CI/CD

Save money & time with automation super powers!

What is CI/CD ?

CI/CD stand for continuous integration and continuous delivery.
CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development
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.

Continuous Integration (CI)

Continuous integration (CI) helps developers merge their code changes back to a shared branch, or “trunk,” more frequently—sometimes even daily. Once a developer’s changes to an application are merged, those changes are validated by automatically building the application and running different levels of automated testing, typically unit and integration tests, to ensure the changes haven’t broken the app. This means testing everything from classes and function to the different modules that comprise the entire app. If automated testing discovers a conflict between new and existing code, CI makes it easier to fix those bugs quickly and often.

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.

CI/CD Benefits

- Make Revenue
  - Faster and More Frequent Production Deployments ensures more quicker releases. Removal of manual checks before deployment means less time to market.
- Protect Revenue
  - Automated smoke test reduces downtime due to deploy related crash or a major bug. Automated rollback due to a job failure means a fast undo from production to working state.

CI/CD Benefits (Cont.)

- Reduction of non-critical defects in backlog
  - By now it's clear CI/CD is a time and money saver, so much so that it gives developers time to work on things they wouldn't normally be able to, such as going back to fix older code and make it cleaner and more efficient. The idea that developers cannot only tackle the backlog (it's called a backlog for a reason after all – who has time for this?), but also work on non-critical defects, is a game-changer brought to teams by DevOps and CI/CD.
- Fail Fast
  - The faster we detect the errors, the faster we act and fix the issues even before it occurs on production, and that would save a lot of time debugging and testing also will save money.

CI/CD Benefits (Cont.)

- Avoid Cost
  - Automation of infrastructure creation hence faster deployment and less human error. Catch unit test failure ensures less bugs in production environment and less time testing. Detecting security vulnerabilities avoids future embarrassment from security attacks.
- Reduce Cost
  - Automation of infrastructure cleanup prevents unwanted cost on unused resources. Catching compile errors after merging reduces time spent on issues from new developer code.