

CI/CD

CI/CD stands for continuous integration and continuous deployment, which are fundamental DevOps practices. Continuous integration is the practice of automating the integration of code changes from multiple contributors/developers into a single software project while continuous deployment is all about built code and deployment, where code that has been built and merged to the main branch is shipped to servers. Both of them work hand in hand to ensure the continuous delivery of value to our clients.

There are considerable benefits of CI/CD to the dev. team which, to list a few, include;

1. Faster build times
2. Increased quality of code
3. Faster recovery from botched integrations and deployments
4. Greater transparency and communication within the dev team
5. A unified and consistent build process

These benefits are not limited to the development team but also translate to the company as a whole. Instances of this include being able to;

1. Catch compile errors after merge which reduces the amount of time developers have to spend on debugging. This translates to a **reduction of cost**
2. Catch unit test failures which increases code quality and ensures there are less bugs in production. This reduces the possibilities of downtime from failed builds, translating to a **protection of revenue and avoidance of cost**.
3. Utilize continuous deployment as a strategy. This is the practice of continuously deploying built code. Therefore, using a deployment strategy like the blue-green deployment, we can deploy a new production version while the old is still running while routing users to the new one. This **reduces downtime** overall while also ensuring **customer satisfaction**. Also, if the green deployment fails, we simply rollback to the previous working “blue” version!
4. Create infrastructure automatically, thereby reducing or even completely **eliminating human errors** and enabling faster deployments which will enable our company to **continually provide value** to our clients, **increase satisfaction** and also **avoid cost** while **increasing revenue**.
5. Automate smoke tests and deploy to production without manual checks. This **reduces the timeline** to market, therefore **increasing revenue** as value is being provided to the clients as soon as possible, allowing us to **profit** as fast as possible.