

CI/CD npm Integration Guide

Project: QuantumLeap AI Website

Purpose: Ensure all CI/CD pipelines use npm (not Yarn)

Date: November 10, 2025

Current State Analysis

Deployment Platforms

Based on project documentation:

- **Primary:** Abacus.AI (Vercel-style deployment)
- **Domain:** quantumleapai.abacusai.app
- **Git Integration:** Pushes to GitHub trigger auto-deploy

CI/CD Configuration Status

 **No CI/CD configuration files found** in repository:

- No `.github/workflows/` directory
- No GitHub Actions files
- No `.circleci/config.yml`
- No `.travis.yml`
- No `vercel.json`
- No `netlify.toml`

Interpretation: Deployment is handled automatically by Abacus.AI platform based on git pushes.

Verification Checklist

1. Abacus.AI Platform Configuration

Action: Verify platform build settings use npm

Steps:

1. Log into Abacus.AI deployment dashboard
2. Navigate to QuantumLeap AI project settings
3. Check “Build Command” configuration
4. Ensure it shows: `npm run build` (NOT `yarn build`)
5. Check “Install Command” configuration
6. Ensure it shows: `npm install --legacy-peer-deps` (NOT `yarn install`)

Expected Configuration:

```
{
  "buildCommand": "npm run build",
  "installCommand": "npm install --legacy-peer-deps",
  "outputDirectory": ".next",
  "framework": "nextjs"
}
```

2. Vercel (if used as fallback)

Action: Check `vercel.json` for Yarn references

Current Status: No `vercel.json` file exists in repository

If deploying to Vercel manually, create:

```
{
  "buildCommand": "npm run build",
  "installCommand": "npm install --legacy-peer-deps",
  "framework": "nextjs",
  "ignoreCommand": "git diff --quiet HEAD^ HEAD ./"
}
```

3. GitHub Actions (Recommended for CI)

Action: Create GitHub Actions workflow for automated testing

Current Status: No `.github/workflows/` directory exists

Recommendation: Add CI workflow to test PRs before merge



Recommended GitHub Actions Workflow

Since no CI is currently configured, here's a recommended workflow:

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

```

name: CI - npm Build & Test

on:
  push:
    branches: [ main, dev, staging ]
  pull_request:
    branches: [ main, dev ]

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

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

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

      - name: Setup Node.js ${{ matrix.node-version }}
        uses: actions/setup-node@v4
        with:
          node-version: ${{ matrix.node-version }}
          cache: 'npm'
          cache-dependency-path: './nextjs_space/package-lock.json'

      - name: Verify no Yarn files
        run: |
          cd nextjs_space
          if [ -f "yarn.lock" ] || [ -f ".yarnrc.yml" ]; then
            echo "✗ ERROR: Yarn files detected!"
            exit 1
          fi
          echo "✓ No Yarn files found"

      - name: Install dependencies
        run: |
          cd nextjs_space
          npm install --legacy-peer-deps

      - name: Run linter
        run: |
          cd nextjs_space
          npm run lint

      - name: Build project
        run: |
          cd nextjs_space
          npm run build

      - name: Upload build artifacts
        uses: actions/upload-artifact@v4
        with:
          name: nextjs-build-${{ matrix.node-version }}
          path: nextjs_space/.next
          retention-days: 7

deploy:
  name: Deploy to Abacus.AI
  needs: build

```

```

runs-on: ubuntu-latest
if: github.ref == 'refs/heads/main' && github.event_name == 'push'

steps:
  - name: Trigger Abacus.AI deployment
    run:
      echo "✅ Deployment triggered by git push"
      echo "Platform will automatically run npm install and npm run build"

```

Blocking Yarn Usage in CI

Pre-commit Hook (Local Prevention)

Create `.git/hooks/pre-commit`:

```

#!/bin/bash
# Prevent committing Yarn files

echo "🔍 Checking for Yarn files..."

# Check for Yarn lock files
if git diff --cached --name-only | grep -q "yarn.lock"; then
  echo "❌ ERROR: Cannot commit yarn.lock"
  echo "This project uses npm. Please remove yarn.lock."
  exit 1
fi

# Check for Yarn config
if git diff --cached --name-only | grep -q ".yarnrc"; then
  echo "❌ ERROR: Cannot commit .yarnrc files"
  echo "This project uses npm."
  exit 1
fi

# Check for Yarn commands in package.json scripts
if git diff --cached --name-only | grep -q "package.json"; then
  if git diff --cached nextjs_space/package.json | grep -q "yarn "; then
    echo "❌ ERROR: Yarn commands detected in package.json"
    echo "Please use npm commands instead."
    exit 1
  fi
fi

echo "✅ No Yarn files detected"
exit 0

```

Installation:

```

cd /home/ubuntu/quantumleap_io
cat > .git/hooks/pre-commit << 'EOF'
[paste script above]
EOF
chmod +x .git/hooks/pre-commit

```

GitHub Actions Guard (Remote Prevention)

Add this job to `.github/workflows/yarn-guard.yml`:

```

name: Yarn Guard - Block Yarn Files

on:
  pull_request:
    types: [opened, synchronize, reopened]
  push:
    branches: [main, dev]

jobs:
  yarn-guard:
    name: Check for Yarn files
    runs-on: ubuntu-latest

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

      - name: Verify no Yarn files
        run: |
          echo "🔍 Scanning for Yarn files..."

          # Check for yarn.lock
          if [ -f "nextjs_space/yarn.lock" ]; then
            echo "❌ ERROR: yarn.lock detected!"
            echo "This project uses npm. Please remove yarn.lock."
            exit 1
          fi

          # Check for .yarnrc files
          if ls nextjs_space/.yarnrc* 1> /dev/null 2>&1; then
            echo "❌ ERROR: Yarn config files detected!"
            exit 1
          fi

          # Check package.json for yarn commands
          if grep -r "yarn" nextjs_space/package.json; then
            echo "❌ ERROR: Yarn commands found in package.json!"
            exit 1
          fi

          echo "✅ No Yarn files detected. Good to merge!"

      - name: Comment on PR
        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.name,
              body: '❌ **Yarn Guard Failed**\n\nThis project uses npm exclusively.
Please remove any Yarn files (`yarn.lock`, `yarnrc.yml`) and update scripts to use
npm commands.\n\nSee `YARN_TO_NPM_MIGRATION.md` for details.
})

```

Testing CI/CD Integration

Test 1: Manual Trigger (GitHub Actions)

```
# Create a test branch
git checkout -b test/ci-npm-validation

# Push to trigger GitHub Actions
git push origin test/ci-npm-validation

# Monitor Actions tab in GitHub
# ✓ Should use npm commands
# ✓ Should not mention Yarn
```

Test 2: Deployment Platform

```
# Make a trivial change
echo "# npm-tested" >> README.md
git add README.md
git commit -m "Test: Verify npm deployment"
git push origin main

# Monitor Abacus.AI deployment logs
# ✓ Should show "npm install"
# ✓ Should show "npm run build"
# ✘ Should NOT show "yarn install"
```

Test 3: PR Review (if GitHub Actions enabled)

```
# Create PR with Yarn file (should fail)
git checkout -b test/yarn-block
touch nextjs_space/yarn.lock
git add nextjs_space/yarn.lock
git commit -m "Test: Try to add yarn.lock"
git push origin test/yarn-block

# Create PR on GitHub
# ✓ Yarn guard should fail
# ✓ PR should be blocked
```



CI/CD Integration Checklist

- [] **Abacus.AI Platform**
- [] Build command uses `npm run build`
- [] Install command uses `npm install --legacy-peer-deps`
- [] No Yarn references in platform config
- [] Deployment logs show npm usage
- [] **GitHub Actions (Optional)**
- [] Created `.github/workflows/ci.yml`
- [] Workflow tests `npm install/build`

- [] Workflow blocks Yarn files
 - [] Matrix tests Node 18 and 20
 - [] **Pre-commit Hooks**
 - [] Created `.git/hooks/pre-commit`
 - [] Hook prevents Yarn file commits
 - [] Hook checks package.json scripts
 - [] Hook is executable
 - [] **Documentation**
 - [] CI/CD setup documented
 - [] npm-only policy communicated
 - [] Deployment process updated
 - [] **Team Training**
 - [] All developers aware of npm-only policy
 - [] CI/CD workflows tested
 - [] Rollback procedures documented
-



Troubleshooting CI/CD Issues

Issue: “command not found: yarn”

Cause: CI environment trying to use Yarn

Solution: Update CI config to use npm commands

Issue: Build fails with peer dependency errors

Cause: Missing `--legacy-peer-deps` flag

Solution: Add flag to install command in CI config

Issue: GitHub Actions shows cached Yarn

Cause: Action cache includes Yarn artifacts

Solution: Clear GitHub Actions cache:

```
- name: Clear cache
  run: |
    npm cache clean --force
    rm -rf ~/.npm
```

Issue: Deployment platform uses wrong command

Cause: Platform detected Yarn from old config

Solution: Delete old config, clear platform cache, redeploy



Next Steps

1. Review Abacus.AI Platform Settings

- Confirm build/install commands
- Update if using Yarn

2. Consider Adding GitHub Actions

- Provides CI testing before deployment
- Prevents Yarn files from being merged

3. Install Pre-commit Hook

- Prevents local commits of Yarn files
- First line of defense

4. Document CI/CD Process

- Update team wiki/docs
- Include in onboarding materials

5. Monitor First Few Deployments

- Watch for npm-related issues
 - Verify build times comparable to Yarn
-



Support

For CI/CD configuration issues:

1. Check platform deployment logs
 2. Review this document
 3. Consult with DevOps team
 4. Update documentation if new issues found
-

Created By: DeepAgent (Abacus.AI)

Last Updated: November 10, 2025

Related Docs: [YARN_TO_NPM_MIGRATION.md](#), [CROSS_PLATFORM_TEST_GUIDE.md](#)