CI/CD Pipelines

CI/CD is a critical part of modern DevOps practices. It automates the process of software integration, testing, packaging, and deployment.

Continuous Integration (CI)

The purpose of CI is to ensure that new code added by different developers integrates smoothly with the existing codebase.

Steps in CI:

- Developers check in code to version control (e.g., Git)
- Code is merged into the main branch (via pull requests)
- A build process compiles the code and checks for issues
- Application is packaged into deployable format (e.g., JAR, Docker image)

Continuous Deployment/Delivery (CD)

While CI builds and packages the code, CD focuses on releasing it to various environments.

Steps in CD:

- CI provides the tested package
- Code is deployed to a server (automated or manual approval)
- Ensures fast feedback loops and consistent delivery to production

Environment Lifecycle – From Dev to Staging

Development Environment (Dev Env)

This is the initial environment where:

- Developers write new code
- Unit testing and code linting are performed
- Code is pushed regularly for integration
- Frequent builds and error fixes happen here

QA Environment (Quality Assurance)

The QA environment is maintained for detailed testing by the QA team.

Process:

- QA team tests the latest build
- Bugs and issues are logged

- Developers fix the bugs
- Retesting happens until the build is stable
- Once no issues are found, the build is marked GREEN (ready for staging)

QA ensures functional correctness and performance readiness before further promotion.

Staging / Integration Environment

This simulates a real-world environment where all modules work together.

Purpose:

- Integration testing of multiple modules
- Tested by experienced testers or automation scripts
- Ensures APIs, services, and systems work together
- Prepares for user acceptance or demo

Staging acts as a full system rehearsal for production.

UAT (User Acceptance Testing)

This is the phase where business stakeholders or clients interact with the system.

Key Activities:

- Validate if business goals and user needs are met
- Use scripts or manual tests
- Formal sign-off is required before deployment to production

It's the last validation layer from a user's perspective.

PPE (Pre-Production Environment)

PPE is essentially a mirror of the production environment with real data and full configuration.

Purpose:

- Final validation under real-world conditions
- Sometimes used for load testing or performance checks
- Confidence check before go-live
- Continuous sync with production infrastructure

It's treated with caution and often monitored like production.

Production Environment (Prod Env)

This is the LIVE environment, used by end-users.

Highlights:

- Live customer traffic
- Any bugs here have real-world impact
- Emergency fixes (hotfixes) are carefully planned
- Metrics, monitoring, and alerts are critical for stability

Production is where value is delivered, so all processes aim to ensure only the most stable code reaches this stage.