

CI/CD Pipeline Guide

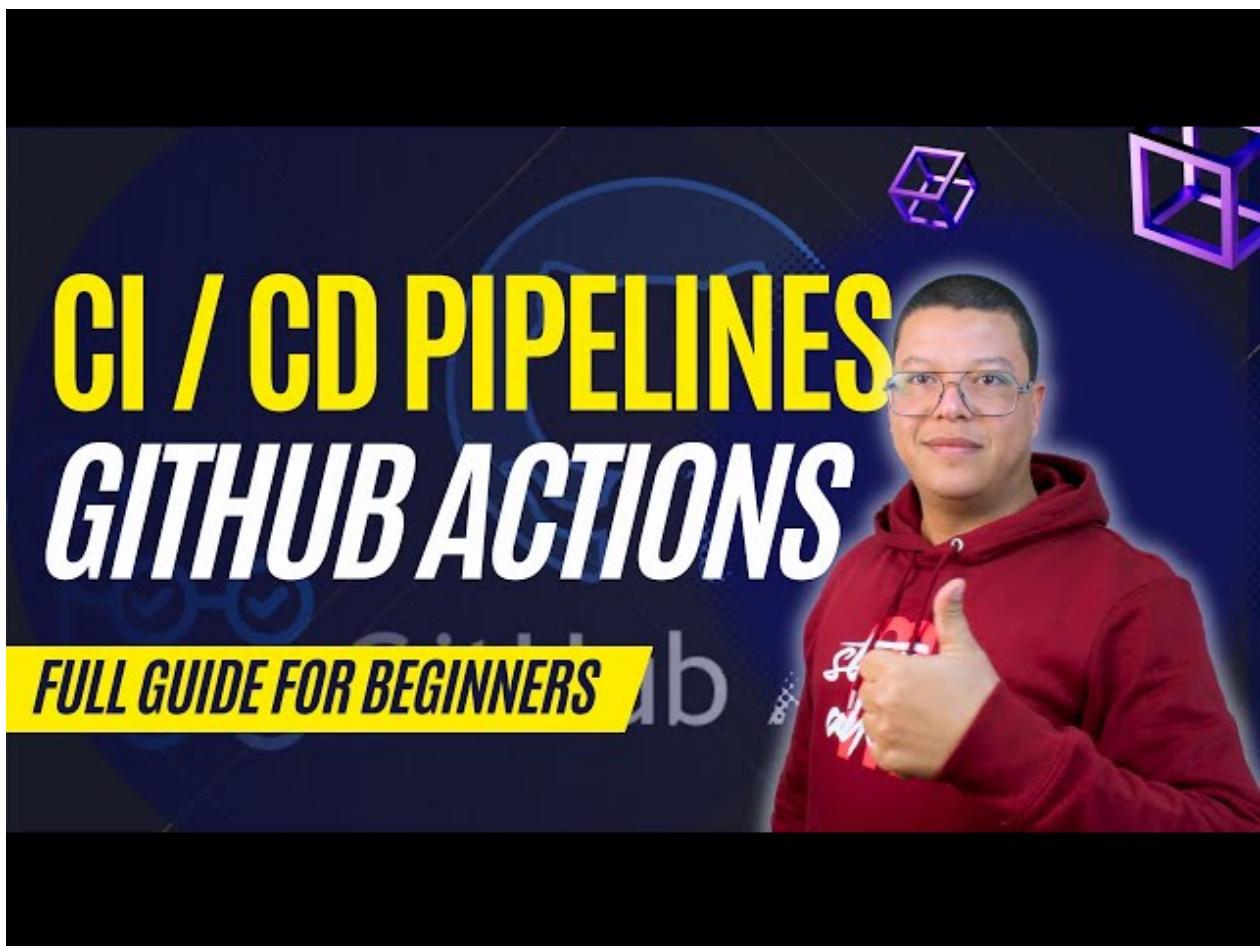
Overview

This project uses GitHub Actions for continuous integration and deployment. The pipeline automatically runs on every push to `master` or `main` branches and on all pull requests.

Workflow Status

Check the current status of the CI/CD pipeline:

- **Badge:**



- **Actions Page:** <https://github.com/salimemp/pro-pdf/actions>

Pipeline Stages

1. Lint and Type Check

Purpose: Ensure code quality and type safety

Steps:

- Checkout code

- Setup Node.js 18
- Install dependencies
- Run TypeScript type checking (`yarn tsc --noEmit`)
- Run ESLint for code quality

Status:  Must pass (ESLint failures are warnings only)

2. Build Application

Purpose: Verify the application builds successfully

Steps:

- Checkout code
- Setup Node.js 18
- Install dependencies
- Generate Prisma Client
- Build Next.js application
- Upload build artifacts

Artifacts Produced:

- `.build/` - Production build
- `.next/` - Next.js cache

Retention: 7 days

Status:  Must pass

3. E2E Tests

Purpose: Run end-to-end tests with Playwright

Steps:

- Setup PostgreSQL test database
- Install Playwright browsers (Chromium)
- Run database migrations and seed
- Execute E2E tests
- Upload test results and screenshots

Test Database:

- PostgreSQL 15
- Ephemeral (destroyed after tests)
- Seeded with test data

Artifacts Produced:

- Playwright HTML report
- Test screenshots (on failure)

Retention: 7 days

Status:  Must pass

4. Security Scan

Purpose: Check for security vulnerabilities

Steps:

- Run `yarn audit` for known vulnerabilities
- Check for outdated dependencies

Status:  Warnings only (continues on error)

5. Build Status Summary

Purpose: Aggregate all pipeline results

Steps:

- Display status of all jobs
- Provide summary of CI/CD run

Status:  Always runs

Environment Variables

The following environment variables are used in the CI/CD pipeline:

```
# Build Configuration
NODE_OPTIONS: '--max-old-space-size=4096'
__NEXT_TEST_MODE: '1'
NEXT_DIST_DIR: '.build'

# Test Database
DATABASE_URL: postgres://testuser:testpassword@localhost:5432/testdb

# Authentication
NEXTAUTH_SECRET: test_secret_key_for_github_actions
NEXTAUTH_URL: http://localhost:3000
```

Running the Pipeline

Automatic Triggers

The pipeline runs automatically on:

1. Push to master/main:

```
bash
git push origin master
```

2. Pull Request to master/main:

```
bash
git checkout -b feature/my-feature
git commit -am "Add new feature"
git push origin feature/my-feature
# Create PR on GitHub
```

Manual Trigger

You can also manually trigger the workflow from GitHub:

1. Go to <https://github.com/salimemp/pro-pdf/actions>
2. Select “CI/CD Pipeline”

3. Click “Run workflow”
4. Choose the branch
5. Click “Run workflow”

Viewing Results

GitHub Actions UI

1. Navigate to <https://github.com/salimemp/pro-pdf/actions>
2. Click on the latest workflow run
3. View each job’s logs and status

Download Artifacts

1. Open a completed workflow run
2. Scroll to “Artifacts” section
3. Download:
 - build-artifacts - Production build files
 - playwright-report - E2E test results
 - test-screenshots - Failed test screenshots (if any)

Local Testing

Run the same checks locally:

```
# Type checking
yarn tsc --noEmit

# Linting
yarn lint

# Build
yarn build

# E2E tests
yarn playwright test
```

Troubleshooting

Failed Type Check

Issue: TypeScript compilation errors

Solution:

```
yarn tsc --noEmit
# Fix the reported errors
```

Failed Build

Issue: Next.js build fails

Solution:

```
yarn build
# Check the error logs
# Ensure all dependencies are installed
yarn install
```

Failed E2E Tests

Issue: Playwright tests fail

Solution:

```
# Run tests locally
yarn playwright test

# View test report
yarn playwright show-report

# Debug specific test
yarn playwright test e2e/auth.spec.ts --debug
```

Common Causes:

- Database not seeded properly
- Element selectors changed
- Timing issues (add more waitFor statements)

Security Scan Warnings

Issue: Vulnerabilities detected

Solution:

```
# View vulnerabilities
yarn audit

# Update dependencies
yarn upgrade-interactive

# Fix specific vulnerabilities
yarn audit fix
```

Best Practices

Before Pushing Code

1. Run tests locally:

```
bash
yarn tsc --noEmit
yarn lint
yarn build
yarn playwright test
```

2. Commit frequently with meaningful messages

3. Create feature branches for new features:

```
bash
git checkout -b feature/new-feature
```

4. Open Pull Requests for code review

Writing Tests

1. Add tests for new features
2. Update existing tests when changing functionality
3. Use descriptive test names
4. Add comments for complex test logic

Monitoring

1. Check CI/CD badge on README.md
2. Review failed builds immediately
3. Fix breaking changes before merging
4. Monitor security alerts

Pipeline Configuration

Modifying the Workflow

Edit `.github/workflows/ci-cd.yml` to:

1. Add new jobs:

```
```yaml
my-custom-job:
 name: My Custom Job
 runs-on: ubuntu-latest
 steps:
 - name: Run custom command
 run: echo "Hello World"
 ...
```

#### 2. Change Node.js version:

```
yaml
- name: Setup Node.js
 uses: actions/setup-node@v4
 with:
 node-version: '20' # Change to Node 20
```

#### 3. Add environment variables:

```
yaml
- name: My Step
 env:
 MY_VAR: 'my-value'
 run: echo $MY_VAR
```

## GitHub Secrets

For sensitive data, use GitHub Secrets:

1. Go to repository **Settings → Secrets and variables → Actions**
2. Click **New repository secret**
3. Add name and value
4. Reference in workflow:

```
yaml
env:
 MY_SECRET: ${{ secrets.MY_SECRET }}
```

## Resources

---

- [GitHub Actions Documentation](https://docs.github.com/en/actions) (<https://docs.github.com/en/actions>)
- [Next.js CI/CD Guide](https://nextjs.org/docs/pages/building-your-application/deploying/ci-build-caching) (<https://nextjs.org/docs/pages/building-your-application/deploying/ci-build-caching>)
- [Playwright CI Guide](https://playwright.dev/docs/ci) (<https://playwright.dev/docs/ci>)
- [GitHub Actions Marketplace](https://github.com/marketplace?type=actions) (<https://github.com/marketplace?type=actions>)

## Support

---

For issues with the CI/CD pipeline:

1. Check the [Actions logs](https://github.com/salimemp/pro-pdf/actions) (<https://github.com/salimemp/pro-pdf/actions>)
  2. Review this guide
  3. Check GitHub Actions [status page](https://www.githubstatus.com/) (<https://www.githubstatus.com/>)
  4. Open an issue in the repository
- 

**Last Updated:** December 2025

**Pipeline Version:** 1.0.0