

Cloudflare Pages와 Vercel에 Astro 사이트 배포하기

| 작성: 2026-09-07

| 대상: docs-site/ Astro Starlight 문서 사이트

이 프로젝트는 Astro 정적 사이트입니다. 로컬 `npm run build`가 생성하는 `docs-site/dist/`를 Cloudflare Pages 또는 Vercel에 올리면 웹으로 서비스할 수 있습니다. 두 서비스 모두 GitHub 저장소에 push할 때 자동으로 다시 빌드할 수 있습니다.

먼저 알아둘 점

Cloudflare Pages와 Vercel은 이 문서 사이트 같은 정적 결과물을 서비스합니다. Caddy, Meet, LiveKit, FileShare API와 같은 기존 서버 프로세스는 이 서비스에서 실행되지 않습니다.

정적 문서: `docs-site/dist/` → Cloudflare Pages 또는 Vercel
동적 서비스: Meet, LiveKit, FileShare → 기존 Caddy 서버

`docs-site`는 빌드 전에 `uv run python scripts/sync_docs.py`를 실행합니다. 따라서 원격 빌드 환경에서도 `uv`를 설치하거나, 이미 통합된 콘텐츠만 빌드하는 CI용 명령을 별도로 마련해야 합니다.

공통 준비

1. GitHub 저장소에 `docs-site/`와 `docs-site/package-lock.json`을 push합니다.
2. 공개하면 안 되는 값이 `docs-site/public/downloads/`나 Markdown에 없는지 확인합니다.
3. 로컬에서 빌드합니다.

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

사이트가 저장소 하위 프로젝트로 배치되는 구조라면 Astro의 `site`와 `base`도 설정해야 합니다.

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

사용자 지정 도메인으로 서비스할 때는 보통 `base`를 설정하지 않습니다.

1. Cloudflare Pages

Cloudflare 공식 Astro 설정은 `npm run build`와 `dist`를 사용합니다. Pages Git 연동은 저장소의 Root directory, Build command, Build output directory를 지정하고 push마다 배포합니다. Cloudflare Astro 가이드, Cloudflare build 설정

대시보드 배포

1. Cloudflare Dashboard에서 Workers & Pages → Create application → Pages → Import an existing Git repository로 이동합니다.
2. GitHub 저장소를 연결합니다.
3. 다음 값을 입력합니다.

항목	값
Production branch	main
Root directory	docs-site
Framework preset	Astro
Build command	npm run build
Build output directory	dist
Node.js	package-lock에 맞는 LTS 버전

4. Save and Deploy를 누릅니다.

5. 생성된 *.pages.dev 주소에서 홈, 문서 페이지, PDF 다운로드를 확인합니다.

Root directory를 docs-site로 지정하면 설치와 빌드가 그 폴더에서 실행됩니다. 저장소 루트에 다른 프로젝트가 있어도 문서 사이트만 배포할 수 있습니다.

uv 동기화 처리

현재 prebuild가 uv를 호출하므로 Cloudflare 빌드 로그에 uv: command not found가 나타날 수 있습니다. 다음 중 하나를 선택합니다.

방법 A: 빌드 명령에서 uv를 준비하는 경우

Cloudflare 빌드 환경에서 사용할 수 있는 uv 설치 명령을 프로젝트의 Build command 앞에 넣습니다. 설치 방식은 Cloudflare 빌드 이미지에 따라 달라질 수 있으므로 첫 배포 로그에서 확인합니다.

방법 B: CI용 동기화와 사이트 빌드를 분리하는 경우

문서 원본과 생성 페이지가 같은 commit에 들어 있는 배포 브랜치에서는 prebuild를 실행하지 않는 별도 script를 사용합니다. Astro 자체 빌드만 실행하는 예시입니다.

```
{
  "scripts": {
    "build:static": "astro build"
  }
}
```

그 후 Cloudflare Build command를 npm run build:static으로 설정합니다. 이 방식은 public/downloads/와 src/content/docs/가 항상 같은 commit에서 이미 갱신되어 있다는 전제가 필요합니다.

사용자 지정 도메인

Cloudflare Pages 프로젝트의 Custom domains → Set up a custom domain에서 도메인을 추가합니다. Cloudflare DNS를 사용한다면 안내에 따라 레코드를 연결하고, 외부 DNS를 사용한다면 Pages가 안내하는 CNAME을 등록합니다. 인증서와 HTTPS 상태가 준비된 뒤 실제 도메인으로 확인합니다.

현재 사이트의 Caddy 도메인과 같은 이름을 Cloudflare Pages에 연결하면 DNS 대상이 바뀔 수 있습니다. 기존 Caddy 서비스와 같은 도메인을 사용할 때는 먼저 어느 서비스가 해당 도메인을 소유할지 결정해야 합니다.

2. Vercel

Vercel은 framework를 자동 감지하지만, 이 저장소처럼 앱이 하위 폴더에 있는 경우 Root Directory를 지정해야 합니다. Vercel 공식 문서는 Root Directory, Build Command, Output Directory를 프로젝트 설정에서 조정할 수 있다고 설명합니다. Vercel build 설정

대시보드 배포

1. Vercel에서 Add New → Project를 선택합니다.
2. GitHub 저장소를 Import합니다.
3. 다음 값을 입력합니다.

항목	값
Framework preset	Astro
Root Directory	docs-site
Build command	npm run build
Output directory	dist
Install command	npm ci 또는 자동 감지

4. Deploy를 누릅니다.
5. Deployment URL에서 문서와 PDF를 확인합니다.

Root Directory를 docs-site로 지정하면 Vercel은 그 폴더를 프로젝트 루트로 취급합니다. 상위 폴더의 파일을 ..으로 참조할 수 없으므로 문서 원본은 현재처럼 docs-site 안에 통합되어 있어야 합니다.

vercel.json으로 고정하기

대시보드 설정 대신 docs-site/vercel.json에 출력 폴더를 고정할 수 있습니다.

```
{
  "$schema": "https://openapi.vercel.sh/vercel.json",
  "outputDirectory": "dist"
}
```

Root Directory는 대시보드에서 docs-site로 설정합니다. outputDirectory는 프로젝트 폴더 기준의 상대 경로입니다. Vercel vercel.json

CLI 배포

Vercel CLI를 사용할 때도 docs-site에서 실행합니다.

```
cd C:\Developments\LivekitDev\docs-site
npx vercel login
npx vercel
npx vercel --prod
```

첫 실행에서 연결할 계정과 프로젝트를 선택합니다. Git 연동을 사용하면 이후 push마다 Preview와 Production 배포를 자동으로 만들 수 있습니다. Vercel CLI deploy

uv 동기화 처리

Vercel 빌드도 현재 prebuild를 실행하므로 uv 설치가 필요할 수 있습니다. 빌드가 uv: command not found로 실패하면 다음 중 하나를 적용합니다.

- 프로젝트의 Build command를 uv 설치 후 npm run build가 되도록 별도 script로 구성합니다.
- build:static: "astro build"를 추가하고 Vercel Build command를 npm run build:static으로 지정합니다.

두 번째 방식은 이미 src/content/docs/가 갱신된 commit만 배포하는 경우에 사용합니다.

무료 플랜 주의사항

Cloudflare Pages는 무료 플랜에서 월 500회 빌드, 사이트당 파일 20,000개, 파일당 25 MiB 제한이 있습니다. 정적 파일 요청은 Pages Functions를 사용하지 않는 한 무료로 처리됩니다. Cloudflare Pages 제한, Cloudflare Pages Functions 요금

Vercel Hobby는 무료이지만 개인·비상업적 사용을 위한 플랜입니다. 상업적 서비스나 업무용 사이트라면 Vercel 약관과 적합한 요금제를 확인합니다. Vercel Hobby 플랜

도메인 이름 자체의 등록비는 두 서비스의 무료 호스팅과 별개입니다.

배포 후 점검

```
/
/guides/getting-started/
/guides/github-pages-deploy/
/downloads/getting-started.pdf
```

- CSS, 검색, 사이드바가 로드되는지 확인합니다.
- PDF 링크가 200으로 응답하는지 확인합니다.
- 프로젝트 URL이 하위 경로라면 base가 모든 내부 링크와 자산에 적용됐는지 확인합니다.
- Cloudflare Pages 또는 Vercel의 build log에서 dist/index.html 생성 여부를 확인합니다.
- 새 commit 후 자동 배포가 실행되는지 확인합니다.

문제 해결

증상	원인과 조치
package.json을 찾지 못함	Root Directory를 docs-site로 설정
dist가 없음	Build command와 Astro 빌드 로그 확인
uv를 찾을 수 없음	uv 설치 단계 추가 또는 astro build 전용 script 사용
CSS·검색이 404	프로젝트 URL의 base를 저장소 경로로 설정
PDF가 없음	public/downloads/에 파일이 있고 dist/downloads/로 복사됐는지 확인
Cloudflare/ Vercel 도메인 접속 실패	기존 Caddy DNS와 중복되지 않는지 확인
Vercel 배포 사용 가능 여부	Hobby 플랜의 비상업적 사용 조건 확인

추천 선택

- 공개 문서 사이트와 사용자 지정 도메인이 필요하다면 Cloudflare Pages를 먼저 검토합니다.
- 개인·비상업 프로젝트에서 Preview 배포 경험이 중요하다면 Vercel이 편합니다.
- 현재 LiveKitDev 운영 서비스는 두 서비스로 옮기지 않고 기존 Caddy에서 계속 운영합니다.