



# Cloudflare Pages to Workers Migration

## Cloudflare Pages から Workers への移行

Prepared for / 宛先

**Japan Activation Capital**

**ジャパン・アクティベーション・キャピタル**

March 11, 2026 / 2026 年 3 月 11 日

---

**English Version**

[See page 3 →](#)

**日本語版**

[14 ページへ →](#)



# Cloudflare Pages to Workers Migration

Prepared for

## Japan Activation Capital

March 11, 2026

---

### Contents

Executive Summary .....	3
Why We Moved .....	3
What Changed in the Codebase .....	4
New Files .....	4
Modified Files .....	4
Deployment Architecture .....	5
Cutover Plan .....	6
Workers Configuration .....	7
Variable Migration Reference .....	8
Build-time Variables (Workers Build Settings) .....	8
Runtime Secrets ( <code>wrangler secret put</code> ) .....	8
Immediate Benefit: Vulnerability Scanner Detection .....	9
Improvement: Browser Language Detection .....	10
Previous Behavior .....	10
New Behavior .....	10
Additional Work Completed .....	11
Staging Environment Migration .....	11
Post-Migration Improvements .....	11
524 Incident Report Follow-up .....	12
Remaining Item .....	13
Contact Us .....	13

---

## Executive Summary

**Prepared by:** eSolia Inc.

**Date:** 21 February - 11 March 2026

**Site:** [japanactivationcapital.com](https://japanactivationcapital.com)

**Status:** Complete (domain cutover done, Pages retained as rollback)

We migrated the [japanactivationcapital.com](https://japanactivationcapital.com) website from [Cloudflare Pages](#) to [Cloudflare Workers](#). This was the primary recommendation from the [524 timeout](#) incident report (eSolia-JAC-Cloudflare-524-Timeout-Report-Client-20260220-en.md): Pages offered no server-side logging, which made diagnosing production issues unnecessarily difficult.

The migration is complete. The production domain is now served by Workers, and the staging environment has also been migrated. The previous Pages deployment is retained temporarily as a rollback option.

---

## Why We Moved

Cloudflare Pages was originally the recommended way to deploy [SvelteKit](#) sites. That has changed: Cloudflare is consolidating Pages into Workers, and Workers provides real operational advantages:

Capability	Pages	Workers
<a href="#">Real-time log streaming</a> ( <code>wrangler tail</code> )	No	Yes
Request-level observability	No	Yes
<a href="#">Smart placement</a> (datacenter selection)	No	Yes
Trace sampling and diagnostics	No	Yes
Future platform investment	Winding down	Active

The JAC Cloudflare account already has Workers Paid active. There is no pricing change.

---

## What Changed in the Codebase

Six files were modified or created. No application logic was changed: all modifications are deployment configuration and TypeScript types.

### New Files

`**wrangler.jsonc**` — Workers deployment configuration. Defines the Worker name (`jac-main-site`), asset handling, smart placement, and full observability (logs + traces at 100% sampling). Custom domain routes are commented out until the cutover step.

`**dev.vars**` — Local development secrets for `wrangler dev`. Contains basic auth credentials for local testing. This file is gitignored.

### Modified Files

`**src/app.d.ts**` — Added the `Platform` interface with TypeScript types for Workers bindings (environment variables like `ENABLE_BASIC_AUTH`, `BASIC_AUTH_USERNAME`, etc., plus Cloudflare-specific objects like `cf` and `ctx`). This gives proper type checking for `platform.env` access throughout the codebase, catching configuration errors at build time rather than at runtime.

`**src/hooks.server.ts**` — Removed a `@ts-expect-error` suppression (a directive that tells the TypeScript compiler to ignore a type error on the next line). The new `Platform` interface in `app.d.ts` makes this workaround unnecessary because the types are now properly defined.

`**github/workflows/notify-search-engines.yml**` — No changes needed for the migration. This workflow was implemented by eSolia after the initial site launch. It is part of a two-layer search engine notification pipeline:

- **Layer 1 — GitHub Actions:** On each push to the `production` branch, the workflow waits for the Cloudflare Workers Build to complete, then calls a dedicated indexing Worker (`jac-indexing`) via its `/ping` endpoint with a bearer token for authentication.
- **Layer 2 — Indexing Worker:** The `jac-indexing` Worker (deployed separately from the main site) fetches the sitemap, then submits all URLs to search engines via two channels: [IndexNow](#) (a single API call that covers Bing, Yandex, Seznam, and Naver simultaneously) and the [Google Indexing API](#) (individual `URL_UPDATED` notifications, authenticated via service account [JWT](#)). The Worker validates that all submitted URLs belong to `japanactivationcapital.com` to prevent abuse.

[Cloudflare Workers Builds](#) handles the site build and deploy automatically via Git integration. The GitHub Action only triggers the post-deploy search engine notification, keeping GitHub Actions minutes usage minimal. This is the same approach eSolia uses for other client sites.

`**gitignore**` — Already had `.dev.vars` listed. No change needed.

---

## Deployment Architecture

```
Push to 'production' branch
  ↓
Cloudflare Workers Builds (Git integration)
  - Runs: yarn install → yarn build
  - Deploys to Workers
  ↓
GitHub Actions (lightweight)
  - Waits 120 seconds for CF build
  - Pings search engine indexing worker
```

Build-time environment variables ( `PUBLIC_SENTRY_DSN` , `PUBLIC_ENVIRONMENT` , etc.) are configured in the Workers Build settings in the Cloudflare dashboard, not in GitHub Actions. eSolia follows this pattern across all managed Workers deployments.

Runtime secrets ( `BASIC_AUTH_PASSWORD` , `BASIC_AUTH_USERNAME` , `ENABLE_BASIC_AUTH` ) are set via `wrangler secret put` and stored encrypted by Cloudflare. The server-side request handler ( `hooks.server.ts` ) reads them at request time.

## Cutover Plan

We followed a staged approach with zero downtime. Pages remained online as a fallback throughout the process.

Step	Action	Status
1	Create API token with Workers permissions	Done
2	Deploy Worker to staging URL	Done
3	Set runtime secrets (encrypted by Cloudflare)	Done
4	Verify all pages on Workers URL (homepage, /team/, /en/, /sitemap.xml, /robots.txt, /llms.txt)	Done
5	Confirm real-time log streaming works	Done
6	Connect Git integration in Workers Build settings	Done
7	Disconnect Git from Pages project	Done
8	Enable production routes, deploy	Done
9	Remove custom domain from Pages	Done
10	Verify japanactivationcapital.com loads on Workers	Done
11	Keep Pages project as rollback	In progress

**Rollback procedure:** If anything goes wrong, the custom domain can be re-pointed to the Pages project. The existing Pages deployment is still intact.

## Workers Configuration

Parameter	Value
Worker name	<code>jac-main-site</code>
Staging URL	<code>https://jac-main-site.teamsds-jaccf.workers.dev</code>
Compatibility date	2026-02-20
Compatibility flags	<code>nodejs_compat</code>
Placement	Smart (Cloudflare selects optimal datacenter)
Log sampling	100%
Trace sampling	100%
Worker bundle size	~4 KB (paid-tier limit: 10 MB)
Adapter	<code>@sveltejs/adapter-cloudflare</code> v7.2.5 (same adapter, no change)

## Variable Migration Reference

### Build-time Variables (Workers Build Settings)

Variable	Value
<code>PUBLIC_ENVIRONMENT</code>	<code>production</code>
<code>PUBLIC_HIDE_ARTICLE</code>	<code>1</code>
<code>PUBLIC_SENTRY_DSN</code>	(same as Pages)
<code>PUBLIC_SENTRY_ENABLED</code>	<code>1</code>
<code>SENTRY_AUTH_TOKEN</code>	(same as Pages, encrypted)
<code>NODE_VERSION</code>	<code>24.4.0</code>
<code>YARN_VERSION</code>	<code>1.22.22</code>

### Runtime Secrets (`wrangler secret put`)

Secret	Notes
<code>BASIC_AUTH_PASSWORD</code>	Same value as Pages
<code>BASIC_AUTH_USERNAME</code>	Same value as Pages
<code>ENABLE_BASIC_AUTH</code>	Set to <code>0</code> for production

---

## Immediate Benefit: Vulnerability Scanner Detection

Within minutes of the cutover, Workers observability logs revealed automated vulnerability scanners probing the site. These requests were invisible under Pages (no server-side logging).

### Probes detected:

Path	Purpose
<code>/debug/default/view</code>	Scanning for exposed debug endpoints
<code>/.vscode/sftp.json</code>	Scanning for exposed IDE credentials
<code>/@vite/env</code>	Scanning for exposed Vite dev server
<code>/.env.dev</code> , <code>/.env.staging</code> , <code>/.env.example</code> , <code>/.env.vite</code>	Scanning for leaked environment secrets
<code>/config.json</code> , <code>/env.json</code> , <code>/constants.js</code>	Scanning for exposed application config
<code>/payment.js</code>	Scanning for exposed payment logic
<code>/wp-content/debug.log</code>	WordPress scanner (this site is not WordPress)

SvelteKit correctly returned 404 for all of these: no data was exposed. However, the requests consumed Worker invocations unnecessarily.

**Action taken:** Enabled [Super Bot Fight Mode \(SBFM\)](#) in Cloudflare's security settings and initially set "Definitely automated traffic" to **Block**. This stopped the scanners but also blocked Bingbot, despite Cloudflare's documentation stating that verified bots are exempt from SBFM rules.

After investigation, we changed the setting to [Managed Challenge](#) and added explicit WAF rules to pass through verified crawlers (Googlebot, Bingbot, etc.) and permitted AI search bots. Cloudflare's own documentation is contradictory on how SBFM interacts with verified bots, so we arrived at the final configuration through testing. Crawler access is now stable. Details are in the WAF hardening report (eSolia-JAC-WAF-Hardening-Report-Client-20260306-en.md).

This is a concrete example of the observability gap that motivated the migration: under Pages, these probes were invisible and could not be identified or acted on.

## Improvement: Browser Language Detection

As part of the post-migration work, we improved the site's browser language detection and language toggle behavior. The site is bilingual: Japanese at `/`, English at `/en/`.

### Previous Behavior

- First visit only: checked the `Accept-Language` header (first value, no `q-value` parsing)
- Redirected all paths (not just `/`) to the English equivalent if the browser preferred English
- Set a session-only `locale-visited` cookie to prevent re-redirects during the session
- The language toggle (JP/EN buttons) did not persist the user's choice: preference was lost when the session ended

### New Behavior

#### Detection priority:

1. `**lang` cookie\*\* — If the visitor previously chose a language via the toggle, that choice persists across sessions (1-year expiry)
2. `**Accept-Language` header with `q-value` weighting\*\* — Proper parsing (e.g., `en-US,en;q=0.9,ja;q=0.8` → English wins)
3. **Default: Japanese** — Unknown visitors see Japanese

**Key principle:** Only the root path (`/`) auto-redirects based on browser language. Deep links (`/team/`, `/strategy/`, etc.) are left alone, because someone arriving at a specific path got there intentionally (a shared link, a search result, a bookmark). Redirecting them to a different language version would break that intent. This is the [recommended approach from Google](#) for multilingual sites: avoid automatic redirects on content pages, and let `hreflang` tags guide search engines to the right language version instead.

#### Key changes:

Scenario	Before	After
First visit to <code>/</code> , browser=EN	Redirect to <code>/en/</code>	Redirect to <code>/en/</code> (same)
First visit to <code>/team/</code> , browser=EN	Redirect to <code>/en/team/</code>	No redirect (the visitor chose this URL)
Click JP toggle, return to <code>/</code> , browser=EN	Redirect to <code>/en/</code> (session cookie expired)	No redirect ( <code>lang=ja</code> cookie wins)
Toggle click persistence	Lost on session end	Persists 1 year

Two files were modified: `src/hooks.server.ts` (server-side request handler: language detection and redirect logic) and `src/components/global/Header.svelte` (the header component: sets the `lang` cookie when the user clicks the JP/EN toggle). The previous session-only `locale-visited` cookie has been replaced by the persistent `lang` cookie and root-only redirect.

## Additional Work Completed

### Staging Environment Migration

The `staging.japanactivationcapital.com` subdomain has been migrated from Pages to a separate staging Worker (`jac-main-site-staging`). It inherits all production settings (observability, smart placement, asset handling) and is protected by [Cloudflare Zero Trust Access](#).

#### Migration steps:

Step	Action	Status
1	Add <code>env.staging</code> to <code>wrangler.jsonc</code>	Done
2	Initial deploy: <code>yarn build &amp;&amp; npx wrangler deploy --env staging</code>	Done
3	Set runtime secrets for staging Worker: <code>wrangler secret put --env staging</code>	Done
4	Connect Workers Builds Git integration to <code>jac-main-site-staging</code> (branch: <code>staging</code> )	Done
5	Remove staging custom domain from Pages, re-add to staging Worker	Done
6	Verify Zero Trust Access policy still applies (same hostname, no change needed)	Done

This eliminates the last dependency on the Pages project. Pages is retained only as a temporary rollback option.

### Post-Migration Improvements

The following work was completed after the initial Workers migration.

#### SEO fixes (February-March 2026):

- Added unique meta descriptions to team member pages, homepage, and fraud warning page, addressing Bing Webmaster Tools findings (110 pages with duplicate descriptions, 71 too-short, 1 missing)
- Updated `robots.txt` to allow GPTBot (OpenAI) and ClaudeBot (Anthropic) to crawl the site. These bots surface content in AI-powered search answers (ChatGPT, Claude). Bots used solely for model training (not search) remain blocked (CCBot, Google-Extended, Bytespider, etc.). Note: Google-

Extended only controls training data for Gemini/Vertex AI; it does not affect Google Search or AI Overviews, which use Googlebot (allowed).

### Stability fixes:

- Fixed null guard on menu data in layout (prevented TypeError on edge cases)
- Wrapped `sessionStorage` calls in try/catch for Safari private browsing compatibility
- Removed leftover `console.log` statements from production code

### WAF hardening (6 March 2026):

Documented separately in eSolia-JAC-WAF-Hardening-Report-Client-20260306-en.md. Eight custom WAF rules deployed: probe blocking, geo-restrictions, scanner user-agent blocking, and AI scraper filtering. Early results: 679 probe requests blocked in the first 6 hours. A Bing indexing issue caused by a too-narrow bot skip rule was identified and fixed.

### CI/CD tightening:

- Hardened the GitHub Actions workflow: added `permissions: {}` (least-privilege) and `timeout-minutes: 5`
- Reduced post-deploy wait from 120 seconds to 30 seconds, as Workers Builds finishes faster than initially estimated

### 524 Incident Report Follow-up

All items from the original 524 timeout incident report have been completed:

- **Cache pre-warming on deploy:** The GitHub Actions post-deploy workflow now warms key pages (both languages of homepage and team, plus sitemap and robots.txt) before notifying search engines. This populates the Worker's in-memory Prismic cache so the first end users get cached responses.
- **Extended stale serving (5 min → 30 min):** The Prismic response cache TTL has been extended from 5 minutes to 30 minutes. For a corporate content site that publishes a handful of articles per month, 30 minutes is a better trade-off: it reduces Prismic API calls and provides a wider safety margin if Prismic's API goes down. The stale fallback (serving expired cache when Prismic is unreachable) still works beyond the 30-minute window.
- **Google Indexing API:** Fully integrated into the `jac-indexing` Worker. On each deploy (and daily via cron), the Worker authenticates via service account JWT, then sends `URL_UPDATED` notifications to Google for all sitemap URLs. This runs in parallel with IndexNow (Bing, Yandex, Seznam, Naver) and Wayback Machine archival. Google's 200 requests/day quota is handled gracefully: if exhausted, the step stops without retrying.
- **Prismic webhook integration:** A Prismic webhook notifies the `jac-indexing` Worker whenever content changes (team members, news articles). This triggers immediate search engine notification via IndexNow and Google Indexing API, ensuring SEO updates are event-driven rather than limited to deploy cycles.

---

## Remaining Item

**Prismic health monitoring** (low priority): Planned via Cloudflare Health Checks — a zero-code monitor on the Prismic CDN API endpoint with email notification. The current stale-cache fallback already handles Prismic outages transparently (the site continues serving cached content), so this adds awareness rather than resilience.

---

## Contact Us

**eSolia Inc.** Shiodome City Center 5F (Workstyling) 1-5-2 Higashi-Shimbashi, Minato-ku Tokyo 105-7105, Japan

<b>Phone</b>	03-4577-3380
<b>Email</b>	<a href="mailto:rick.cogley@esolia.co.jp">rick.cogley@esolia.co.jp</a>
<b>Web</b>	<a href="https://esolia.co.jp/en">https://esolia.co.jp/en</a>
<b>Hours</b>	Monday-Friday, 9:00-18:00 JST

# Cloudflare Pages から Workers への移行

宛先

ジャパン・アクティベーション・キャピタル

2026 年 3 月 11 日

## 目次

概要 .....	15
移行の理由 .....	15
コードベースの変更内容 .....	16
新規ファイル .....	16
変更ファイル .....	16
デプロイアーキテクチャ .....	17
切り替え手順 .....	18
Workers 設定 .....	19
変数移行リファレンス .....	20
ビルド時変数 (Workers Build 設定) .....	20
ランタイムシークレット (wrangler secret put) .....	20
即時効果：脆弱性スキャナーの検出 .....	21
改善：ブラウザ言語検出 .....	22
改善前の動作 .....	22
改善後の動作 .....	22
追加作業の完了報告 .....	23
ステージング環境の移行 .....	23
移行後の改善 .....	23
524 障害レポートのフォローアップ .....	24
残項目 .....	25
お問い合わせ .....	25

## 概要

**作成:** 株式会社イソリア

**日付:** 2026年2月21日~3月11日

**サイト:** japanactivationcapital.com

**ステータス:** 完了 (ドメイン切り替え済み、Pages をロールバック用に保持)

japanactivationcapital.com のホスティング基盤を [Cloudflare Pages](#) から [Cloudflare Workers](#) に移行しました。[524 タイムアウト障害レポート \(eSolia-JAC-Cloudflare-524-Timeout-Report-Client-20260220-ja.md\)](#) での主要な推奨事項でした。Pages ではサーバーサイドのログが取得できず、本番環境の問題調査に支障がありました。

移行は完了しました。本番ドメインは Workers で配信されており、ステージング環境も移行済みです。以前の Pages デプロイは、ロールバック用に一時的に保持しています。

## 移行の理由

Cloudflare Pages は元々 [SvelteKit](#) サイトの推奨デプロイ先でした。しかし Cloudflare は Pages を Workers に統合する方向に舵を切り、Workers には運用上の明確な優位性があります。

機能	Pages	Workers
<a href="#">リアルタイムログ配信</a> ( wrangler tail )	非対応	対応
リクエスト単位のオブザーバビリティ	非対応	対応
<a href="#">スマートプレースメント</a> (最適データセンター選択)	非対応	対応
トレースサンプリングと診断	非対応	対応
プラットフォームへの今後の投資	縮小方向	活発

JAC の Cloudflare アカウントでは Workers Paid が有効済みのため、料金の変更はありません。

## コードベースの変更内容

6 ファイルの追加・変更を行いました。アプリケーションロジックの変更はなく、すべてデプロイ設定と TypeScript 型定義の変更です。

### 新規ファイル

`**wrangler.jsonc**` – Workers デプロイ設定。Worker 名 ( `jac-main-site` )、アセット処理、スマートプレースメント、完全なオブザーバビリティ ( ログ・トレース 100% サンプリング ) を定義。カスタムドメインのルートは切り替えステップまでコメントアウト。

`**dev.vars**` – `wrangler dev` 用のローカル開発シークレット。Basic 認証のテスト用認証情報を含む。gitignore 対象。

### 変更ファイル

`**src/app.d.ts**` – Workers バインディング用の Platform インターフェース ( TypeScript 型定義 ) を追加。環境変数 ( `ENABLE_BASIC_AUTH`、`BASIC_AUTH_USERNAME` 等 ) や Cloudflare 固有のオブジェクト ( `cf`、`ctx` ) の型を定義し、`platform.env` アクセス時にビルド段階で設定エラーを検出できるようになりました。

`**src/hooks.server.ts**` – `@ts-expect-error` 抑制コメント ( TypeScript コンパイラに次の行の型エラーを無視させるディレクティブ ) を削除。 `app.d.ts` で型が正しく定義されたことで、このワークアラウンドが不要になりました。

`**github/workflows/notify-search-engines.yml**` – 移行に伴う変更なし。このワークフローはサイト公開後にイソリアが実装したものです。2 層構成の検索エンジン通知パイプラインの一部です。

- **第 1 層 – GitHub Actions :** `production` ブランチへのプッシュのたびに、Cloudflare Workers Build の完了を待ってから、専用のインデキシング Worker ( `jac-indexing` ) の `/ping` エンドポイントを Bearer トークン認証付きで呼び出します。
- **第 2 層 – インデキシング Worker :** `jac-indexing` Worker ( メインサイトとは別にデプロイ ) がサイトマップを取得し、2 つのチャンネルで検索エンジンに全 URL を送信します。 [IndexNow](#) ( 1 回の API コールで Bing、Yandex、Seznam、Naver に同時通知 ) と [Google Indexing API](#) ( サービスアカウント [JWT](#) 認証による個別の `URL_UPDATED` 通知 ) です。Worker は送信される URL が `japanactivationcapital.com` に属することをバリデーションし、不正利用を防止します。

[Cloudflare Workers Builds](#) が Git 連携でサイトのビルドとデプロイを自動処理し、この GitHub Action はデプロイ後の検索エンジン通知のみを担当するため、GitHub Actions の使用時間を最小限に抑えています。イソリアが他のクライアントサイトで使用しているのと同じ方式です。

`**gitignore**` – `.dev.vars` は既に記載済み。変更不要。

## デプロイアーキテクチャ

```
'production' ブランチへのプッシュ
↓
Cloudflare Workers Builds (Git 連携)
- 実行: yarn install → yarn build
- Workers へデプロイ
↓
GitHub Actions (軽量)
- CF ビルド完了まで 120 秒待機
- 検索エンジンインデキシング Worker へ通知
```

ビルド時環境変数（`PUBLIC_SENTRY_DSN`、`PUBLIC_ENVIRONMENT` 等）は Cloudflare ダッシュボードの Workers Build 設定で管理します（GitHub Actions ではありません）。イソリアが管理する Workers デプロイ全体で共通のパターンです。

ランタイムシークレット（`BASIC_AUTH_PASSWORD`、`BASIC_AUTH_USERNAME`、`ENABLE_BASIC_AUTH`）は `wrangler secret put` で設定し、Cloudflare が暗号化して保管します。サーバーサイドの処理ハンドラ（`hooks.server.ts`）がリクエスト時に読み取ります。

## 切り替え手順

ダウンタイムなしで段階的に進めました。切り替え中も Pages はフォールバックとしてオンラインのまま維持しました。

ステップ	作業	状態
1	Workers 権限付き API トークン作成	完了
2	Worker をステージング URL にデプロイ	完了
3	ランタイムシークレット設定 (Cloudflare が暗号化保管)	完了
4	Workers URL で全ページ検証 (ホーム、/team/、/en/、/ sitemap.xml、/robots.txt、/ llms.txt)	完了
5	リアルタイムログ配信の確認	完了
6	Workers Build 設定で Git 連携を接続	完了
7	Pages プロジェクトの Git 連携を切断	完了
8	本番ルートを有効化、デプロイ	完了
9	Pages からカスタムドメインを削除	完了
10	japanactivationcapital.com が Workers で表示されることを確認	完了
11	ロールバック用に Pages プロジェクトを保持	進行中

**ロールバック手順：**問題が発生した場合、カスタムドメインを Pages プロジェクトに再設定するだけで元に戻せます。既存の Pages デプロイはそのまま保持されています。

## Workers 設定

パラメータ	値
Worker 名	jac-main-site
ステージング URL	https://jac-main-site.teamsds-jaccf.workers.dev
互換性日付	2026-02-20
互換性フラグ	nodejs_compat
プレースメント	Smart (Cloudflare が最適なデータセンターを選択)
ログサンプリング	100%
トレースサンプリング	100%
Worker バンドルサイズ	約 4 KB (有料プラン上限: 10 MB)
アダプター	@sveltejs/adapter-cloudflare v7.2.5 (変更なし)

## 変数移行リファレンス

### ビルド時変数 (Workers Build 設定)

変数	値
<code>PUBLIC_ENVIRONMENT</code>	<code>production</code>
<code>PUBLIC_HIDE_ARTICLE</code>	<code>1</code>
<code>PUBLIC_SENTRY_DSN</code>	(Pages と同一)
<code>PUBLIC_SENTRY_ENABLED</code>	<code>1</code>
<code>SENTRY_AUTH_TOKEN</code>	(Pages と同一、暗号化)
<code>NODE_VERSION</code>	<code>24.4.0</code>
<code>YARN_VERSION</code>	<code>1.22.22</code>

### ランタイムシークレット (`wrangler secret put`)

シークレット	備考
<code>BASIC_AUTH_PASSWORD</code>	Pages と同じ値
<code>BASIC_AUTH_USERNAME</code>	Pages と同じ値
<code>ENABLE_BASIC_AUTH</code>	本番は <code>0</code> に設定

## 即時効果：脆弱性スキャナーの検出

ドメイン切り替え後数分で、Workers のオブザーバビリティログに自動化された脆弱性スキャナーによるプローブが記録されました。Pages ではサーバーサイドのログがなかったため、このようなリクエストは見えませんでした。

### 検出されたプローブ：

パス	目的
<code>/debug/default/view</code>	デバッグエンドポイントの露出を探索
<code>/.vscode/sftp.json</code>	IDE の認証情報の露出を探索
<code>/@vite/env</code>	Vite 開発サーバーの露出を探索
<code>/.env.dev</code> 、 <code>/.env.staging</code> 、 <code>/.env.example</code> 、 <code>/.env.vite</code>	環境シークレットの漏洩を探索
<code>/config.json</code> 、 <code>/env.json</code> 、 <code>/constants.js</code>	アプリケーション設定の露出を探索
<code>/payment.js</code>	決済ロジックの露出を探索
<code>/wp-content/debug.log</code>	WordPress スキャナー（当サイトは WordPress ではない）

SvelteKit はこれらすべてに対して正しく 404 を返しており、データの漏洩はありません。ただし、これらのリクエストは Worker の呼び出し回数を不必要に消費していました。

**対応：** Cloudflare のセキュリティ設定で **Super Bot Fight Mode (SBFM)** を有効化し、当初「Definitely automated traffic（確実に自動化されたトラフィック）」を **Block** に設定しました。スキャナーは遮断できましたが、Bingbot もブロックされました。Cloudflare のドキュメントでは検証済みボットは SBFM ルールの対象外とされていますが、実際には異なりました。

調査の結果、設定を **Managed Challenge** に変更し、検証済みクローラー（Googlebot、Bingbot 等）と許可対象の AI 検索ボットを明示的に通過させる WAF ルールを追加しました。Cloudflare 自身のドキュメントにも SBFM と検証済みボットの相互作用について矛盾する記載があり、テストを繰り返して最終的な設定を決定しました。現在、クローラーのアクセスは安定しています。詳細は WAF 強化レポート（eSolia-JAC-WAF-Hardening-Report-Client-20260306-ja.md）を参照してください。

これは移行の動機となったオブザーバビリティの課題を具体的に示す事例です。Pages ではこれらのプローブは不可視であり、特定も対処もできませんでした。

## 改善：ブラウザ言語検出

移行後の作業として、サイトのブラウザ言語検出と言語切り替え（トグル）の動作を改善しました。サイトはバイリンガル対応で、日本語は `/`、英語は `/en/` で提供しています。

### 改善前の動作

- 初回訪問時のみ：`Accept-Language` ヘッダーを確認（最初の値のみ、`q` 値の解析なし）
- ブラウザが英語を優先する場合、すべてのパス（`/` だけでなく）を英語版にリダイレクト
- セッション限定の `locale-visited` Cookie を設定し、セッション中の再リダイレクトを防止
- 言語トグル（JP/EN ボタン）はユーザーの選択を保持せず、セッション終了時に設定が失われていた

### 改善後の動作

#### 検出の優先順位：

- `**lang` Cookie\*\* – 訪問者がトグルで言語を選択した場合、その選択がセッションをまたいで保持される（有効期限 1 年）
- `**Accept-Language` ヘッダー（`q` 値加重）\*\* – 適切な解析（例：`en-US,en;q=0.9,ja;q=0.8` → 英語が優先）
- デフォルト：日本語** – 不明な訪問者には日本語を表示

**基本方針：**ブラウザ言語による自動リダイレクトはルートパス（`/`）のみで行います。`/team/`、`/strategy/` などの深いリンクはリダイレクトしません。特定のパスに到達した訪問者は、共有リンク・検索結果・ブックマークなど、意図的にその URL にアクセスしています。別の言語版にリダイレクトすると、その意図を損なうこととなります。これは[多言語サイトに対する Google の推奨](#)に沿った方式で、コンテンツページでの自動リダイレクトを避け、`hreflang` タグで検索エンジンを適切な言語版に誘導します。

#### 主な変更点：

シナリオ	改善前	改善後
<code>/</code> に初回訪問、ブラウザ=EN	<code>/en/</code> にリダイレクト	<code>/en/</code> にリダイレクト（同じ）
<code>/team/</code> に初回訪問、ブラウザ=EN	<code>/en/team/</code> にリダイレクト	リダイレクトなし（訪問者がこの URL を選択）
JP トグルをクリック後 <code>/</code> に再訪問、ブラウザ=EN	<code>/en/</code> にリダイレクト（セッション Cookie 期限切れ）	リダイレクトなし（ <code>lang=ja</code> Cookie が優先）
トグル選択の保持	セッション終了で喪失	1 年間保持

2 つのファイルを変更しました。`src/hooks.server.ts`（サーバーサイドのリクエスト処理ハンドラ：言語検出とリダイレクトロジック）と `src/components/global/Header.svelte`（ヘッダーコンポーネント：JP/EN トグルクリック時に `lang` Cookie を設定）です。従来のセッション限定 `locale-visited` Cookie は、永続的な `lang` Cookie とルートパスのみのリダイレクトに置き換えました。

## 追加作業の完了報告

### ステージング環境の移行

`staging.japanactivationcapital.com` サブドメインを Pages から専用のステージング Worker (`jac-main-site-staging`) に移行しました。本番環境のすべての設定（オブザーバビリティ、スマートプレースメント、アセット処理）を継承し、[Cloudflare Zero Trust Access](#) で保護されています。

#### 移行手順：

ステップ	作業	状態
1	<code>wrangler.jsonc</code> に <code>env.staging</code> を追加	完了
2	初回デプロイ： <code>yarn build &amp;&amp; npx wrangler deploy --env staging</code>	完了
3	ステージング Worker のランタイム シークレット設定： <code>wrangler secret put --env staging</code>	完了
4	Workers Builds Git 連携を <code>jac-main-site-staging</code> に接 続（ブランチ： <code>staging</code> ）	完了
5	Pages からステージングのカスタム ドメインを削除、ステージング Worker に再追加	完了
6	Zero Trust Access ポリシーの確認 （ホスト名は同一のため変更不要）	完了

これにより Pages プロジェクトへの最後の依存がなくなりました。Pages はロールバック用に一時的に保持しています。

### 移行後の改善

Workers 移行後に以下の作業を実施しました。

#### SEO 改善（2026 年 2 月～3 月）：

- チームメンバーページ、ホームページ、詐欺警告ページに固有のメタディスクリプションを追加。Bing Webmaster Tools の指摘事項（重複ディスクリプション 110 ページ、短すぎ 71 ページ、欠落 1 ページ）に対応しました。
- `robots.txt` を更新し、GPTBot (OpenAI) と ClaudeBot (Anthropic) のクロールを許可。AI 検索の回答 (ChatGPT、Claude) でコンテンツが表示されるようになります。検索ではなくモデル学習専用のボットは引き続きブロック (CCBot、Google-Extended、Bytespider 等)。なお Google-Extended は Gemini/Vertex AI の学習データ収集のみを制御するもので、Google 検索や AI Overview には影響しません（それらは Googlebot が担当し、許可済み）。

## 安定性の修正：

- レイアウトのメニューデータに null ガードを追加（エッジケースでの TypeError を防止）
- Safari プライベートブラウジングとの互換性のため、`sessionStorage` 呼び出しを try/catch でラップ
- 本番コードに残っていた `console.log` を削除

## WAF 強化（2026 年 3 月 6 日）：

別途 eSolia-JAC-WAF-Hardening-Report-Client-20260306-ja.md に記載。カスタム WAF ルール 8 件をデプロイ：プローブブロック、地域制限、スキャナー UA ブロック、AI スクレイパーフィルタリング。初期結果：6 時間で 679 件のプローブリクエストをブロック。ボットスキップルールの条件が狭すぎて Bing のインデックスが停止した問題を特定・修正しました。

## CI/CD の強化：

- GitHub Actions ワークフローを強化：`permissions: {}`（最小権限）と `timeout-minutes: 5` を追加
- デプロイ後の待機時間を 120 秒から 30 秒に短縮（Workers Builds の完了が当初の想定より速いため）

## 524 障害レポートのフォローアップ

524 タイムアウト障害レポートの全項目が完了しました。

- **デプロイ時のキャッシュプリウォーミング**：GitHub Actions のデプロイ後ワークフローで、検索エンジン通知の前に主要ページ（両言語のホームページとチームページ、サイトマップ、robots.txt）をウォームアップするようにしました。Worker のインメモリ Prismic キャッシュが事前に投入されるため、最初のエンドユーザーにもキャッシュ済みのレスポンスが返ります。
- **Stale キャッシュ配信の延長（5 分 → 30 分）**：Prismic レスポンスキャッシュの TTL を 5 分から 30 分に延長しました。月に数本の記事を公開するコーポレートサイトでは、30 分のほうが適切なバランスです。Prismic API 呼び出しが減り、Prismic の API が停止した場合の安全マージンも広がります。Stale フォールバック（Prismic 到達不能時に期限切れキャッシュを配信）は 30 分を超えても引き続き機能します。
- **Google Indexing API**：`jac-indexing` Worker に完全統合済みです。デプロイごと（および毎日の cron で）、Worker がサービスアカウント JWT で認証し、サイトマップの全 URL に対して Google へ `URL_UPDATED` 通知を送信します。IndexNow (Bing, Yandex, Seznam, Naver) および Wayback Machine アーカイブと並行して実行されます。Google の 1 日 200 リクエスト上限は適切に処理され、上限到達時はリトライせず停止します。
- **Prismic Webhook 連携**：Prismic の Webhook が、コンテンツの変更（チームメンバー、ニュース記事）を `jac-indexing` Worker に通知します。IndexNow および Google Indexing API による検索エンジン通知が即座にトリガーされ、SEO の更新がデプロイサイクルに限定されず、イベント駆動で行われるようになりました。

---

## 残項目

**Prismic ヘルスモニタリング** (優先度 :低) :Cloudflare Health Checks で実施予定。Prismic CDN API エンドポイントをコード不要で監視し、メール通知を設定します。現在の Stale キャッシュフォールバックが Prismic 障害を透過的に処理している (サイトはキャッシュ済みコンテンツを配信し続ける) ため、これは耐障害性ではなく状況認識のための施策です。

---

## お問い合わせ

株式会社イソリア 〒105-7105 東京都港区東新橋 1-5-2 汐留シティセンター 5 階 (Workstyling)

電話	03-4577-3380
メール	<a href="mailto:rick.cogley@esolia.co.jp">rick.cogley@esolia.co.jp</a>
Web	<a href="https://esolia.co.jp">https://esolia.co.jp</a>
営業時間	月～金、9:00～18:00