

STUDIOTOHUB PIPELINE SPECIFICATION

STANDARDIZED CI/CD ARCHITECTURE FOR REACT & VITE APPLICATIONS

Publisher: Marie Landry Spy Shop

Author / ORCID: Marie-Soleil Seshat Landry (0009-0008-5027-3337)

Domain: marielandryspyshop.com

Verification State: Verified Open-Source Release v1.2

AI Disclosure: Generated via Gemini (Google). Upgraded for Node.js 24 runner lifecycles, actions version pins (@v4), and modern Vite/React bundling rules as of September 25, 2026.

Abstract

This technical specification outlines a standardized, production-ready continuous integration and continuous deployment (CI/CD) pipeline for transitioning Single-Page Applications (SPAs) generated via Google AI Studio into static hosting environments on GitHub Pages. It resolves critical structural deployment challenges in Vite and React applications, including nested repository asset path resolutions, headless GitHub Actions authentication, secure environment variable injection, and client-side routing fallback strategies. Updated to support clean runners using Node.js 24 LTS and legacy peer dependency fallback mechanics.

1. Scope & Objective

This specification standardizes the automated transition of web applications generated via AI Studio into live production environments on GitHub Pages. By enforcing Vite base path compliance, modern GitHub Actions runner defaults (Node.js 24), and legacy peer dependency resolution, development teams eliminate manual build failures and asset routing breakage.

2. Technical Implementation Architecture

Step 1: Configure Vite for Relative Asset Paths

Vite defaults to an absolute root path (/). When deploying to nested GitHub Pages project repositories (https://username.github.io/repo-name/), asset lookups fail unless the base property is adjusted.

- Action:** Update vite.config.ts to set base: './' and define the production output directory.

```
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';

export default defineConfig({
  plugins: [react()],
  base: './', // CRITICAL for GitHub Pages sub-path hosting
  build: {
    outDir: 'dist',
  }
});
```

2. Technical Implementation Architecture (Cont.)

Step 2: Establish the GitHub Actions Deployment Workflow

- **Runtime Upgrade:** GitHub Actions runners run on active **Node.js 24** LTS environments. The workflow template explicitly binds `actions/setup-node@v4` to target runtime stability.
- **Triggers & Permissions:** Configured to activate on pushes to main or master, granting explicit repository permissions (contents: read, pages: write, id-token: write).
- **Robust Dependency Installation:** Uses `npm install --legacy-peer-deps` to gracefully bypass modern React ecosystem package collision errors.

3. Workflow Definition (.github/workflows/deploy.yml)

```
name: Deploy to GitHub Pages

on:
  push:
    branches: ['main', 'master']
  workflow_dispatch:

permissions:
  contents: read
  pages: write
  id-token: write

concurrency:
  group: 'pages'
  cancel-in-progress: false

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: Set up Node.js
        uses: actions/setup-node@v4
        with:
          node-version: '24' # Updated for active Node.js 24 runner LTS

      - name: Install dependencies
        run: npm install --legacy-peer-deps

      - name: Build application
        run: npm run build

      - name: Setup Pages
        uses: actions/configure-pages@v5

      - name: Upload artifact
        uses: actions/upload-pages-artifact@v3
        with:
          path: './dist'

  deploy:
    environment:
      name: github-pages
      url: ${ steps.deployment.outputs.page_url }
    runs-on: ubuntu-latest
    needs: build
    steps:
      - name: Deploy to GitHub Pages
        id: deployment
        uses: actions/deploy-pages@v4
```