Implementation Plan

[Overview]
Fix recurring CI failures (git checkout/ref errors, jest setup resolution, missing CLI invocations, and Node24 compatibility) by correcting GitHub Actions workflows, upgrading devDependencies for Node 24, and adding CI diagnostics and validation so the pipeline reliably passes and deployment becomes possible.

This plan scopes only changes required to stabilize CI and the quality gate: (1) correct malformed actions/checkout usage across workflows, (2) ensure Jest config and test setup files resolve in CI, (3) remove/replace any bare CLI invocations (e.g., `tenant-flow`) so steps run in CI, (4) upgrade devDependencies to be Node v24 compatible and fix any breaking changes, and (5) add diagnostic steps + stricter failure reporting. Changes are minimal, follow existing monorepo patterns, and include tests for CI helper scripts. The approach: (a) edit workflow YAMLs with exact replacements, (b) update package.json devDependencies and lockfile, (c) add small CI validation scripts and unit tests, (d) re-run CI and iterate.

[Types]
Describe CI-related type definitions and validation for any new TypeScript scripts.

- Single-sentence: Add a small set of TypeScript types for CI helper scripts and validation utilities.
- Definitions:
  - interface CiCheckoutConfig
    - fetchDepth: number (>=0) — validation: integer, default 0
    - fetchTags: boolean — default true
    - submodules: boolean — default true
    - ref: string — must be non-empty
  - interface WorkflowDiagnosticsResult
    - ok: boolean
    - nodeVersion: string
    - gitRef: string | null
    - hasTenantFlowCli: boolean
    - err?: string
  - enum NodeStrategy
    - PIN_18 = 'pin-18'
    - UPGRADE_24 = 'upgrade-24'
  - validation rules:
    - CiCheckoutConfig.ref must match /^([0-9a-f]{7,40}|refs/heads/.+|.+)$/ or be resolved from github.head_ref/github.sha.
    - WorkflowDiagnosticsResult.ok === true only if hasTenantFlowCli === true and gitRef !== null.

[Files]
Single-sentence describing file modifications.

- New files to create:
  - scripts/ci/verify-workflows.ts — small TS utility that validates workflow checkout blocks and confirms no bare CLI invocations; exits non-zero on issues. Purpose: local/CI validation and unit

tests.
    - scripts/ci/print-env.sh — shell snippet used only in debug runs to print PATH, git remote, and node/npm versions.
    - scripts/ci/ci-types.ts — exported types/interfaces listed above (if using TS for the verify script).
    - tests/ci/verify-workflows.spec.ts — unit tests for the verify-workflows.ts utility.
- Existing files to modify (exact paths and changes):
    - .github/workflows/ci.yml
        - Replace all actions/checkout@v4 steps that contain malformed conditional lines in their with: blocks with the canonical block (see exact replacement below).
        - Add a diagnostic step (calls scripts/ci/print-env.sh) early in failing jobs (only in debug runs or gated by a boolean input).
        - Ensure actions/setup-node@v4 uses node-version: "24" (kept) and cache settings updated if necessary.
    - .github/workflows/deploy.yml
        - Same checkout replacement and diagnostic step insertion.
    - .github/workflows/fast-feedback.yml (or any other workflow under .github/workflows/)
        - Same checkout replacement and diagnostics; scan all workflows and update consistently.
    - apps/backend/jest.config.js
        - Ensure setupFilesAfterEnv uses path.resolve(__dirname, 'test', 'setup.ts') or an equivalent path that resolves reliably in CI; confirm file exists at apps/backend/test/setup.ts.
    - package.json (root) and apps/backend/package.json
        - Update devDependencies: bump eslint, jest, ts-jest, @types/jest, typescript, @types/node, rimraf replacement if used, and other deprecated packages indicated by npm warnings (list in Dependencies section). Add a script:
            - "ci:verify-workflows": "ts-node scripts/ci/verify-workflows.ts" (or node built script).
- Files to delete or move:
    - None required. Avoid deleting existing workflow files; update in place.
- Configuration updates:
    - tsconfig.json (if adding scripts in TS): add "scripts/ci/tsconfig.json" or ensure scripts compile under existing project tsconfig; prefer to run verify-workflows with ts-node via devDependency.
    - .github/workflows: ensure that checkout 'ref' is explicitly set where needed (see snippet).

[Functions]
Single-sentence describing function modifications.

- New functions/scripts:
    - verifyWorkflows(config?: {path?: string}): Promise — scripts/ci/verify-workflows.ts — scans .github/workflows/*.yml, validates checkout blocks, searches for bare CLI invocations (regex /\btenant-flow\b/), and exits non-zero if issues found.
    - printEnv(): shell script scripts/ci/print-env.sh — prints PATH, node -v, npm -v, git remote -v, git fetch --all --prune, git show -s --format=%H, and lists available executables (which tenant-flow || echo none).
- Modified functions:

    - No runtime code functions in app sources are required to be modified for CI stabilization; only config and workflow script changes.
- Removed functions:
    - None.

[Classes]
Single-sentence describing class modifications.

- New classes:
    - CIValidator (scripts/ci/verify-workflows.ts)
        - key methods:
            - loadWorkflows(): Promise<string[]> — reads YAML workflow files.
            - validateCheckoutBlocks(workflowYaml: string): string[] — returns array of problems.
            - findBareCliInvocations(workflowYaml: string): string[] — matches suspicious bare commands.
            - runDiagnostics(): Promise
        - No inheritance; simple utility class used by the verify script and tests.
- Modified classes:
    - None in application code.
- Removed classes:
    - None.

[Dependencies]
Single-sentence describing dependency modifications.

- Add or upgrade devDependencies to Node 24 compatible versions and remove deprecated packages.

Detailed changes:

- New/Upgraded devDependencies (root and workspace packages as appropriate):
    - "typescript": "^5.x" (align with repo usage; ensure compatibility with Node 24)
    - "ts-node": "^10.x" or latest compatible with TS version for running TS scripts in CI
    - "jest": latest 29+ compatible with ts-jest
    - "ts-jest": latest compatible with jest & TS
    - "@types/jest": latest matching jest
    - "@types/node": "^20.x" (compatible with Node 24)
    - "eslint": upgrade to latest supported major (replace deprecated 8.57.1)
    - Replace "rimraf" v3 with "rimraf" v4 or use "del-cli" depending on scripts
    - Remove or replace deprecated packages flagged by yarn/npm audit: inflight, glob v7 → update to v9-compatible packages in deps that depend on it (transitive upgrade via npm update/resolution).
- Integration requirements:
    - After package bumps, run `npm run typecheck` and `npm run test:unit`; fix any type errors due to upgraded types.

[Testing]
Single-sentence describing testing approach.

- Add unit tests for new CI verifier scripts; run full monorepo claude:check after upgrades and ensure CI passes; add a CI debug job that uploads workflow validation logs on failure.

Details:

- tests/ci/verify-workflows.spec.ts
  - Test cases:
    - Valid checkout block passes.
    - Malformed conditional lines or missing ref yields specific errors.
    - Bare 'tenant-flow' command detection test.
  - Use vitest or jest depending on repo standard (use existing runner for consistency).
- Workflow-level testing:
  - Add a workflow dispatch job "ci-sanity" that runs only the verify-workflows script to validate workflows before merging changes.
  - Ensure `npm run claude:check` still runs and passes; update scripts where necessary.
- Validation steps in pipeline:
  - Add an early "workflow-lint" job that runs scripts/ci/verify-workflows.ts and fails fast with a clear message.
  - Upload diagnostic artifact (logs) when jobs fail for easier triage.

[Implementation Order]
Single-sentence describing the implementation sequence.

- Apply changes in small, reversible commits: update workflows, add verify scripts and tests, bump devDependencies, run local checks, then open PR and iterate until CI is green.

Ordered steps:

1. Create scripts/ci/verify-workflows.ts and scripts/ci/print-env.sh plus tests/ci/verify-workflows.spec.ts; commit to a feature branch.
2. Apply canonical checkout replacement to all .github/workflows/*.yml files (ci.yml, deploy.yml, fast-feedback.yml). See exact replacement below.
3. Add a small "workflow-lint" job to CI that runs the verify-workflows script and fails fast on issues.
4. Run `npm ci` locally (or `npm install`) and run `npm run typecheck` and unit tests; fix any issues.
5. Upgrade devDependencies in package.json (root and workspaces), update lockfile (npm install), and adjust configs as required by upgrades.
6. Ensure apps/backend/jest.config.js uses path.resolve to test/setup.ts and that file exists; if missing, create apps/backend/test/setup.ts or adjust path.
7. Remove or replace any bare CLI calls in scripts/workflows with npm scripts or npx invocations; add a PATH diagnostic step if uncertain.
8. Open PR and run CI; if failing, use uploaded diagnostics to iterate.
9. Once CI passes and quality gate succeeds, merge and monitor first deployment run.

Exact checkout replacement (apply verbatim in each actions/checkout@v4 with: block):
uses: actions/checkout@v4
with:
fetch-depth: 0
fetch-tags: true

submodules: true
ref: ${{ github.head_ref || github.sha }}

Recommended small diagnostic step (YAML snippet to add near the top of jobs for debugging):

- name: Print CI environment (debug only)
  run: |
  bash .github/scripts/print-env.sh
  if: ${{ env.CI_DEBUG == 'true' }}

Notes:

- Do not insert GH conditional expressions as anonymous mapping entries inside `with:`. Use explicit `ref:` keys.
- Keep Node Version = 24 per your choice; plan to upgrade devDependencies listed in Dependencies.