

# CI/CD Pipeline

## Drop CI/CD Pipeline

**Last updated:** 2026-02-13 **Source:** `src/drop-app/package.json`, `Dockerfile`, `fly.toml`, `vitest.config.ts`, `playwright.config.ts`

---

## Current State

Drop is in **MVP/pre-production** stage. Core CI/CD infrastructure exists including a GitHub Actions workflow.

### What exists:

- GitHub Actions CI workflow ( `.github/workflows/ci.yml` ) with 5 jobs: lint-and-typecheck, test, build, e2e, docker-build
- Dockerfile with multi-stage build ( `Dockerfile:1-63` )
- docker-compose for local and production ( `docker-compose.yml` , `docker-compose.production.yml` )
- Fly.io deployment config ( `fly.toml` )
- Vitest unit/integration test framework ( `vitest.config.ts` )
- Playwright E2E test framework ( `playwright.config.ts` )
- Health check endpoint ( `/api/health` )
- QA report generation via `scripts/qa-report.js` (automated in CI)

### What does not exist yet:

- Automated deployment pipeline (CI builds but does not deploy)
  - Container registry integration
  - Automated security scanning (npm audit, Snyk)
  - Test coverage reporting
  - Staging environment (Fly.io config exists but not deployed)
- 

## Build Pipeline

## Step 1: Install Dependencies

```
npm ci
```

Installs exact versions from `package-lock.json`.

## Step 2: Lint

```
npm run lint # eslint
```

## Step 3: Type Check

```
npx tsc --noEmit
```

## Step 4: Unit + Integration Tests

```
npm test # vitest run
```

Runs all tests in `tests/**/*.test.ts` (from `vitest.config.ts:7`). Test setup: `tests/setup.ts` sets `NODE_ENV=test`.

## Step 5: Build

```
npm run build # next build
```

Produces standalone output for Docker deployment.

## Step 6: Docker Build

```
docker build -t drop-app .
```

Multi-stage build: `deps -> builder -> runner`.

## Step 7: E2E Tests (requires running server)

```
npx playwright test
```

Requires dev server on `http://localhost:3000`. Playwright auto-starts it via `webServer` config.

---

# Test Framework Configuration

## Vitest (Unit + Integration)

**Config:** `src/drop-app/vitest.config.ts:1-15`

Setting	Value
Environment	<code>node</code>
Include	<code>tests/**/*.test.ts</code>
Setup	<code>tests/setup.ts</code>
Path alias	<code>@ -&gt; ./src</code>

## Playwright (E2E)

**Config:** `src/drop-app/playwright.config.ts:1-39`

Setting	Value
Test dir	<code>./tests/e2e</code>
Parallel	<code>false</code> (serial -- rate limiter is shared)
Workers	1
Retries (CI)	2
Timeout	30,000ms
Base URL	<code>http://localhost:3000</code>
Reporter	HTML
Trace	<code>on-first-retry</code>

### Test projects:

1. `user-flows` -- Basic user journey tests (`user-flows.spec.ts`)
2. `full-flows` -- Complete feature journeys (`full-flows.spec.ts`)
3. `input-chaos` -- Malicious/edge-case input testing (`input-chaos.spec.ts`). Depends on `user-flows`.

**Web server config:** Auto-starts `npm run dev` for E2E tests. Reuses existing server if running. 30s timeout.

---

# Deployment Targets

## Fly.io (Staging)

**Config:** `fly.toml:1-28`

```
# Deploy to Fly.io staging
fly deploy

# Set secrets
fly secrets set JWT_SECRET="your-secret"
fly secrets set NEXT_PUBLIC_SERVICE_MODE="mock"
```

**Region:** `arn` (Stockholm) **Auto-scaling:** Scales to 0 when idle, auto-starts on request.

## Docker (Self-hosted)

```
# Local dev (PostgreSQL 16 via Docker)
docker compose up -d

# Apply schema
make db-push
```

---

# Existing GitHub Actions CI Workflow

**File:** `.github/workflows/ci.yml`

Triggers on push/PR to `main` or `master`:

Jobs:

1. lint-and-typecheck – npm ci, npm run lint, tsc --noEmit
2. test – npm ci, npm test --if-present (depends on lint-and-typecheck)
3. build – npm ci, npm run build with JWT\_SECRET placeholder (depends on lint-and-typecheck)
4. e2e – npm ci, npx playwright install chromium, npm run build, npm run start (production mode), npx playwright test user-flows + full-flows, generate QA report, upload artifacts (depends on build)
5. docker-build – docker build -t drop-app:ci (depends on test + build + e2e)

### Artifacts uploaded:

- playwright-report/ — Playwright HTML report (7 day retention)
- qa-report.html — QA metrics report (pass/fail, execution time)

### Not yet implemented:

- Security scan (npm audit, Snyk)
- Deploy to staging (fly deploy)
- Deploy to production (manual approval gate)

**Status:** Full CI pipeline including E2E tests in place. CD deployment tracked in security hardening checklist ([security/hardening-checklist.md:120-126](#)).

---

Revision #7

Created 2026-02-18 08:44:28 UTC by John

Updated 2026-05-23 10:58:18 UTC by John