**CI/CD Pipeline**

CI/CD stands for **Continuous Integration** and **Continuous Delivery/Deployment**. It is a **DevOps practice** that automates the software development lifecycle, enabling teams to deliver applications faster and with fewer errors.

* **CI (Continuous Integration)**: Developers merge their code changes frequently into a shared repository, where automated builds and tests are run.
* **CD (Continuous Delivery/Deployment)**: Automates the release process, allowing updates to be pushed to production or staging environments efficiently.

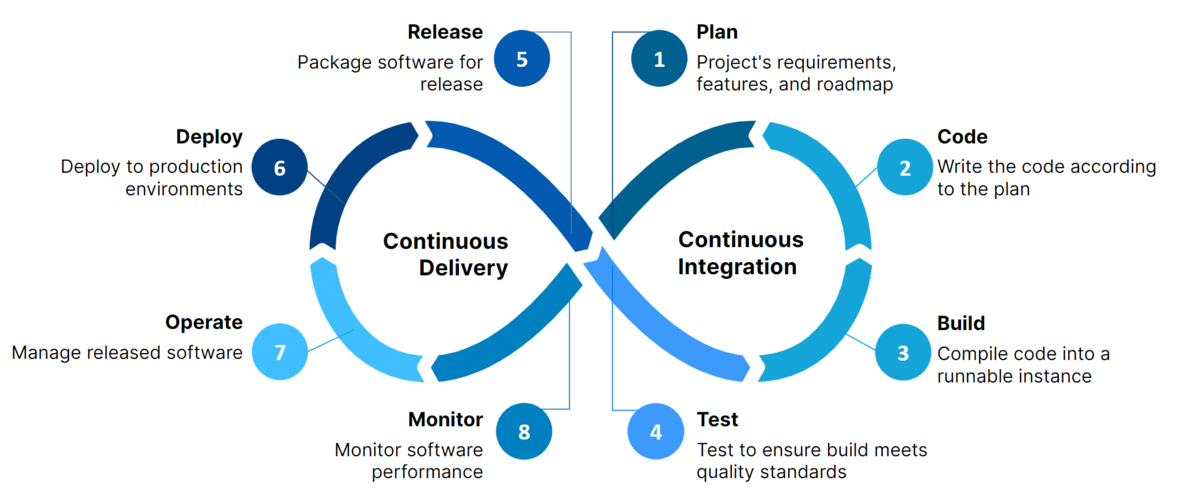
**Why CI/CD?**

CI/CD solves major challenges like:

* Manual errors during deployment
* Long testing and release cycles
* Poor collaboration among development and operations teams

**Key Goals:**

* Deliver software faster and safely
* Minimize manual effort
* Increase release frequency
* Improve code quality



**CI/CD pipeline**is the combination of automated steps that helps software development teams to deliver code in quick, secure and efficient way. In simple words we can CI/CD pipeline is a process of taking your code from development to production environment automatically and smoothly. Let's try to understand it by a simple example:

The following are the CI/CD pipeline steps:

**1. Source Stage**

Riya is a developer who writes code using any IDEs like VSCode or Pycharm. After writing the complete code she pushes that code to shared repository(like Github,Gitlab). The Github stores everything like Code, test scripts, documentation and even build files.

**2. Build Stage**

Once the code is pushed,a CI tool like Github Actions or Jenkins is triggered automatically. The pipeline pulls the code, install all the dependencies, compiles the code and packages it.

**3. Test Stage**

After the build is ready, a series of automated tests are run:

* **Unit tests** to check small pieces of code
* **Integration tests** to check how components work together
* **Static code analysis** by using Synk or SonarQube for security issues.
* **User Acceptance Tests (UAT)**by using tools like Selenium to validate functionality.

If all tests pass, the pipeline continues.

If any tests fails, the pipeline will stop working and report the issue. The test reports are integrated into Github Actions or Jenkins.

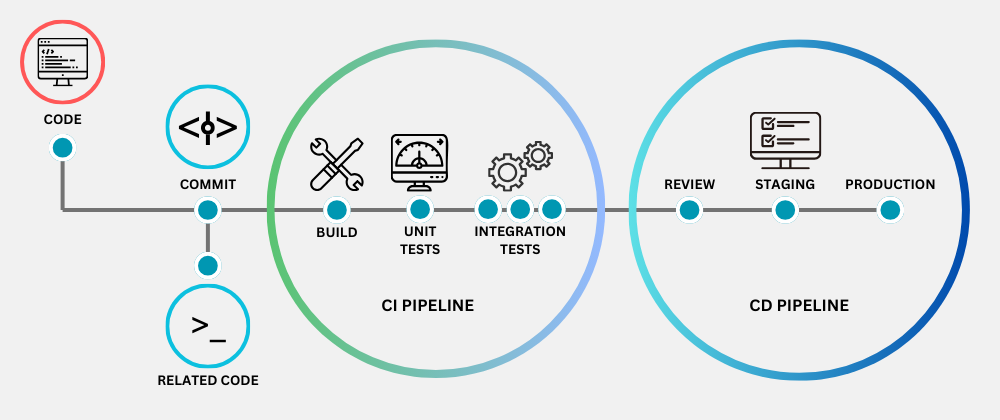
**4. Deployment Stage**

If tests pass, the pipeline deploys the app:

* First, it is deployed to a staging server for final checks.
* After approval, it automatically goes to production server where users can use it.

For example:

* The app is packaged into a **Docker image** and stored in **Docker Hub**
* Then it’s deployed using **Ansible**, which sets up the server and runs the app



**Benefits of CI/CD:**

* **Improves Software Quality & Security:** Automated testing helps catch bugs early, making the software more reliable and secure.
* **Makes Code More Profitable:** By reducing errors, companies spend less time fixing issues and more time delivering value.
* **Speeds Up Product Releases:** CI/CD pipelines help roll out new features faster, keeping customers happy.
* **Reduces Developer Workload:** Automation takes care of repetitive tasks, allowing developers to focus on innovation.
* **Gives a Competitive Edge:** Faster updates help businesses stay ahead of their competitors.
* **Attracts Skilled Professionals:** A well-structured CI/CD process creates an efficient work environment that appeals to top talent.
* **Eliminates Bottlenecks:** By moving away from slow, step-by-step processes, teams can work more efficiently.

**Environments in Azure DevOps**

|  |  |  |
| --- | --- | --- |
| Environment | Typical Use | Deployment Type |
| Dev | Developer testing | Auto after every build |
| QA/Test | Testers validate changes | Manual/Auto |
| Staging | Final pre-prod check | Manual + Approvals |
| Production | Live users | Manual + Appro |

**Popular CI/CD Tools**

|  |  |
| --- | --- |
| Category | Tools |
| CI | Jenkins, GitHub Actions, Azure Pipelines |
| CD | Spinnaker, ArgoCD, GitLab CI/CD |
| Version Control | Git, Bitbucket |
| Monitoring | Prometheus, Grafana, ELK Stack |

CI/CD is a **foundational element of modern DevOps**. It transforms how software is delivered—from slow, risky, and manual processes to **automated, scalable, and collaborative delivery pipelines**.

By adopting CI/CD, teams can respond faster to customer needs, improve software quality, and increase deployment frequency—making it a crucial strategy in today’s competitive software industry.