Fundamentals of CI/CD

Continuous Integration, also known as CI, is the practice of merging all developers' working copies into a shared codebase several times a day. Everything related to the code fits here, and it all culminates in the ultimate goal of CI: a high-quality, deployable artifact! Examples of CI phases include Compile, Unit Test, Static Analysis, Dependency vulnerability testing, Store artifact, etc.

Continuous deployment, also known as CD, is an approach to software engineering in which value is delivered frequently through automated deployments. Everything related to deploying the artifact fits here. Examples of CD phases include: Creating infrastructure, Provisioning servers, Copies, Promotion to production, Smoke Testing (aka Verify), Rollbacks, etc.

Benefits of CI/CD

1. Catching errors in the codebase involves compile and runtime errors through linters and static code analysis tools. This tool helps developers spend less time on issues from new code, which helps with reducing costs.
2. Deployment without manual steps, instead of spending so much time and getting code to production delayed through manual steps, we can deploy code to production automatically without any manual steps. This benefit increases our time to market and, in turn, increases revenue.
3. Automating smoke tests as part of CI/CD helps us catch bugs before they get into production. It helps us reduce the download from a deployment-related crash and bug. Smoke test helps increase the integrity of our apps and protects our revenue.