CI/CD – Empower Your Delivery to Production

Continuous Delivery

Continuous Delivery is the ability to get changes of all types—including new features, configuration changes, bug fixes and experiments—into production, or into the hands of users, safely and quickly in a sustainable way.

Continuous Integration

Continuous Integration (**CI**) is a development practice where developers integrate code into a shared repository frequently, preferably several times a day.

New changes to the code need to be validated, verified, exercised, worked over, massaged and squeezed to see if there are leaks.

We do this by compiling, linting, running unit tests, performing static analysis, checking dependencies for security vulnerabilities and other things.

Once the source code has been built in CI, we're ready to ship it to servers and devices either in the same network or elsewhere.

Continuous Deployment

Continuous Deployment (CD) is a software release process that uses **automated** testing to validate if changes to a codebase are correct and stable for immediate autonomous **deployment** to a production environment.

Continuous Deployment can be an incredible tool in your arsenal. Not only does CD save time, but it opens some unexpected doors that have a ripple effect over the entire organization.

Why do we need Continuous Delivery?

Technical Language	Value	Translation
Catch Compile Errors After Merge	Reduce Cost	Less developer time on issues from new developer code
Catch Unit Test Failures	Avoid Cost	Less bugs in production and less time in testing
Detect Security Vulnerabilities	Avoid Cost	Prevent embarrassing or costly security holes
Automate Infrastructure Creation	Avoid Cost	Less human error, Faster deployments
Automate Infrastructure Cleanup	Reduce Cost	Less infrastructure costs from unused resources
Faster and More Frequent Production Deployments	Increase Revenue	New value-generating features released more quickly
Deploy to Production Without Manual Checks	Increase Revenue	Less time to market
Automated Smoke Tests	Protect Revenue	Reduced downtime from a deploy-related crash or major bug
Automated Rollback Triggered by Job Failure	Protect Revenue	Quick undo to return production to working state