



CI/CD

CONTINUOUS INTEGRATION
AND CONTINUOUS
DEPLOYMENT

Continuous Integration (CI)

CI: This is 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! Some common CI-related phases might include:

- Compile
- Unit Test
- Static Analysis
- Dependency vulnerability testing
- Store artifact



CONTINUOUS DEPLOYMENT (CD)

CD: 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. Some common CD-related phases might include:

- Creating infrastructure
- Provisioning servers
- Copying files
- Promoting to production
- Smoke Testing (aka Verify)
- Rollbacks



Benefits Of CI/CD

- Catch Compile Errors After Merge: Less developer time on issues from new developer code hence a reduction in cost.
- Catch Unit Test Failures: Less bugs in production and less time in testing hence an avoidance of cost.
- Deploy to Production Without Manual Checks: Less time to market hence an increase in revenue.
- Faster and More Frequent Production Deployments: New value-generating features released more quickly hence an increase in revenue.



Benefits Of CI/CD contd.

- Automate Infrastructure Cleanup: Less infrastructure costs from unused resources and hence a reduction in cost.
- Automated Smoke Tests: Reduced downtime from a deploy-related crash or major bug, this helps protect the revenue.
- Automate Infrastructure Creation: Less human error, Faster deployments which helps avoid cost.
- Automated Rollback Triggered by Job Failure: Quick undo to return production to working state which helps to protect the revenue.

...A penny saved is a penny earned. Thank You.