Digital Career Institute

TDD Concepts

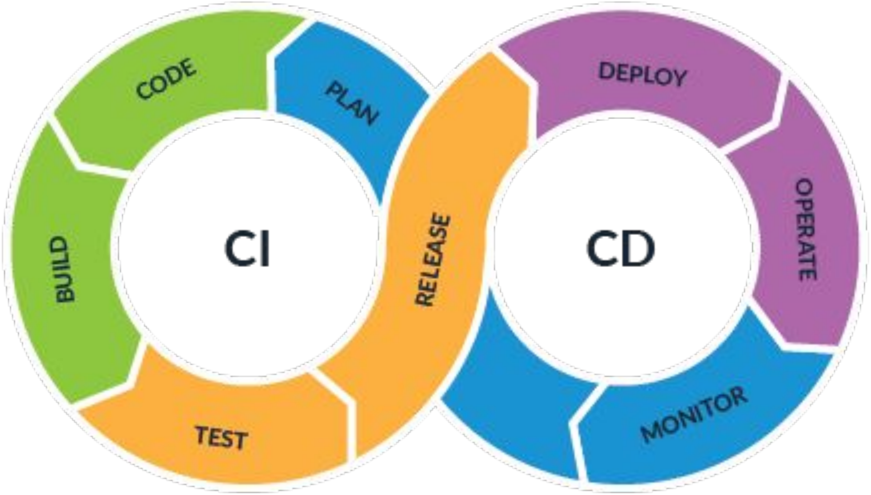
Digital Career Institute

DCI

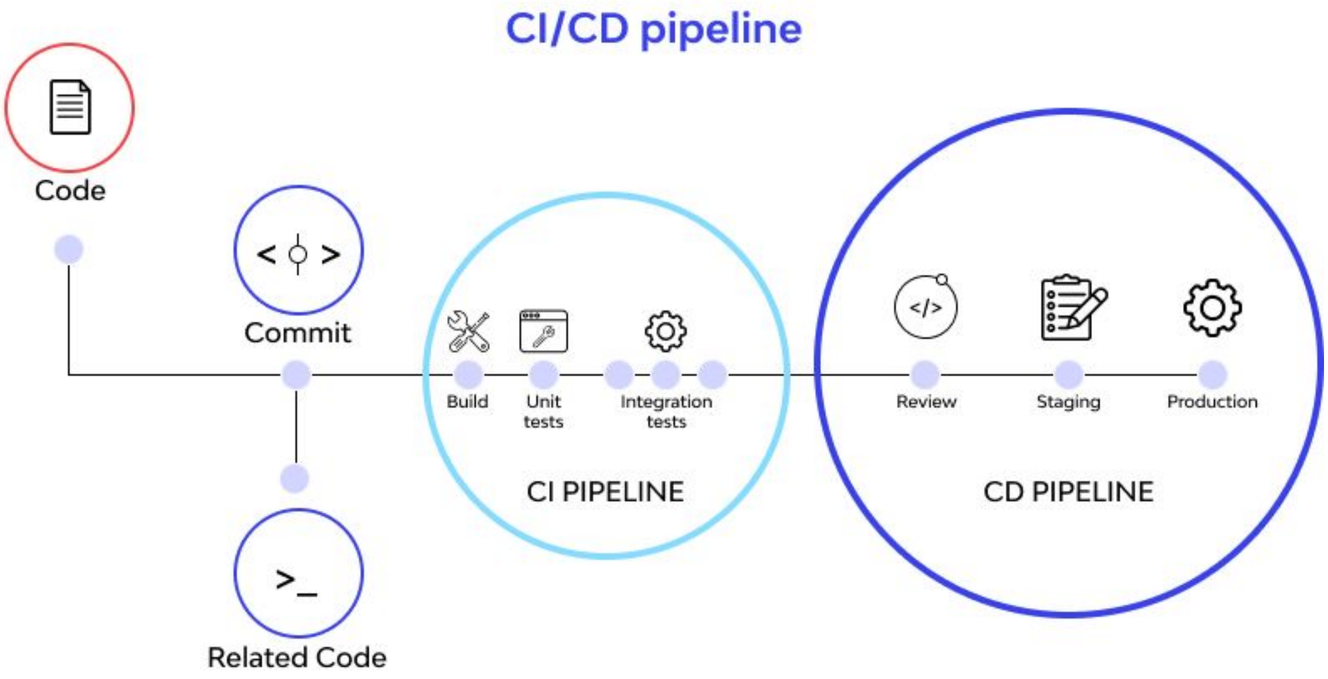
CI/CD

Digital Career Institute

CI/CD



CI/CD


CI/CD pipeline
Code
Commit
Related Code
Build
Unit tests
Integration tests
CI PIPELINE
Review
Staging
Production
CD PIPELINE

What is CI/CD

CI/CD stands for Continuous Integration and Continuous Delivery/Deployment. It's a set of practices and tools that help software development teams to release high-quality code quickly and efficiently. The goal of CI/CD is to automate the entire software development process, from testing to deployment, and to make it repeatable and reliable.

CI vs CD (Deployment vs. Delivery)

Continuous Integration (CI) is the practice of merging code changes from multiple developers into a central repository frequently. The code is then automatically built, tested, and verified for integration issues. CI helps to catch bugs and issues early on in the development cycle.

Continuous Delivery (CD) is the practice of deploying code changes to a staging or production environment automatically. With CD, you ensure that your code changes are always ready for deployment, and you can release new features quickly and efficiently.

Continuous Deployment (CD) is the practice of automatically deploying code changes to production as soon as they pass the testing phase. With CD, you can release changes to your users quickly and with confidence.

Elements of a CI/CD pipeline

A typical CI/CD pipeline includes the following stages:

- Code: Developers write code and commit it to a version control system.
- Build: The code is built and tested in a controlled environment to ensure it works as expected.
- Test: Automated tests are run to ensure that the code works as expected.
- Deploy: The code is deployed to a staging environment for final testing.
- Release: Once the code is tested and verified, it's released to production.

CI/CD tools

There are many CI/CD tools available that can help you set up and automate your pipeline, including:

- Jenkins
- CircleCI
- Travis CI
- GitLab CI/CD
- Azure DevOps

Benefits of pipelines

- Faster release cycles: CI/CD pipelines help you release new features quickly and efficiently, reducing time-to-market.
- Improved quality: Automated testing and validation help you catch bugs and issues early on, reducing the likelihood of defects making it into production.
- Better collaboration: CI/CD encourages collaboration between developers, testers, and operations teams, improving communication and coordination across the entire development cycle.
- Reduced risk: CI/CD pipelines provide visibility and control over the entire software development process, reducing the risk of errors, downtime, and security vulnerabilities.

What makes a good pipeline?

A good CI/CD pipeline is:

- Automated: Manual processes should be minimized or eliminated entirely, to ensure consistency and repeatability.
- Fast: The pipeline should be optimized for speed, so that code changes can be tested and deployed quickly.
- Reliable: The pipeline should be designed to catch errors and issues early on, and ensure that only verified code changes are deployed to production.
- Scalable: The pipeline should be able to handle increasing volumes of code changes and traffic as your application grows.
- Secure: The pipeline should be designed to minimize the risk of security vulnerabilities and protect your code and data from unauthorized access.

Real-world use cases/examples:

- E-commerce websites: E-commerce websites need to release new features and updates frequently to stay competitive. CI/CD pipelines help to ensure that new features are tested thoroughly and released quickly, reducing time-to-market and improving user experience.
- Mobile app development: Mobile app developers can use CI/CD pipelines to automate testing and deployment, reducing the time and effort required to release new versions of the app.
- Cloud infrastructure: Cloud infrastructure providers can use CI/CD pipelines to deploy new services and updates to their infrastructure automatically, ensuring high availability and reducing the risk of downtime.

To get started with CI/CD, you can choose one of the CI/CD tools mentioned above and integrate it with your version control system. Define your pipeline stages, including build, test, and deploy, and automate as much of the process as possible. Test and refine your pipeline to ensure it meets your needs and provides the desired benefits, such as faster release cycles and improved quality.

THANK YOU

Contact Details
DCI Digital Career Institute gGmbH

Digital Career Institute
DCI