

# FLUX WIKI

## STARTER KIT — DEPLOYMENT (PUBLISH CONTRACT)

---

THE PUBLISH ALGORITHM, S3 KEY LAYOUT, CACHE-HEADER MATRIX, CLOUDFRONT  
INVALIDATION CONTRACT, AND THE HUB+EMBED SYNC INVARIANT.

[flux.dantesisofo.com/wiki/starter-kit-deployment/](https://flux.dantesisofo.com/wiki/starter-kit-deployment/)

FLUX\_WIKI\_v2.0

JUNE 2026

FLUX DOCUMENTATION SYSTEM Layer 11 – STARTER KIT | starter-kit-deployment  
 flux.dantesisofo.com/wiki/starter-kit-deployment/

## STARTER KIT – DEPLOYMENT (PUBLISH CONTRACT)

How a draft becomes live, and how the public surfaces stay consistent. Keys/consts from CONFIGURATION.

### 1. S3 KEY LAYOUT

|                              |                                          |
|------------------------------|------------------------------------------|
| data/catalog.json            | # the index (no-cache)                   |
| <slug>/index.html            | # project page (no-cache)                |
| <slug>/large/<photo_id>.jpg  | # web image (immutable)                  |
| <slug>/thumbs/<photo_id>.jpg | # thumbnail (immutable)                  |
| <slug>/contact_sheet.png     | # immutable                              |
| <slug>/<slug>-project.zip    | # offline export (no-cache or immutable) |
| hub/index.html               | # master hub (no-cache)                  |
| hub/embed/index.html         | # iframe embed (no-cache)                |
| catalog/index.html           | # non-collection listing (no-cache)      |

### 2. CACHE-HEADER MATRIX

| Content                                 | Content-Type                 | Cache-Control                                 |
|-----------------------------------------|------------------------------|-----------------------------------------------|
| *.html, *.json                          | text/html / application/json | no-cache, no-store, must-revalidate           |
| images (large/, thumbs/, contact sheet) | image/jpeg / image/png       | public, max-age=31536000, immutable           |
| export .zip                             | application/zip              | no-cache (so re-publish updates the download) |

HTML/JSON are no-cache so updates appear immediately; images are content-addressed (id = sha) so they can be cached forever.

### 3. PUBLISH ALGORITHM – publish(run\_dir, slug)

Idempotent / replace-in-place. Steps:

1. Load submission.json; load catalog.json (S3 authoritative, local fallback).
2. **Freeze identity:** reuse existing walk\_id+slug if the slug is already published; else assign next walk\_id. Add slug to reserved\_slugs.
3. **Upload derivatives** (large/, thumbs/) – skip keys already present (immutable) for fast re-publish.
4. **Upload** contact sheet, metadata.json, project index.html, project ZIP.
5. **Upsert the catalog WalkRecord** (replace any record with same walk\_id): copy stats; set collections = walk.collections or default\_for(location); set route =

```
downsample(stats.route, ROUTE_MAX_POINTS), bbox = bbox(stats.route); cover_image =
<slug>/large/<photos[0].id>.jpg (unless a cover was chosen). Write catalog.json to S3 and
local mirror.
```

6. `build_catalog_index(catalog)` → upload LISTING\_KEY.
7. `deploy_surfaces(s3, catalog)` → upload HUB\_KEY **and** EMBED\_KEY together (§5).
8. **CloudFront invalidate** the changed paths (§4): `/<slug>/*`, `/+CATALOG_KEY`, `/+LISTING_KEY`, and `PHILLY_SURFACE_PATHS` (hub+embed).

**Acceptance:** after publish, `GET FLUX_BASE_URL/<slug>/` is 200; the hub shows the new project; `catalog.json` contains the WalkRecord.

---

#### 4. CLOUDFRONT INVALIDATION CONTRACT

- Define once: `SURFACE_PATHS = [HUB_URL, HUB_URL+"*", EMBED_URL, EMBED_URL+"*"]`.
- Every state change invalidates the changed project paths **plus** `SURFACE_PATHS` **plus** `/+CATALOG_KEY` and `/+LISTING_KEY`.
- `CreateInvalidation` is the only CloudFront write the IAM user needs.
- A wildcard `/*` invalidation is acceptable after bulk changes.

---

#### 5. THE HUB+EMBED SYNC INVARIANT (MOST IMPORTANT RULE)

There must be exactly **one** function that uploads the hub, and it uploads the embed in the same call:

```
def deploy_surfaces(s3, catalog):
    upload(s3, HUB_KEY, build_hub(catalog), "text/html", NO_CACHE)
    upload(s3, EMBED_KEY, build_embed(catalog), "text/html", NO_CACHE)
    return [HUB_KEY, EMBED_KEY]
```

Route **every** state change through it: `publish`, `unpublish`, `set_cover`, and any future editing action. **Audit rule:** `grep` for `build_hub(/HUB_KEY` upload sites – the only one allowed is inside `deploy_surfaces`. (This exact class of bug – a regenerate path that updated the hub but not the embed – is the failure to guard against.)

---

#### 6. UNPUBLISH – `unpublish(slug)`

- Delete `s3://YOUR_BUCKET/<slug>/*` (or leave + redirect if external links exist).
- Remove the WalkRecord from `catalog.json`; keep slug in `reserved_slugs`.
- `build_catalog_index` + `deploy_surfaces` + `invalidate (/<slug>/* + surfaces + catalog + listing)`.
- **Never** delete the local draft/originals (data preservation).

---

#### 7. FULL-SITE DEPLOY (STATIC ASSETS) – `deploy.py`

For one-off pushes of shared/static assets (not per-publish), wrap `aws s3 sync LOCAL_SITE/ s3://YOUR_BUCKET/ --no-progress` with correct `--cache-control` per type, then a CloudFront invalidation. **Caution:** a full sync deploys the **local** copy – keep local static sources in sync, or a sync can overwrite live pages with stale local versions.

---

## 8. CDN/CACHE TROUBLESHOOTING

---

- Change not visible? Confirm the object's `Cache-Control` is `no-cache` and an invalidation was created (the IAM user can't `GetInvalidation`, but `Create` suffices). Hard-refresh.
- Stale output after a code edit? **Restart the portal** (stale in-memory module).

Next: PROJECT WORKFLOW.

---

FLUX\_WIKI\_v2.0 – [flux.dantesisofo.com/wiki/starter-kit-deployment/](https://flux.dantesisofo.com/wiki/starter-kit-deployment/)