ONE PAGE SUMMARY

IT & Software › CI/CD › CI/CD Fundamentals

Introduction to CI/CD Fundamentals

Master CI/CD for seamless software delivery!

DEFINITION

> CI/CD stands for Continuous Integration and Continuous Delivery, practices that automate software development processes. They help teams deliver code changes more frequently and reliably.

KEY CONCEPTS

Continuous Integration

A practice where developers frequently merge code changes into a central repository, ensuring early detection of issues.

Continuous Delivery

An extension of CI that ensures code changes are automatically prepared for release to production.

Automated Testing

Tests that run automatically to verify that new code does not break existing functionality.

Deployment Pipeline

A series of automated processes that code changes go through from development to production.

Version Control

A system that records changes to files over time, allowing for easy collaboration and tracking.

EXAMPLES

→ Using GitHub Actions to automate testing and deployment.

→ Setting up Jenkins to build and deploy applications automatically.

→ Implementing Docker for consistent environments across development and production.

MEMORY TIPS

★ Remember CI as 'Code Integration' and CD as 'Code Delivery'.

★ Think of a pipeline as a conveyor belt that moves code from development to production.

★ Use the acronym 'ACT' for Automated Testing, Continuous Integration, and Continuous Delivery.

COMMON MISTAKES

✗ Neglecting to write automated tests, leading to broken builds.

✗ Failing to regularly merge code, causing integration issues.

✗ Overcomplicating the deployment pipeline, making it harder to manage.

QUICK RECAP

CI/CD is essential for modern software development, enabling faster and more reliable releases. Focus on integrating code frequently, automating tests, and maintaining a smooth deployment pipeline.