

# GitHub Actions CI – Plain English Side-by-Side

This shows each key YAML fragment with a clear explanation. Tweak confidently, knowing what each step does.

YAML Fragment	Plain English
<code>name: CI</code>	Workflow name you'll see in GitHub Actions.
<code>on: pull_request: push: branches: [ main ]</code>	Run this workflow on every Pull Request, and on pushes to `main` (useful to guard direct releases).
<code>jobs: build-and-test: runs-on: ubuntu-latest</code>	Define a single job called `build-and-test`. It runs on a fresh Ubuntu VM container.
<code>steps: - name: Checkout uses: actions/checkout@v4</code>	Fetch your repository code into the runner so following steps can access files.
<code>- name: Setup Node uses: actions/setup-node@v4 with: node-version: '20'</code>	Install Node.js v20 so your project scripts run with a known version.
<code>- name: Setup Supabase CLI uses: supabase/setup-cli@v1</code>	Install Supabase CLI so we can start services and apply migrations in CI.
<code>- name: Start local Supabase services run: supabase start</code>	Bring up a local Supabase stack (Postgres, auth, etc.) in the runner for isolated testing.
<code>- name: Verify migrations from scratch run: supabase db reset</code>	Drop & recreate the DB, then apply all migrations. This proves migrations are clean/idempotent.
<code>- name: Install deps run: npm ci</code>	Clean install Node dependencies using package-lock (reproducible, faster than npm install).
<code>- name: Typecheck run:   if [ -f package.json ] &amp;&amp; cat package.json   jq -e '.scripts.typecheck' &gt; /dev/null; then npm run typecheck else echo "No typecheck script; skipping." fi</code>	Run your TypeScript check if a `typecheck` script exists; otherwise skip gracefully.
<code>- name: Lint run:   if [ -f package.json ] &amp;&amp; cat package.json   jq -e '.scripts.lint' &gt; /dev/null; then npm run lint else echo "No lint script; skipping." fi</code>	Run ESLint (or your linter) if configured to catch style/errors early.
<code>- name: Unit tests run:   if [ -f package.json ] &amp;&amp; cat package.json   jq -e '.scripts.test' &gt; /dev/null; then npm test -- --ci --reporters=default else echo "No test script; skipping." fi</code>	Execute unit/integration tests if present. `--ci` makes results deterministic.
<code>- name: Build (smoke) run:   if [ -f package.json ] &amp;&amp; cat package.json   jq -e '.scripts.build' &gt; /dev/null; then npm run build else echo "No build script; skipping." fi</code>	Attempt a production build. If it compiles, your bundle is likely in good shape.

## Common Tweaks

- To add Playwright/Cypress: insert a step after build to run E2E tests.
- To cache node modules: add `actions/cache` keyed by `package-lock.json`.
- To speed up: split DB checks and app checks into two parallel jobs.

## Branch Protection Tip

Once this workflow runs once on a PR, go to Settings → Branches → Branch protection rules, and require the “build-and-test” check to pass before merging to main.