

Issue #5: GitHub Actions CI/CD workflow

Labels: infrastructure, ci-cd, automation

Milestone: Production Ready

Estimated Effort: 4-6 hours

Description

Implement a comprehensive CI/CD pipeline using GitHub Actions that automatically tests, builds, and deploys pp-bot to Railway.com on every push to the main branch.

Goals

- Automated testing on every PR and push
- Automated deployment to Railway.com on main branch
- Code quality checks (linting, type checking)
- Test coverage reporting
- Fast feedback loop (<5 minutes)

Tasks

Phase 1: CI Pipeline (2-3 hours)

- [] Create `.github/workflows/ci.yml` for testing
- [] Configure test job with multiple Node versions
- [] Add lint job for code quality
- [] Add type-check job (if TypeScript)
- [] Add test coverage reporting
- [] Configure branch protection rules

Phase 2: CD Pipeline (2-3 hours)

- [] Create `.github/workflows/deploy.yml` for deployment
- [] Configure Railway deployment on main branch
- [] Add deployment environment protection
- [] Set up Railway API token as GitHub secret
- [] Add deployment status notifications
- [] Test deployment workflow

Phase 3: Optimizations (1-2 hours)

- [] Add caching for node_modules
- [] Add caching for build artifacts
- [] Optimize workflow run time
- [] Add workflow status badges to README
- [] Document workflow in README

Workflow Files

CI Workflow: `.github/workflows/ci.yml`

```

name: CI

on:
  push:
    branches: [ main, develop ]
  pull_request:
    branches: [ main, develop ]

jobs:
  test:
    name: Test (Node ${{ matrix.node-version }})
    runs-on: ubuntu-latest

    strategy:
      matrix:
        node-version: [18.x, 20.x]

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Setup Node.js
        uses: actions/setup-node@v4
        with:
          node-version: ${{ matrix.node-version }}
          cache: 'npm'

      - name: Install dependencies
        run: npm ci

      - name: Run tests
        run: npm test -- --coverage

      - name: Upload coverage to Codecov
        uses: codecov/codecov-action@v4
        with:
          file: ./coverage/coverage-final.json
          flags: unittests
          name: codecov-umbrella
          fail_ci_if_error: false

  lint:
    name: Lint
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Setup Node.js
        uses: actions/setup-node@v4
        with:
          node-version: '20.x'
          cache: 'npm'

      - name: Install dependencies
        run: npm ci

      - name: Run ESLint
        run: npm run lint
        continue-on-error: true

```

```
type-check:
  name: Type Check
  runs-on: ubuntu-latest
  if: contains(github.event.head_commit.message, 'typescript') || true

steps:
  - name: Checkout code
    uses: actions/checkout@v4

  - name: Setup Node.js
    uses: actions/setup-node@v4
    with:
      node-version: '20.x'
      cache: 'npm'

  - name: Install dependencies
    run: npm ci

  - name: Type check
    run: npm run type-check
    continue-on-error: false

build:
  name: Build
  runs-on: ubuntu-latest
  needs: [test, lint]

steps:
  - name: Checkout code
    uses: actions/checkout@v4

  - name: Setup Node.js
    uses: actions/setup-node@v4
    with:
      node-version: '20.x'
      cache: 'npm'

  - name: Install dependencies
    run: npm ci

  - name: Build application
    run: npm run build

  - name: Upload build artifacts
    uses: actions/upload-artifact@v4
    with:
      name: dist
      path: dist/
      retention-days: 7
```

Deployment Workflow: `.github/workflows/deploy.yml`

```

name: Deploy to Railway

on:
  push:
    branches: [ main ]
  workflow_dispatch:

jobs:
  deploy:
    name: Deploy to Railway.com
    runs-on: ubuntu-latest
    environment:
      name: production
      url: https://pp-bot-production.up.railway.app

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Setup Node.js
        uses: actions/setup-node@v4
        with:
          node-version: '20.x'
          cache: 'npm'

      - name: Install dependencies
        run: npm ci

      - name: Run tests
        run: npm test

      - name: Build application
        run: npm run build

      - name: Install Railway CLI
        run: npm install -g @railway/cli

      - name: Deploy to Railway
        env:
          RAILWAY_TOKEN: ${{ secrets.RAILWAY_TOKEN }}
        run: railway up --detach

      - name: Wait for deployment
        run: sleep 30

      - name: Verify deployment
        run: |
          curl -f https://pp-bot-production.up.railway.app/health || exit 1

      - name: Notify deployment success
        if: success()
        uses: actions/github-script@v7
        with:
          script: |
            github.rest.issues.createComment({
              issue_number: context.issue.number,
              owner: context.repo.owner,
              repo: context.repo.repo,
              body: '✅ Successfully deployed to Railway.com!'
            })

      - name: Notify deployment failure

```

```
if: failure()
uses: actions/github-script@v7
with:
  script: |
    github.rest.issues.createComment({
      issue_number: context.issue.number,
      owner: context.repo.owner,
      repo: context.repo.repo,
      body: '✖ Deployment to Railway.com failed. Check the logs for details.'
})
```

Combined Workflow (Alternative): `.github/workflows/main.yml`

```

name: CI/CD Pipeline

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  test-and-build:
    name: Test and Build
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v4

      - name: Setup Node.js
        uses: actions/setup-node@v4
        with:
          node-version: '20.x'
          cache: 'npm'

      - name: Install dependencies
        run: npm ci

      - name: Run lint
        run: npm run lint
        continue-on-error: true

      - name: Run tests
        run: npm test -- --coverage

      - name: Type check
        run: npm run type-check
        continue-on-error: true

      - name: Build
        run: npm run build

      - name: Upload build artifacts
        if: github.ref == 'refs/heads/main'
        uses: actions/upload-artifact@v4
        with:
          name: dist
          path: dist/

deploy:
  name: Deploy to Railway
  needs: test-and-build
  if: github.ref == 'refs/heads/main' && github.event_name == 'push'
  runs-on: ubuntu-latest
  environment: production

  steps:
    - uses: actions/checkout@v4

    - name: Download build artifacts
      uses: actions/download-artifact@v4
      with:
        name: dist
        path: dist/

```

```

- name: Install Railway CLI
  run: npm install -g @railway/cli

- name: Deploy to Railway
  env:
    RAILWAY_TOKEN: ${{ secrets.RAILWAY_TOKEN }}
  run: railway up --detach

- name: Verify deployment
  run: |
    sleep 30
    curl -f https://pp-bot-production.up.railway.app/health || exit 1
  
```

GitHub Secrets Configuration

Required Secrets

Navigate to: Settings → Secrets and variables → Actions → New repository secret

1. RAILWAY_TOKEN

- Get from Railway.com dashboard
- Navigate to: Account Settings → Tokens → Create Token
- Copy token and add to GitHub secrets

Optional Secrets (for enhanced features)

- CODECOV_TOKEN - For code coverage reporting
- SLACK_WEBHOOK_URL - For deployment notifications to Slack
- SENTRY_AUTH_TOKEN - For error tracking setup

Branch Protection Rules

Configure Protection for main Branch

Settings → Branches → Add branch protection rule

- [x] Require pull request reviews before merging
- [x] Require status checks to pass before merging
- Select: test, lint, build
- [x] Require branches to be up to date before merging
- [x] Require conversation resolution before merging
- [] Require signed commits (optional)
- [x] Include administrators

Workflow Optimization

Caching Strategy

```
# Cache node_modules
- name: Cache node modules
  uses: actions/cache@v4
  with:
    path: ~/.npm
    key: ${{ runner.os }}-node-${{ hashFiles('**/package-lock.json') }}
    restore-keys: |
      ${{ runner.os }}-node-
      ${{ runner.os }}-node-*

# Cache TypeScript build
- name: Cache TypeScript build
  uses: actions/cache@v4
  with:
    path: |
      dist/
      .tsbuildinfo
    key: ${{ runner.os }}-ts-build-${{ hashFiles('src/**/*.ts') }}
    restore-keys: |
      ${{ runner.os }}-ts-build-
```

Workflow Speed Optimizations

1. Run jobs in parallel when possible
2. Use `npm ci` instead of `npm install`
3. Cache dependencies and build artifacts
4. Skip unnecessary steps with conditionals
5. Use matrix builds only when needed

Status Badges

Add to README.md

```
# pp-bot

[
[
[
[
```

Notification Setup (Optional)

Slack Notifications

```
- name: Notify Slack
  if: always()
  uses: 8398a7/action-slack@v3
  with:
    status: ${{ job.status }}
    text: 'Deployment to Railway.com ${{ job.status }}'
    webhook_url: ${{ secrets.SLACK_WEBHOOK_URL }}
    fields: repo,message,commit,author,action,eventName,ref,workflow
```

Email Notifications

GitHub automatically sends email notifications for workflow failures if you have notifications enabled.

Testing the Workflow

Local Testing with `act`

```
# Install act (https://github.com/nektos/act)
brew install act # macOS
# or
curl https://raw.githubusercontent.com/nektos/act/master/install.sh | sudo bash

# Test workflows locally
act push
act pull_request
act -j test # Run specific job
```

Manual Workflow Trigger

Use `workflow_dispatch` to manually trigger workflows from GitHub UI:

- Navigate to Actions tab
- Select workflow
- Click “Run workflow”

Acceptance Criteria

- [] CI workflow runs on every PR and push
- [] All tests pass in CI environment
- [] Linting and type checking work correctly
- [] Deployment workflow triggers on push to main
- [] Deployment to Railway.com succeeds
- [] Health check verification passes after deployment
- [] Branch protection rules are configured
- [] GitHub secrets are properly set
- [] Workflow completes in <5 minutes
- [] Status badges added to README
- [] Documentation updated with workflow information

Testing Checklist

- [] Create test PR and verify CI runs
- [] Verify lint catches style issues
- [] Verify tests run and report coverage
- [] Push to main and verify deployment
- [] Verify Railway.com receives deployment
- [] Verify health check passes
- [] Test manual workflow dispatch
- [] Verify failure notifications work

Troubleshooting Guide

Common Issues

Issue: Deployment fails with authentication error

Solution: Verify RAILWAY_TOKEN secret `is` correct

- Generate `new` token from Railway.com
- Update GitHub secret
- Re-run workflow

Issue: Tests pass locally but fail in CI

Solution: Environment differences

- Check Node.js versions match
- Verify environment variables
- Check `for` timing/async issues
- Review CI logs `for` specific errors

Issue: Workflow takes too long (>10 minutes)

Solution: Optimize workflow

- Add caching `for` dependencies
- Run jobs `in` parallel
- Skip unnecessary steps
- Use smaller Docker images

Resources

- [GitHub Actions Documentation](https://docs.github.com/en/actions) (<https://docs.github.com/en/actions>)
- [Railway.com CI/CD Guide](https://docs.railway.app/deploy/integrations) (<https://docs.railway.app/deploy/integrations>)
- [GitHub Actions Marketplace](https://github.com/marketplace?type=actions) (<https://github.com/marketplace?type=actions>)
- [Workflow Syntax Reference](https://docs.github.com/en/actions/using-workflows/workflow-syntax-for-github-actions) (<https://docs.github.com/en/actions/using-workflows/workflow-syntax-for-github-actions>)

Dependencies

Requires:

- Issue #4: Railway.com deployment (deployment target must exist)

Recommended:

- Issue #3: TypeScript migration (for type-check job)
- Issue #10: Testing improvements (better test coverage)