**CICD PIPELINE**

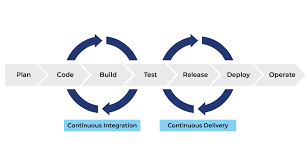
CI/CD stands for Continuous Integration and Continuous Delivery/Deployment.  
It’s a way of automating software development, testing, and release so that changes can move quickly and safely from a developer’s laptop to production.

CI – CONTINUOUS INTEGRATION:

* Developers frequently **merge (integrate)** their code changes into a shared repository (like GitHub, GitLab, Azure DevOps).
* Each change triggers an **automated build and test** process.
* This ensures the new code works well with the existing codebase.

CD – CONTINUOUS DEPLOYMENT:

* After successful CI, the code is automatically prepared for **release**.
* It is deployed to **staging / testing environments** where QA teams or automated tests verify it.
* After tests pass, the system automatically **deploys the changes to production** without manual approval.



STEPS :

1. **Code** – Developer pushes code to Git repository.
2. **Build** – Code is compiled and dependencies installed.
3. **Test** – Unit tests, integration tests, and security checks run.
4. **Package** – Code is packaged into an artifact (e.g., Docker image, .jar, .zip).
5. **Deploy** – Code is deployed to staging/production environment.
6. **Monitor** – Application logs and performance are monitored.

BENEFITS:

* Faster software delivery
* Fewer bugs in production
* Easy rollback if something fails
* Developers focus on coding, not manual deployments
* Consistent, repeatable process

DEVELOPMENT ENVIRONMENT:

Where developers write and test their code.

Fast, flexible, often runs on local machines or a shared dev server.

Can be unstable since new features and fixes are tried here.

QUALITY ASSURANCE ENVIRONMENT:

Code from Dev is moved here.

QA/test engineers run functional, regression, integration tests.

Uses test data (not real customer data).

STAGING ENVIRONMENT:

Almost a **copy of Production** (same configs, same setup).

Final testing happens here (performance, load, security, etc.).

Often connected to real external services (but with test data).

USER ACCEPTANCE TESTING:

Used by **business users / clients** to validate features.

Ensures the system meets business requirements.

Feedback from end-users before production.

PRE PRODUCTION ENVIRONMENT:

Sometimes called **pre-prod** or **sandbox**.

Mirrors production as closely as possible.

Last verification step before going live.

Sometimes used for training or demo.

PRODUCTION ENVIRONMENT:

The **live system** where real users interact.

Must be **stable, secure, and monitored**.

Issues here directly impact customers.

