Fundamentals and Benefits of CI/CD

Samuel Anozie

CI

- Continuous integration (CI) is the practice of integrating all your code changes into the main branch of a shared early and often, automatically testing each change kicking off a build when a commit is made or merged.
- Continuous integration helps developers merge their code changes back to a shared branch, or “trunk,” more frequently.
- With continuous integration, errors and security issues can be identified and fixed more easily, and much earlier in the software development lifecycle.

CD

- Continuous delivery (CD) is a software development practice that works in together with continuous integration to automate the infrastructure provisioning and application release process.
- Once code has been tested and built in the CI process, continuous delivery takes over during the final stages to ensure it can be deployed packaged with everything it needs to any environment at any time.
- With continuous delivery, the software is built so that it can be deployed to production at any time.

FUNDAMENTALS OF CI/CD

- **A single source repository**: One single repository for all developers
- **Frequent check-ins to main branch**: Use small segments of code and merge them into the branch as frequently as possible
- **Self-testing builds:** Testing scripts should ensure that the failure of a test results in a failed build
- **Frequent iterations:** Make small, frequent iterations rather than major changes
- **Automated builds:** Scripts should include everything you need to build from a single command
- **Stable testing environments:** Code should be tested in a cloned version of the production environment
- **Predictable deployments anytime:** Deployments are so routine and low-risk that the team's comfortable doing them anytime
- **Maximum visibility:** Every developer should be able to access the latest executables and see any changes made

BENEFITS OF CI/CD

- **Accelerated time-to-value**: When you can deploy anytime, you can bring products and new features to market faster. Your development costs are lower
- **Hit dates more reliably**: Removing deployment bottlenecks and making deployments predictable can remove a lot of the uncertainty around hitting key dates.
- **Happier users and customers:** Fewer bugs and errors make it into production, so your users and customers have a better experience.
- **Recover faster:** CI/CD makes it easier to fix issues and recover from incidents (MTTR). Continuous deployment practices mean frequent small software updates so when bugs appear, it's easier to pin them down.