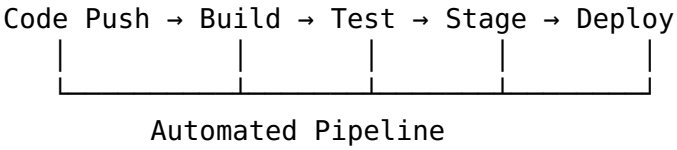
CI/CD Explained — Lesson Notes

What is CI/CD?

- **CI (Continuous Integration):** Automatically build and test code every time a developer pushes changes. Catches bugs early.
- **CD (Continuous Delivery):** Automatically prepare releases for deployment. Every change that passes tests is ready to ship.
- **CD (Continuous Deployment):** Automatically deploy every passing change to production. No manual approval needed.

The CI/CD Pipeline



Stages

Stage	What Happens
Build	Compile code, install dependencies, create artifacts
Test	Run unit tests, integration tests, linting
Stage	Deploy to a staging environment for QA
Deploy	Push to production (manual or automatic)

Popular CI/CD Tools

Tool	Hosted By	Notes
GitHub Actions	GitHub	Built-in, YAML-based
GitLab CI/CD	GitLab	Built-in, powerful pipelines

Tool	Hosted By	Notes
Jenkins	Self-hosted	Highly customizable, plugin ecosystem
CircleCI	Cloud	Fast, Docker-native

Best Practices

1. **Run tests on every push** — never merge broken code.
2. **Keep builds fast** — parallelize tests, cache dependencies.
3. **Use feature branches** — isolate changes with pull requests.
4. **Automate everything** — linting, testing, security scans, deploys.
5. **Monitor after deploy** — use health checks and rollback strategies.

Key Takeaways

1. CI catches bugs before they reach production.
2. CD ensures every passing build is deployable.
3. A good pipeline gives confidence to ship fast.
4. Start simple (build + test), then add stages incrementally.
