ONE PAGE SUMMARY

IT & Software › Full Stack Development › Deployment Pipelines

Introduction to Deployment Pipelines

Streamline your software delivery with effective deployment pipelines.

DEFINITION

> Deployment pipelines are automated processes that help developers build, test, and release software efficiently. They ensure code changes are deployed reliably and quickly to production environments.

KEY CONCEPTS

Continuous Integration

A practice where developers frequently merge code changes into a shared repository, triggering automated builds and tests.

Continuous Delivery

An extension of continuous integration that ensures code is always in a deployable state, allowing for frequent releases.

Automated Testing

Tests that run automatically as part of the deployment pipeline to catch bugs early and ensure code quality.

Version Control

A system that records changes to code over time, enabling collaboration and tracking of code history.

Deployment Automation

The use of tools to automate the deployment process, reducing manual errors and speeding up releases.

EXAMPLES

→ Using Jenkins to automate build and test processes.

→ Implementing GitHub Actions for continuous integration workflows.

→ Deploying applications to AWS with automated scripts.

MEMORY TIPS

★ Remember CI/CD: Continuous Integration leads to Continuous Delivery.

★ Think of a pipeline as a factory line where code flows through stages.

★ Use the acronym VATE: Version control, Automated testing, Deployment automation, and Execution.

COMMON MISTAKES

✗ Neglecting to write automated tests, leading to undetected bugs.

✗ Failing to keep the deployment pipeline updated with new tools.

✗ Overcomplicating the pipeline with unnecessary steps.

QUICK RECAP

Deployment pipelines automate the software delivery process, enhancing efficiency and reliability. Key components include continuous integration, automated testing, and deployment automation, which together ensure smooth releases.