CI CD in a nutshell

parijat purohit

February 2025

1 CI vs. CD in a CI/CD Pipeline

In Continuous Integration (CI), the focus is on automating the process of building and testing code every time a change is made. Developers commit code, which triggers a CI pipeline to run unit tests, check code quality, scan for vulnerabilities, and build the software. If all tests pass and the build succeeds, the output is stored as an artifact, such as a Docker image or compiled binary. Deploying to staging in CI is also possible, allowing you to test the build in an environment that closely resembles production, ensuring integration works as expected before the code moves to the next phase.

In Continuous Delivery (CD), the focus shifts to deploying the build to a staging environment for further testing or to production for live usage. Once the build passes all tests in CI, it's automatically deployed (in Continuous Deployment) or manually approved for deployment (in Continuous Delivery) to staging. After confirming everything works, the code is then deployed to production, ensuring the latest changes are safely available to end users. Post-deployment checks, including health checks and monitoring, are done to ensure the application is functioning as expected.

"Continuous Delivery" refers to automating the software release process up until the point of production, where a manual approval is needed before deployment, whereas "Continuous Deployment" automatically deploys changes to production as soon as they pass all tests, eliminating the need for manual intervention at the final stage.