

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

CI/CD

Continuous Integration Continuous Deployment

CI/CD Definition

Continuous Integration:

The practice of merging all developers' working copies to a shared mainline several times a day. It's the process of "Making". Everything related to the code fits here, and it all culminates in the ultimate goal of CI: a high quality, deployable artifact!

Continuous Deployment :

A software engineering approach in which the value is delivered frequently through automated deployments. Everything related to deploying the artifact fits here. It's the process of "Moving" the artifact from the shelf to the spotlight

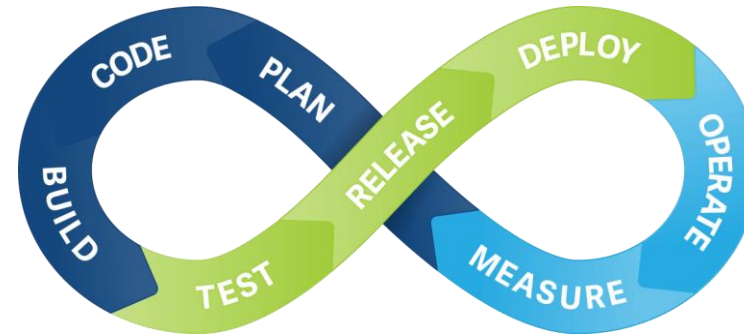


Benefits of CI/CD

- ❑ Automate Infrastructure Creation and clean up: Eliminating human errors and avoid unnecessary cost of unused or invalid infrastructure
- ❑ Faster to production: By automating the pipeline to production this way we can deploy features as soon as created which will help increase revenue
- ❑ Automated Rollback Triggered by Job Failure: Automate the process of rolling back and cleaning any infrastructure left which would help in reducing cost and lower down time
- ❑ Catch Compile Errors After Merge: Discover errors as soon as the developer make his commit which will help reduce the time of developers and reduce cost
- ❑ Catch Unit Test Failures: Unit tests are not neglected with CICD which will increase code quality and catch errors early before production which would decrease cost
- ❑ Automated Smoke Tests: Automate smoke test after deployment and automatic rollback in case of failure which will decrease downtime and reduce cost

Stages of a CI/CD pipeline

- ❑ **Source.** Takes a change made in an app's source code, configuration, environment or data and triggers a new instance of the pipeline.
- ❑ **Build.** The pipeline builds (compiles) the application, creates redistributable packages from the source code, and ensures the code doesn't have any syntax errors and typos deeming it unusable.
- ❑ **Test.** Tests the code, including binaries, configuration, environment and data.
- ❑ **Deploy.** Releases the software to the environment and runs a set of functional tests to verify performance and security.



How it saves Money and time

❑ Reduce Cost

- ❑ Save time for developer in finding and resolve issue that related to source code
- ❑ Detect issue soon ad rollback immediately. This will be reduce risk

❑ Increase Revenue

- ❑ Faster and more frequent production deployments
- ❑ Deploy to production Without manual Checks

❑ Avoid Cost

- ❑ Less bugs in production and less time in testing
- ❑ Prevent embarrassing or costly security holes
- ❑ Less human error, Faster deployments