

GitHub Pages에 Astro 문서 사이트 올리기

| 작성: 2026-09-07

| 대상: docs-site/ Astro Starlight 사이트

이 문서는 docs-site를 GitHub Pages에 자동 배포하는 방법을 설명합니다. GitHub Pages는 정적으로 생성된 파일을 제공하므로 Astro의 build 결과인 dist/를 GitHub Actions가 Pages 아티팩트로 업로드하는 방식이 적합합니다.

배포 구조

```
git push → GitHub Actions → npm ci → npm run build → docs-site/dist/
→ Pages artifact → https://<owner>.github.io/<repository>/
```

소스 저장소에 dist/를 커밋하지 않습니다. Actions가 빌드하고 결과물만 Pages에 전달합니다. GitHub는 정적 사이트 생성기처럼 별도 빌드가 필요한 경우 GitHub Actions publishing source를 사용하도록 안내합니다. GitHub Pages publishing source

1. 먼저 확인할 것

- GitHub 저장소에 이 프로젝트가 push되어 있어야 합니다.
- docs-site/package-lock.json을 함께 커밋합니다. Actions에서는 npm ci를 사용합니다.
- docs-site/public/downloads/의 Markdown과 PDF도 공개 사이트에 포함되므로 비밀값을 넣지 않습니다.
- GitHub Pages는 인터넷에 공개됩니다. 저장소가 비공개여도 요금제와 조직 설정에 따라 게시된 사이트는 공개될 수 있습니다.

```
cd C:\Developments\LivekitDev\docs-site
npm ci
npm run build
Test-Path .\dist\index.html
```

2. Astro URL 설정

배포 유형	주소 예시	base 설정
사용자/조직 사이트	https://owner.github.io/	보통 없음
프로젝트 사이트	https://owner.github.io/repository/	base: '/repository'
사용자 지정 도메인	https://docs.example.com/	보통 없음

프로젝트 사이트라면 astro.config.mjs에 저장소 이름을 넣습니다.

```
export default defineConfig({
  site: 'https://OWNER.github.io',
  base: '/REPOSITORY',
  // 기존 Starlight integrations 유지
});
```

OWNER는 GitHub 사용자 또는 조직 이름이고 REPOSITORY는 저장소 이름입니다. 사용자/조직 사이트 저장소가 OWNER.github.io와 일치하면 base가 필요하지 않습니다. Astro 공식 가이드도 프로젝트 URL에는 site와 base를 설정하도록 안내합니다. Astro GitHub Pages 배포

직접 작성한 HTML 또는 Markdown 링크는 base를 포함해야 합니다.

사용자 지정 도메인은 저장소 이름을 경로에 넣지 않습니다.

3. GitHub Actions workflow

저장소 루트에 `.github/workflows/deploy-docs.yml`을 만들고 저장합니다.

```
name: Deploy Astro docs to GitHub Pages

on:
  push:
    branches: [main]
    paths: ['docs-site/**', '.github/workflows/deploy-docs.yml']
  workflow_dispatch:

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

concurrency:
  group: pages
  cancel-in-progress: true

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/configure-pages@v5
      - uses: actions/setup-node@v4
        with:
          node-version: 22
          cache: npm
          cache-dependency-path: docs-site/package-lock.json
      - name: Install uv
        uses: astral-sh/setup-uv@v6
        with:
          enable-cache: true
      - name: Install dependencies
        working-directory: docs-site
        run: npm ci
      - name: Build Astro site
        working-directory: docs-site
        run: npm run build
      - uses: actions/upload-pages-artifact@v3
        with:
          path: docs-site/dist

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

`configure-pages`, `upload-pages-artifact`, `deploy-pages`가 각각 Pages 설정, dist/ 업로드, 배포를 담당합니다.

현재 사이트의 prebuild가 `uv run python scripts/sync_docs.py`를 실행하므로 workflow에 uv 설치 단계를 포함했습니다. GitHub custom workflow

4. Pages 활성화

1. 저장소의 Settings → Pages로 이동합니다.
2. Build and deployment → Source에서 GitHub Actions를 선택합니다.
3. workflow를 main에 push합니다.
4. Actions 탭에서 실행 결과를 확인합니다.
5. 성공하면 github-pages environment에 게시 URL이 표시됩니다.

Branch 방식의 /docs 폴더 게시 대신, 이 Astro 사이트는 dist/를 Actions artifact로 배포합니다.

5. 배포 확인

```
https://OWNER.github.io/REPOSITORY/  
https://OWNER.github.io/REPOSITORY/guides/getting-started/  
https://OWNER.github.io/REPOSITORY/downloads/getting-started.pdf
```

프로젝트 사이트에서 홈은 열리지만 CSS-검색-사이드바가 404이면 astro.config.mjs의 base가 저장소 이름과 일치하는지 확인합니다. 브라우저 Network에서 /_astro/와 Pagefind 요청도 확인합니다.

6. 사용자 지정 도메인

1. GitHub 저장소 Settings → Pages → Custom domain에 도메인을 먼저 입력하고 저장합니다.
2. DNS에서 해당 하위 도메인의 CNAME을 OWNER.github.io로 지정합니다.
3. DNS 전파 후 Pages 설정에서 상태를 확인합니다.
4. Enforce HTTPS를 켭니다.

Actions 방식에서는 CNAME 파일을 직접 만들 필요가 없습니다. Pages 설정의 Custom domain을 사용합니다. GitHub는 도메인을 Pages에 먼저 등록한 뒤 DNS를 연결하도록 안내합니다. GitHub custom domains

```
Type: CNAME  
Name: docs  
Value: OWNER.github.io
```

인증서가 준비되기까지 시간이 걸릴 수 있으며, GitHub Pages는 HTTPS 강제 옵션을 제공합니다. GitHub HTTPS

7. 공개 전 보안 점검

- public/downloads/에는 API secret, FileShare 토큰, 인증서 키, 개인정보를 넣지 않습니다.
- devkey/secret 같은 개발 예시는 공개 배포 전에 placeholder로 교체합니다.
- .env, config.json, 로그, node_modules, .astro, dist를 저장소에 커밋하지 않습니다.
- GitHub Pages는 서버 실행·WebSocket 프록시·파일 업로드 API를 제공하지 않습니다. Caddy·Meet·FileShare는 기존 서버에서 별도로 운영합니다.

8. 오류 해결

증상	확인할 내용
workflow가 보이지 않음	위치가 .github/workflows/deploy-docs.yml인지 확인
npm ci 실패	package.json과 package-lock.json을 함께 커밋했는지 확인
uv 오류	workflow에 astral-sh/setup-uv@v6가 있는지 확인

증상	확인할 내용
CSS·검색 404	base가 저장소 이름과 일치하는지 확인
내부 링크 404	직접 작성한 절대 링크에 /REPOSITORY를 포함했는지 확인
custom domain HTTPS 대기	Pages 설정, CNAME, DNS 전파, Enforce HTTPS 확인
변경 미반영	paths 조건, main push, Actions 실행 및 캐시 확인
PDF·이미지 누락	public/ 아래에 있고 dist/에 복사되는지 확인

설정 오류를 수정한 뒤 Actions를 다시 실행합니다. 소스 변경 없이 workflow만 재실행하려면 workflow_dispatch를 사용합니다.

9. 배포 전 체크리스트

- [] 저장소 이름에 맞는 site와 base 설정
- [] workflow 추가 및 package-lock.json 커밋
- [] Actions 권한과 Pages source를 GitHub Actions로 설정
- [] npm run build 로컬 성공
- [] 공개하면 안 되는 비밀값 제거
- [] 홈, 문서, 검색, PDF 링크 확인
- [] custom domain 사용 시 Pages에 먼저 등록하고 DNS CNAME 연결
- [] HTTPS 인증서가 준비된 뒤 Enforce HTTPS 활성화

참고 링크

- Astro: Deploy your Astro Site to GitHub Pages
- GitHub: Configuring a publishing source
- GitHub: Managing a custom domain
- GitHub: Enforce HTTPS