CI/CD

MOHAMED NASSER MOHAMED

WHAT IS CI/CD?

- CI/CD stand for continuous integration and continuous delivery/continuous deployment , combines the practices of continuous integration and continuous delivery .
- CI/CD is a set of practices that automate the building, testing, and deployment stages of software development.
- CI/CD is a set of processes that help software development teams deliver code changes more frequently and reliably.
- CI/CD is part of DevOps, which helps shorten the software development lifecycle.
- CI/CD is a best practice for DevOps and agile development.

WHAT IS THE DIFFERENCE BETWEEN CI AND CD?

- Continuous integration (CI) is the process of automatically building and testing code every time a team member commits code changes to version control. A code commit to the main or trunk branch of a shared repository triggers the automated build system to build, test, and validate the full branch. CI encourages developers to share their code and unit tests by merging their changes into the shared version control repository every time they complete a task.
- Continuous delivery (CD) is defined as the ability to deliver product updates to customers as quickly and frequently as possible. Whether these updates involve simple bug fixes, improved functionality, or a newly designed interface, Continuous Delivery automates the entire software release process. The final decision to deploy to a live production environment can be triggered by the developer/project lead as required.

BENEFITS OF CI/CD

- 1. Automation in the CI/CD pipeline reduces the number of errors that can take place in the many repetitive steps of CI and CD. Doing so also frees up developer time that could be spent on product development as there aren't as many code changes to fix down the road if the error is caught quickly. Another thing to keep in mind: increasing code quality with automation also increases your ROI -> Reduce Costs
- 2. Customer Satisfaction: The advantages of CI/CD do not only fall into the technical aspect but also in an organization scope. The first few moments of a new customer trying out your product is a make-or-break-it moment -> Increase Revenue
- 3. Faster Release Rate: Failures are detected faster and as such, can be repaired faster, leading to increasing release rates. However, frequent releases are possible only if the code is developed in a continuously moving system -> Reduce Costs
- 4. Smaller backlog: Incorporating CI/CD into your organization's development process reduces the number of non-critical defects in your backlog. These small defects are detected prior to production and fixed before being released to end-users . The benefits of solving non-critical issues ahead-of-time are many. For example, your developers have more time to focus on larger problems or improving the system and your testers can focus less on small problems so they can find larger problems before being released. Another benefit (and perhaps the best one) is keeping your customers happy by preventing them from finding many errors in your product -> Increase Revenue