To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Migrating from Jest to Vitest

Class Duration

7 hours of live training delivered over 1 day

Student Prerequisites

- Professional JavaScript or TypeScript development experience
- Experience writing and maintaining tests with Jest
- Familiarity with npm-based tooling and CI pipelines

Target Audience

Developers, tech leads, and platform engineers who maintain Jest test suites and plan to move them to Vitest. It fits front-end teams on Vite, Node.js and TypeScript teams with ESM transform problems, and monorepo owners migrating package by package. For more on AI-assisted workflows, see AI-Assisted Development with JavaScript and TypeScript.

Description

This one-day course moves an existing Jest suite to **Vitest** without a long freeze or lost coverage. Students map Jest configuration to Vitest and convert the APIs that the official migration guide calls out. These include `vi` mocks, module mock factories, `vi.importActual`, mock reset behavior, hooks, and fake timers. The course also covers snapshots, ESM and TypeScript tests, DOM environments, coverage, and CI. The last part covers migration at scale with codemods, AI coding agents, and Jest and Vitest running side by side.

Learning Outcomes

- Decide whether a Jest suite should move to Vitest.
- Translate Jest configuration to Vitest configuration.
- Convert Jest mocks, spies, hooks, and timers to Vitest APIs.
- Migrate snapshot tests without churning snapshot files.
- Run ES module and TypeScript tests without a transform pipeline.
- Choose between jsdom, happy-dom, and Browser Mode.
- Match Jest coverage and CI behavior with V8 coverage and sharding.

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

- Automate conversion with codemods and AI coding agents.
- Migrate a monorepo incrementally with Vitest projects.

Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. No local student setup is required, this course is delivered with GitHub Codespaces. The free allotment of Codespaces hours from GitHub is sufficient to complete the course.

Versions covered: Vitest 5 (with Vite 6.4 or later) on Node.js 22.12 or later, migrating from Jest 29 or 30.

Training Topics

Why Teams Move to Vitest

- Native ESM and TypeScript
- Sharing Vite config with the app
- Watch mode and smart re-runs
- Parallel execution
- What carries over unchanged
- When to stay on Jest

Planning the Migration

- Inventorying a Jest setup
- Custom matchers and shared mocks
- Big-bang vs. incremental migration
- Success criteria and CI time

Configuration Mapping

- `jest.config` to `vitest.config.ts`
- The `test` block in `vite.config`
- `testEnvironment` to `environment`
- `setupFilesAfterEnv` to `setupFiles`

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

- `moduleNameMapper` to `resolve.alias`
- Transforms to Vite plugins
- `testMatch` to `include` and `exclude`
- `globals: true` vs. explicit imports
- `vitest/globals` types
- `JEST_WORKER_ID` to `VITEST_POOL_ID`

API Differences

- `vi.fn`, `vi.spyOn`, and `vi.mocked`
- Module mocking with `vi.mock`
- Hoisting and `vi.hoisted`
- Mock factories that return every export
- `jest.requireActual` to `vi.importActual`
- The `importOriginal` helper
- When `__mocks__` folders apply
- `mockReset` semantics
- Clearing mocks by default
- Converting `done` callbacks to promises
- Hooks that return teardown functions
- Hook ordering
- `jest.setTimeout` to `vi.setConfig`
- Test-name filtering with `-t`

Timers and Dates

- `vi.useFakeTimers` and `vi.advanceTimersByTime`
- Async timer methods
- No legacy fake timers
- Faking `Date` and `Temporal`
- `vi.setSystemTime`

Snapshots

- `toMatchSnapshot` and inline snapshots
- File snapshots
- Snapshot format differences
- Regenerating vs. preserving snapshots
- Reviewing snapshot diffs
- Custom serializers

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

ES Modules and TypeScript

- Tests without Babel or ts-jest
- Path aliases and `tsconfig`
- Type tests with `expectTypeOf`
- The `--typecheck` flag
- Mixed CommonJS code

Test Environments

- `node`, `jsdom`, and `happy-dom`
- Per-file `@vitest-environment` comments
- Browser Mode with Playwright
- Real browser vs. simulated DOM
- Testing Library under Vitest

Coverage

- V8 and Istanbul coverage
- `coverage.include` for untested files
- Coverage thresholds
- Comparing with Jest reports
- Reporters for CI and code review

CI

- JUnit and GitHub Actions reporters
- Sharding with `--shard`
- Blob reports and `--merge-reports`
- Affected tests with `--changed`
- Forks and threads pools
- Isolation and run-time tuning

Codemods

- Community Jest-to-Vitest codemods
- What codemods miss
- Dry runs and diff review
- Finishing the manual cases

Migrating with AI Coding Agents

- Migration instructions for agents
- File-by-file conversion

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

- The test runner as verifier
- Small, reviewable batches
- Catching invented APIs
- Catching skipped tests
- Catching weakened assertions
- Codemods, agents, and human review

Incremental Migration in Monorepos

- Jest and Vitest side by side
- Vitest `projects` per package
- Sharing configuration across packages
- Task runners and CI during transition
- Retiring Jest

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved