



HK OPERATIONAL RESOURCES

Policies & Procedures

Cloudflare Deployment Verification

Document ID	HK-OPERATIONAL-V1.0.0-08072026
Effective Date & Time	August 7, 2026 / 2056
Applies To	System-wide
Status	ACTIVE POLICY

MANDATORY POLICY

A deployment is not approved because Cloudflare labels it Production, because a file uploaded successfully, or because the page is visually present. Approval requires three separate verification layers to pass: Package/File Verification, Full Functional Verification, and Deployment Identity Verification.

1. Verification Architecture

The verification system is cumulative. Each layer answers a different question. No layer replaces another, and a deployment is approved only after all three pass.

Layer	Question Answered	Required Result	Failure Meaning
1. Package / File	Did the complete package deploy and are required resources reachable?	PASS	The deployment is incomplete, stale, or structurally incorrect.
2. Full Functional	Do the real interactive controls, content, and workflows actually work?	PASS with 0 functional failures	The site is not operational even if it looks correct.
3. Deployment Identity	Which exact package and Cloudflare deployment are we approving?	Identity recorded and matched	We cannot reliably identify the approved deployment.

1.1 Layer 1 - Package / File Verification

- ☐ Verify every required file in the deployed package is present and reachable from the live site.
- ☐ Verify the main HTML, CSS, JavaScript, manifests, service worker, icons, data files, verification files, modules, repository manifests, and permanent source files that the package claims to contain.
- ☐ A referenced resource that is missing, returns an error, or cannot be loaded is a verification failure.
- ☐ The package verifier may use fingerprints or hashes as an integrity check, but those fingerprints are verification evidence - not deployment identity fields.

1.2 Layer 2 - Full Functional Verification

This layer verifies the deployed interface itself. It must exercise the real controls in the real deployed package, not a mock interface and not a static inventory of elements.

- ☐ Primary and secondary navigation controls.
- ☐ Buttons, dropdown menus, filters, toggles, tabs, cards, and modal controls.
- ☐ Search fields, including confirmation that searches return the expected real content.
- ☐ Automation simulations and workflow tools, including success and failure paths where applicable.
- ☐ Hammerspoon Script Repository functions: search, filtering, records, permanent Lua source availability, import/staging functions, source viewing, copying, downloading, and export functions that exist in the package.
- ☐ Reference modules and their internal controls.
- ☐ Persistent browser-local features, such as checklists or saved progress, where the package claims persistence.
- ☐ Content completeness: interactive controls must not be empty shells; menus, records, scripts, and data must contain the content the package claims to provide.
- ☐ Runtime health: zero uncaught page errors, zero unhandled promise errors, and no JavaScript syntax failure that disables interactivity.

Verifier launcher requirement: Full Functional Verification must be executable from the deployed site without depending on popup behavior. If the verification launcher itself does not work, the deployment has failed functional verification.

2. Deployment Identity Verification

Deployment Identity Verification is an additional control. It does not replace Package/File Verification or Full Functional Verification. Its sole purpose is to identify the exact package and Cloudflare deployment being approved.

2.1 The Only Three Identity Values

Identity Value	Authority / Source	Rule
Package ID	Internal HK package metadata	Outlined in the Cloudflare Deployment Verification P&P document and embedded in the package.
Cloudflare Deployment ID	Cloudflare	Copied directly from Cloudflare. Do not infer, shorten, or invent it.
Cloudflare Deployment URL	Cloudflare	Copied directly from Cloudflare. Do not substitute the general project URL.

Explicit exclusion: No Cloudflare Short ID, Main URL Build, Cloudflare Deployment Build, or other invented identity field belongs in the permanent identity block.

2.2 Background Matching Is a Result, Not Another ID

The verifier may compare the Package ID served by the general project URL with the Package ID served by the Cloudflare deployment-specific URL. If used, that comparison should be reported as a simple verification result such as "Main URL matches selected deployment - PASS." The comparison must not create additional identity fields.

3. Permanent Package ID Procedure

The Package ID is the internal, human-readable identity of the ZIP/package itself. It exists so the deployed application can state exactly which package version it is running.

HK - {PROJECT} - V{MAJOR} . {MINOR} . {PATCH} - {MMDDYYYY}

Example - Matrix: HK-MATRIX-V5.3.1-08072026

Example - HK Enterprise umbrella: HK-ENTERPRISE-V1.0.0-08072026

3.1 Package ID Components

Component	Meaning	Example
HK	Enterprise namespace	HK
PROJECT	Stable short project identifier	MATRIX
MAJOR	Architectural rebuild or major redesign	5
MINOR	New feature or significant capability	3
PATCH	Bug fix or corrective revision without a major feature change	1
MMDDYYYY	Package date in month-day-year order	08072026

3.2 Versioning Rules

- ☐ **MAJOR:** Increase for a substantial architecture change, major redesign, or compatibility-breaking package change.
- ☐ **MINOR:** Increase when adding a meaningful new feature, section, capability, or verification layer.
- ☐ **PATCH:** Increase for a bug fix, repair, correction, or small revision that does not add a major new capability.

No separate build number is used. If the package changes, the semantic version changes. This eliminates duplicate B01/B02-style identifiers and keeps one authoritative version identity.

4. Folder/Zip Package System Filing

There is no second naming system for any produced, executed or deployed folder/zip packages. Any and all folder/zip packages will follow the same identifier as the Package ID explained in the previous section. The project/site name will remain the full human-readable name, but it will no become the package filename under any circumstances. The only policy to be applied to this system filing is the format in which is written. Uppercase will be assigned to any formal Package ID, and lowercase for the filesystem/package make.

Example

Package ID: HK-MATRIX-V5.3.1-08072026	→	UPPERCASE
Folder: hk-matrix-v5.3.1-08072026	→	lowercase
Zip: hk-matrix-v5.3.1-08072026.zip	→	lowercase

5. Post-Deployment Approval Procedure

Step	Action	Approval Requirement
1	Deploy the package	Upload the intended package to the existing or new Cloudflare Pages project.
2	Record deployment identity	Record the Package ID from the package and copy the Cloudflare Deployment ID and Cloudflare Deployment URL directly from Cloudflare.
3	Run Package/File Verification	Confirm all required deployed resources are present, reachable, and structurally valid.
4	Run Full Functional Verification	Exercise the real interface and require all expected interactive behavior, content, and repository functions to pass.
5	Run Deployment Identity Verification	Confirm that the exact package and Cloudflare deployment being tested are the ones recorded in the three-field identity block.
6	Approve or reject	Approve only when all three verification layers pass. Any failed layer keeps the deployment unapproved.

6. Deployment Approval Record

Every approved deployment should be capable of producing or retaining a verification receipt with the following minimum record.

Project	
Package ID	
Cloudflare Deployment ID	
Cloudflare Deployment URL	
Package / File Verification	PASS / FAIL
Full Functional Verification	PASS / FAIL
Deployment Identity Verification	PASS / FAIL
Functional Failures	0 required for approval
Final Status	APPROVED / REJECTED

7. Document Authority & Compliance

This document establishes the official and authoritative verification policy. If a later chat response, implementation note, or package description conflicts with this policy, this document supersedes prior documentation unless an updated procedure is formally approved.

- ☐ This policy defines the required verification architecture and Package ID format for HK Cloudflare Pages projects.
- ☐ The package's canonical manifest is the authority for the internal Package ID.
- ☐ Cloudflare is the authority for the Cloudflare Deployment ID and Cloudflare Deployment URL.
- ☐ The deployed verifier is the authority for Package/File and Full Functional Verification results.
- ☐ Cloudflare dashboard wording such as Production, Current, or Rollback is not sufficient by itself to approve a deployment.
- ☐ A visually correct page is not sufficient. Interactivity and content must be verified.
- ☐ A successful functional test is not sufficient if the deployment identity has not been recorded.
- ☐ A deployment with a broken verifier cannot be marked approved; the verifier is part of the deployment's required functionality.
- ☐ Future amendments to this procedure must be documented as a new version of this policy rather than silently changing the rules.

FINAL APPROVAL POLICY

**PACKAGE/FILE PASS + FULL FUNCTIONAL PASS + DEPLOYMENT IDENTITY PASS =
DEPLOYMENT APPROVED**