

CI/CD Pipelines: Complete Learning Guide

1. Background of CI/CD

Continuous Integration (CI) and Continuous Delivery/Deployment (CD) evolved from Agile and DevOps practices in the early 2000s. The goal was to automate software building, testing, and deployment to increase reliability and speed.

Today CI/CD pipelines are core components of modern DevOps, cloud-native development, microservices, and infrastructure automation.

2. How to Learn CI/CD

Step 1: Understand DevOps principles and software lifecycle.

Step 2: Learn Git workflows and branching strategies.

Step 3: Build a basic pipeline (build → test → deploy).

Step 4: Integrate Docker and container registries.

Step 5: Deploy to cloud or Kubernetes.

Step 6: Add monitoring, rollback strategies, and security scanning.

3. Skill Levels

Beginner: Create simple pipelines for build and test.

Intermediate: Multi-stage pipelines, Docker builds, environment management.

Advanced: Kubernetes deployment, blue/green and canary releases.

Architect: Enterprise pipeline design, security integration, GitOps workflows.

4. Certifications

AWS Certified DevOps Engineer.

Azure DevOps Engineer Expert.

Google Professional Cloud DevOps Engineer.

GitLab CI/CD Certifications.

5. Tools for Learning and Practice

CI/CD Platforms: Jenkins, GitHub Actions, GitLab CI/CD, Azure DevOps.

Containers: Docker, Kubernetes.

Artifact Repositories: Nexus, Artifactory.

Monitoring & Security: Prometheus, Grafana, SonarQube.

Conclusion

CI/CD pipelines are essential for DevOps and cloud engineering roles. Mastery requires hands-on labs, automation experience, and understanding deployment strategies.