

ONE PAGE SUMMARY

IT & Software > Full Stack Development > CI/CD for Full Stack

Introduction to CI/CD for Full Stack

Streamline Development with CI/CD for Full Stack Applications

DEFINITION

CI/CD stands for Continuous Integration and Continuous Deployment, essential for automating software delivery. It helps developers frequently integrate code changes and deploy them efficiently.

KEY CONCEPTS

Continuous Integration

A practice where developers regularly merge their code changes into a shared repository, ensuring early detection of issues.

Continuous Deployment

An extension of continuous integration where code changes are automatically deployed to production after passing tests.

Version Control

A system that records changes to files over time, allowing multiple developers to collaborate effectively.

Automated Testing

Tests that run automatically to verify that the code behaves as expected, reducing manual testing effort.

Deployment Pipeline

A series of automated processes that code changes go through before being deployed to production.

EXAMPLES

- Using GitHub Actions for CI/CD workflows.
- Deploying a Node.js app to Heroku automatically after a successful build.
- Running unit tests on every pull request to ensure code quality.

MEMORY TIPS

- ★ Remember CI/CD as 'Continuous Improvement/Continuous Delivery'.
- ★ Think of the pipeline as a conveyor belt moving code to production.
- ★ Use the acronym 'VATE' for Version control, Automated testing, and Deployment pipeline.

COMMON MISTAKES

- ✗ Neglecting to write tests, leading to broken deployments.
- ✗ Failing to regularly update the CI/CD pipeline configuration.
- ✗ Overlooking environment differences between development and production.

QUICK RECAP

CI/CD automates the integration and deployment of code changes, enhancing collaboration and efficiency. Key practices include automated testing and maintaining a deployment pipeline to ensure quality and speed.