughly can also be extremely time consuming. A central part of any CI/CD pipeline is a series of automated tests that are run on each and every build. Although writing automated tests requires an investment of time and expertise, doing so pays significant dividends.

Smoother path to production

As we all know, practice makes perfect, and what's true of shooting hoops or mastering scales also applies to software releases. Adopting CI/CD is best done incrementally, starting with CI practices and building up your pipeline over time. As you start deploying changes more frequently you'll identify pain points and steps in your current process that slow you down, such as refreshing data in a test environment or having to reconfigure parameters before deploying on a particular machine.

Efficient infrastructure

Automation is a central part of any CI/CD pipeline, serving to make the release process repeatable and reliable. In the early stages of implementing continuous integration, your focus will be on automating the build process and on writing and running automated tests. Once you've established a solid CI foundation, the next stage is to automate deployment of your build to test and staging environments.