

Benefits of Continuous Integration and Continuous Delivery

CI/CD continuously merges codes and continuously deploys them to production after thorough testing, keeping the code in a release-ready state. It's important to have as part of deployment a production environment set up that closely mimics that which end-users will ultimately be using. Containerization is a great method to test the code in a production environment to test only the area that will be affected by the release.

Designing your system with CI/CD ensures that fault isolations are faster to detect and easier to implement. Fault isolations combine monitoring the system, identifying when the fault occurred, and triggering its location. Thus, the consequences of bugs appearing in the application are limited in scope. Sudden breakdowns and other critical issues can be prevented from occurring with the ability to isolate the problem before it can cause damage to the entire system.

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.

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.

What will happen if we changed?

- Less developer time on issues from new code.
- Less Bugs in production and less testing time
- Minimize security vulnerabilities
- Less human errors and faster deployment

- More frequent releases
- Less time to market
- Less downtime
- Faster rollback in case of failure
- Less costs because we use only resources as needed