CICD

Continuous Integration Continuous Deployment

CONTINUOUS INTEGRATION

- A software development practice in which developers merge their changes to the main branch many times per day. Each merge triggers an automated code build and test sequence, which ideally runs in less than 10 minutes. A successful CI build may lead to further stages of continuous delivery.

CONTINUOUS DEPLOYMENT

- A strategy for software releases wherein any code commit that passes the automated testing phase is automatically released into the production environment, making changes that are visible to the software's users.

BENEFITS OF CI/CD

- **1. Reduce risk :**Finding and fixing bugs late in the development process is expensive and time-consuming. This is especially true when there are issues with features that have already been released to production. With a CI/CD pipeline, you can test and deploy code more frequently, giving testers the ability to detect issues as soon as they occur and to fix them immediately.

- **2. Deliver faster:** Organizations are moving toward releasing features multiple times a day. This is not an easy task; only a handful of companies like Netflix, Amazon, and Facebook have been able to achieve this goal. But, with a seamless CI/CD pipeline, multiple daily releases can be made a reality. Teams can build, test and deploy features automatically with almost no manual intervention. This is accomplished using various tools, frameworks, and systems.

BENEFITS OF CI/CD

3. Expend less manual effort : To align with the shift-left paradigm, we need automation right from the start. This is also a vital component of having a successful CI/CD implementation. Once you build features and check in code, tests should be automatically triggered to make sure that the new code does not break existing features and that the new features are working correctly

4. Make easier rollbacks:

- One of the biggest advantages of a CI/CD pipeline is you can roll back changes quickly. If any new code changes break the production application, you can immediately return the application to its previous state. Usually, the last successful build gets immediately deployed to prevent production outages.