

JENKINS

CODE COMPIRATION ?

- Developers write human-readable code
- Computers need machine-readable format
- "Compilation" converts one to the other.

SOURCE CODE MANAGEMENT ?

- ↳ central system to store and manage code
 - ↳ multiple developers, single source of truth
 - ↳ change tracking & roll back
- ex: GitHub, GitLab, Bitbucket

UNIT TESTS ?

- ↳ checks if a small piece of code behaves correctly.

did the developer break anything obvious?

eg: Input = $2+2$

Expected output = 4

Unit test checks if expected output is 4 or not

BUILD ARTIFACT ?

- ↳ final output after code is compiled and packaged

eg:

- jar
- exe
- dockerImage

The same artifact should move from Dev → Stage → Prod

PULL REQUEST ?

why? PR introduces governance and accountability

a formal request to merge new code into the main codebase.

includes:

- code review
- approval
- automated checks

What is Jenkins?

↳ automation tool that runs CI/CD delivery steps automatically whenever code changes.

- code goes in → Tested, packaged CI/CD comes out
- same steps, every time
- measurable, auditable, predictable

Challenges Before Jenkins?

(Manual process)

- Developers compiled code on laptops
- Tests were run "when time allowed"
- Deployments happened at nights or weekends.
- Release depended on "that one expert"
- No single view of build status

Problems

- Human errors
- Finger pointing when broke

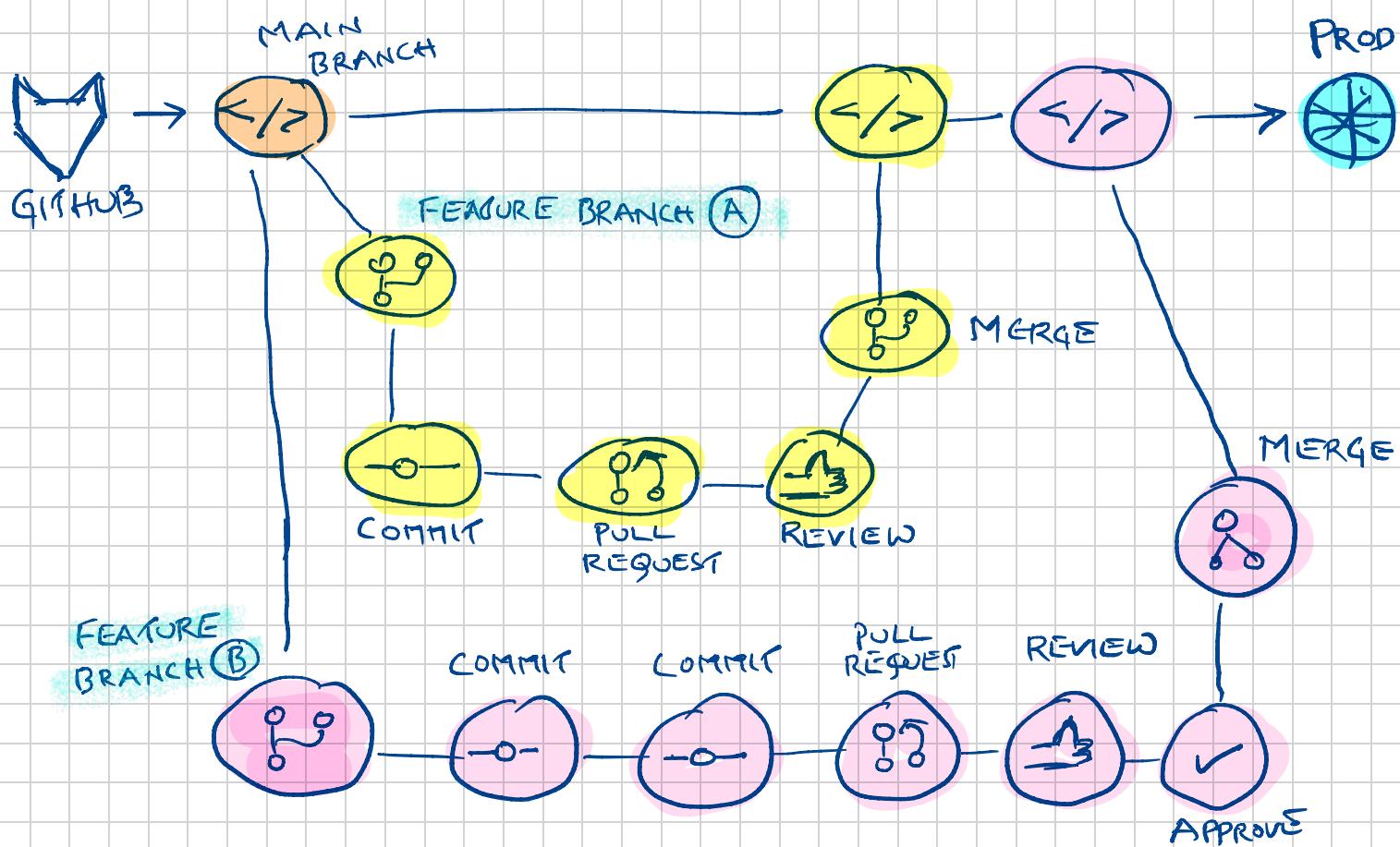
Typical "It worked on my machine"

What Jenkins actually automates?

- Every code change triggers the same pipeline.
- Tests run automatically
- Failures are visible immediately
- Builds are reproducible
 - Auditors can trace who changed what.

Outcome

- Faster releases
- Fewer incidents
- Predictable delivery



Typical Jenkins Flow :

- ① Fetch code
- ② Compile
- ③ Run tests
- ④ Create artifact
- ⑤ Deploy
- ⑥ Notify teams

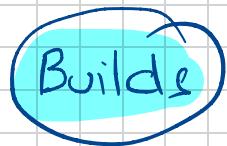
JENKINS Core Concepts



→ Define specific tasks

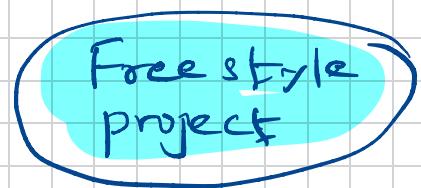


- compiling code
- running tests
- deployment to an environment.

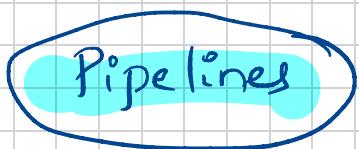


→ each execution of a job

→ maintains a history for troubleshooting



→ The default project type, offering flexible job definition

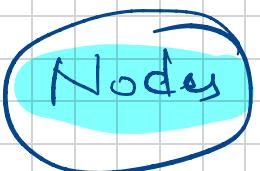


→ chain jobs together to automate tasks from development to deployment.

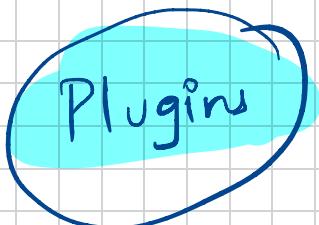
↓ within Pipelines



→ groups related job into phases like "build", "test" and "deploy" for clarity



→ Machines where Jenkins executes jobs, with a single controller or multiple agents



→ Extends Jenkins functionality: integrates with various tools and technologies