ALAI Static Hosting Blueprint (2026-04-20)

ALAI Static Hosting Blueprint

Author: ALAI | **Date:** 2026-04-20 | **MC:** #8481 | **Last updated:** 2026-04-20 (Phantom Domain Removal Protocol added per MC #8526; rollback fix per MC #8494)

---

1. Platform Decision

Winner: Cloudflare Pages

ALAI already runs alai.no on Cloudflare Pages and has Cloudflare as DNS provider for 6 of 12 domains. The migration path is lowest-friction of any option: git push triggers build, custom domains are free, SSL is automatic, and Cloudflare Access (already deployed for internal tools) works natively. The free tier covers unlimited sites, 500 builds/month, and unlimited bandwidth — all 12 static sites fit without spending a euro. Critically, ALAI does not need object-storage complexity (GCS/S3) or a separate CDN layer for static marketing/demo sites. Cloudflare Pages is the right tool at this scale.

The call on vendor lock-in: ALAI is already locked to Cloudflare for DNS. Extending that to hosting is concentration risk, but the blast radius is recoverable — all sites are git-backed, migrating to any other platform is a 30-minute operation per site. The cost and operational savings outweigh the risk.

Platform Comparison (12 sites, 1 GB each, 100 GB egress/month)

Criterion	Cloudflare Pages	GCP Cloud Storage + CDN	AWS S3 + CloudFront	Azure Static Web Apps
Monthly cost (12 sites)	€0 (free tier)	~€12 (storage €1.20 + CDN egress ~€10)	~€14 (S3 €0.25 + CF egress ~€8 + requests ~€6)	€0 Free / €9 Standard (2 sites free, rest €4.50/mo each)

Criterion	Cloudflare Pages	GCP Cloud Storage + CDN	AWS S3 + CloudFront	Azure Static Web Apps
Build minutes	500/month free	N/A (no built-in CI)	N/A (no built-in CI)	60 min/month free, then €0.009/min
DX (git push to live)	Native (GitHub/GitLab direct)	Requires Cloud Build + gsutil	Requires CodePipeline or GitHub Action + aws CLI	Native (GitHub Actions integrated)
Custom domains	Unlimited	Per load balancer config	Per distribution ($0.0075/10k requests)	5 per plan
SSL	Automatic, free	Managed certificate, manual setup	ACM free but requires distribution config	Automatic, free
Preview URLs per PR	Yes (automatic)	No (requires custom setup)	No (requires custom Lambda@Edge)	Yes (staging environments)
DDoS/WAF	Included free (Cloudflare network)	Cloud Armor (add-on, ~€5+/mo)	AWS Shield Standard free, WAF extra	Azure DDoS Basic free, WAF add-on
Vendor lock-in	Medium (proprietary build env, but output is static)	Low (standard GCS)	Low (standard S3)	Medium (Azure-specific config)

Decision: Cloudflare Pages wins on cost (€0 vs €12-14/mo), DX (native git integration), DDoS/WAF included, and operational alignment with existing CF infrastructure.

---

2. Deploy Blueprint

Repo Convention

Every static site lives in its own repo or a dedicated directory in a monorepo. Naming convention: `alai-<product>-web` for ALAI properties, `client-<slug>-web` for client sites. The Cloudflare Pages project name matches the repo name exactly.

Build output must be in one of: `dist/`, `out/`, `public/`, `.next/` (for Next.js static export). For plain HTML sites, the root directory is the publish directory.

Step 1: Create Cloudflare Pages Project (one-time per site)

```
# Via Cloudflare dashboard or wrangler CLI
npx wrangler pages project create <project-name> \
```

```
  --production-branch main
```

Connect GitHub repo in the Pages dashboard. Set build command and output directory per framework:

Framework	Build command	Output dir
Static HTML	(none)	/
Next.js (static export)	`next build`	`out`
Next.js (app router)	`next build`	`.next`
Astro	`astro build`	`dist`

Step 2: GitHub Actions CI (copy-paste ready)

Save as `.github/workflows/deploy.yml` in every site repo:

```
name: Deploy to Cloudflare Pages

on:
  push:
    branches: [main]
  pull_request:
    branches: [main]

jobs:
  deploy:
    runs-on: ubuntu-latest
    permissions:
      contents: read
      deployments: write
      pull-requests: write

    steps:
      - uses: actions/checkout@v4

      - uses: actions/setup-node@v4
        with:
          node-version: '20'
          cache: 'npm'
```

```
      - name: Install dependencies
        run: npm ci

      - name: Build
        run: npm run build
        env:
          NODE_ENV: production

      - name: Deploy to Cloudflare Pages
        uses: cloudflare/wrangler-action@v3
        with:
          apiToken: ${{ secrets.CLOUDFLARE_API_TOKEN }}
          accountId: ${{ secrets.CLOUDFLARE_ACCOUNT_ID }}
          command: pages deploy ./out --project-name=${{ vars.CF_PROJECT_NAME }} --branch=${{
github.ref_name }}

      - name: Comment preview URL on PR
        if: github.event_name == 'pull_request'
        uses: actions/github-script@v7
        with:
          script: |
            const { data: deployments } = await github.rest.repos.listDeployments({
              owner: context.repo.owner,
              repo: context.repo.repo,
              ref: context.payload.pull_request.head.sha,
              per_page: 1
            });
            if (deployments.length > 0) {
              github.rest.issues.createComment({
                owner: context.repo.owner,
                repo: context.repo.repo,
                issue_number: context.payload.pull_request.number,
                body: `Preview deployed:
https://${context.payload.pull_request.head.sha.substring(0,8)}.${process.env.CF_PROJECT_NAME}
.pages.dev`
              });
            }
```

For plain HTML sites with no build step, remove the `Install dependencies` and `Build` steps, and change the deploy path to `./` instead of `./out`.

Step 3: Custom Domain (one-time per site)

```
# In Cloudflare dashboard: Pages > Project > Custom Domains > Add custom domain
# Or via API:
curl -X POST
"https://api.cloudflare.com/client/v4/accounts/$CF_ACCOUNT_ID/pages/projects/$PROJECT_NAME/dom
ains" \
  -H "Authorization: Bearer $CLOUDFLARE_API_TOKEN" \
  -H "Content-Type: application/json" \
  --data '{"name":"example.alai.no"}'
```

Because ALAI uses Cloudflare DNS, the CNAME/alias record is created automatically when adding the custom domain inside Cloudflare Pages.

Preview URL Per PR

Cloudflare Pages creates a preview URL automatically for every PR push. Format: `https://<commit-hash>.<project-name>.pages.dev`. No configuration needed. Preview environments are isolated and do not affect production traffic.

Phantom Domain Removal Protocol

ZAKON: Before `vercel domains rm <phantom>` — verify real domain is not implicitly routing through phantom.

Safe sequence for phantom removal:

1. `vercel domains inspect <real-domain>` — confirm direct attachment to authoritative project
2. If real domain does NOT show direct attachment → `vercel domains add <real> --project <authoritative>` FIRST
3. `curl -sI https://<real>` — confirm HTTP 200 with new attachment
4. ONLY THEN: `vercel domains rm <phantom> --yes`
5. Re-verify: `curl -sI https://<real>` HTTP 200

Forbidden: Remove phantom without prior explicit attachment of real domain → risk implicit routing break.

Incident reference: 2026-04-20 kenyhot.pro cleanup, 35s downtime, MC #8526.

Evidence: `/Users/makinja/system/evidence/kenyhot-vercel-cleanup/execution-log-*.txt`

Rollback (< 60 seconds)

> **NOTE — wrangler 4.x breaking change:** `wrangler pages deployment rollback` was removed in wrangler 4.x. The subcommand no longer exists and the `/rollback` CF API endpoint returns 405 for direct-upload deployments. Do NOT use it. Use the alternatives below. (Reference: wrangler upstream release notes; verified in Proveo pilot on basicconsulting.no, MC #8494.)

Primary — CF API re-deploy (copy-paste ready):

```
# Required env vars — set once per shell session or in ~/.zshrc
export CF_API_TOKEN="<your-cloudflare-api-token>"   # scope: Cloudflare Pages: Edit
export CF_ACCOUNT_ID="<your-cloudflare-account-id>"
export CF_PROJECT_NAME="<project-name>"

# 1. List recent deployments and grab the target deployment ID
curl -s
"https://api.cloudflare.com/client/v4/accounts/${CF_ACCOUNT_ID}/pages/projects/${CF_PROJECT_NA
ME}/deployments" \
  -H "Authorization: Bearer ${CF_API_TOKEN}" | \
  python3 -c "import sys,json; [print(d['id'], d['created_on'][:19],
d.get('deployment_trigger',{}).get('metadata',{}).get('commit_message','')[:60]) for d in
json.load(sys.stdin)['result'][:10]]"

# 2. Re-deploy the target deployment (replace <deployment-id> with ID from step 1)
curl -s -X POST \

"https://api.cloudflare.com/client/v4/accounts/${CF_ACCOUNT_ID}/pages/projects/${CF_PROJECT_NA
ME}/deployments/<deployment-id>/retry" \
  -H "Authorization: Bearer ${CF_API_TOKEN}" \
  -H "Content-Type: application/json" | python3 -c "import sys,json; r=json.load(sys.stdin);
print('OK —', r['result']['id']) if r['success'] else print('ERROR:', r['errors'])"
```

CF reuses content-hash cache — files already on the CDN are not re-uploaded. Measured time: ~11 seconds. No build step required.

Secondary — CF Dashboard rollback (GitHub-connected repos):

1. Open https://dash.cloudflare.com > Pages > select project
2. Click "Deployments" tab
3. Find the target deployment row, click the three-dot menu
4. Select "Rollback to this deployment"
5. Confirm — live traffic switches in < 30 seconds

Total time to identify + execute: under 30 seconds for either path.

Secrets Management

Secret	Storage	How to use
`CLOUDFLARE_API_TOKEN`	GitHub repository secret	Set in: Repo > Settings > Secrets > Actions
`CLOUDFLARE_ACCOUNT_ID`	GitHub repository variable	Set in: Repo > Settings > Variables > Actions
`CF_PROJECT_NAME`	GitHub repository variable	Set per repo, matches CF Pages project name
Build-time env vars (API keys, etc.)	Cloudflare Pages > Settings > Environment variables	Available during build and at runtime for SSR

Token scope required: `Cloudflare Pages: Edit` only. Create at: https://dash.cloudflare.com/profile/api-tokens

New-Site Template (one command)

Save as `/Users/makinja/system/tools/alai-new-site.sh`:

```
#!/usr/bin/env bash
# Usage: bash alai-new-site.sh <site-name> [--framework next|html|astro]
set -euo pipefail

SITE_NAME="${1:?Usage: alai-new-site.sh <site-name> [--framework next|html|astro]}"
FRAMEWORK="${3:-html}"
REPO_DIR="/Users/makinja/ALAI/sites/${SITE_NAME}"

echo "Creating site: ${SITE_NAME} (${FRAMEWORK})"

# 1. Create repo directory
mkdir -p "${REPO_DIR}/.github/workflows"

# 2. Copy workflow template
cp /Users/makinja/system/specs/templates/cf-pages-deploy.yml
"${REPO_DIR}/.github/workflows/deploy.yml"

# 3. Create wrangler.toml
cat > "${REPO_DIR}/wrangler.toml" <<EOF
```

```
name = "${SITE_NAME}"
compatibility_date = "2026-01-01"

[env.production]
EOF

# 4. Init git
cd "${REPO_DIR}" && git init && git add . && git commit -m "init: ${SITE_NAME}"

# 5. Create Cloudflare Pages project
npx wrangler pages project create "${SITE_NAME}" --production-branch main

echo "Done. Next: connect GitHub repo in Cloudflare dashboard."
echo "  https://dash.cloudflare.com/pages"
```

---

3. Maintenance

SSL Auto-Renewal

Cloudflare Pages provisions and auto-renews SSL certificates via Cloudflare's certificate authority. No manual action required. Certificates renew 30 days before expiry. The only failure mode is if a custom domain's DNS stops pointing to Cloudflare — the alert system in Section 4 catches this.

DNS Consolidation

Target: All domains to Cloudflare DNS.

Current state: 2 on Cloudflare, 1 on Vercel, 1 on AWS Route53, 3 on one.com nameservers, 3 unknown/third-party.

Migration steps per domain:

1. Log in to registrar, change nameservers to `ana.ns.cloudflare.com` and `bob.ns.cloudflare.com`
2. Cloudflare imports existing DNS records automatically (zone scan)
3. Verify records in Cloudflare dashboard, then activate proxy (orange cloud) for web traffic

Registrar note: Domains registered at one.com (.no TIDs) — nameserver change takes 15 minutes to 4 hours for .no domains. For .ba domains, the registrar controls this; requires contacting them directly.

Dependency Updates (Renovate)

Save as `renovate.json` in every repo root:

```
{
  "$schema": "https://docs.renovatebot.com/renovate-schema.json",
  "extends": ["config:recommended"],
  "schedule": ["every sunday"],
  "prCreationDelay": "0 minutes",
  "packageRules": [
    {
      "matchUpdateTypes": ["minor", "patch"],
      "automerge": true,
      "automergeType": "pr",
      "automergeStrategy": "squash"
    },
    {
      "matchUpdateTypes": ["major"],
      "automerge": false,
      "labels": ["dependencies", "major-update"]
    }
  ],
  "vulnerabilityAlerts": {
    "enabled": true,
    "labels": ["security"]
  }
}
```

Enable Renovate at https://github.com/apps/renovate for each repo. No server needed.

Backup Strategy

Asset	What	Where	Retention
Source code	Full git history	GitHub (primary)	Permanent
Source code mirror	Bare git clone	Azure VM `/opt/backups/git-mirrors/`	90 days rolling
Cloudflare Pages deployments	Build artifacts	Cloudflare (automatic, last 25 builds)	Automatic
DNS zone	Export via CF API	`/Users/makinja/system/backups/dns/` (weekly cron)	12 months

Asset	What	Where	Retention
Secrets inventory	Encrypted note	Vaultwarden (vault.basicconsulting.no)	Permanent

DNS zone backup cron (add to crontab):

```
# Weekly DNS zone backup — runs every Sunday 02:00
0 2 * * 0 curl -s "https://api.cloudflare.com/client/v4/zones?per_page=50" \
  -H "Authorization: Bearer $CF_API_TOKEN" | \
  node /Users/makinja/system/tools/cf-zone-export.js > \
  /Users/makinja/system/backups/dns/zones-$(date +%Y%m%d).json
```

DR: Restore Site in < 60 Seconds

> **NOTE — wrangler 4.x breaking change:** `wrangler pages deployment rollback` is removed in wrangler 4.x and must NOT be used. See MC #8494. Option A below replaces it with the CF API re-deploy path.

```
# Option A: CF API re-deploy (STANDARD DR PATH — replaces deprecated wrangler rollback)
# Time: ~11 seconds. CF content-hash cache means zero bytes re-uploaded for unchanged files.
export CF_API_TOKEN="<your-cloudflare-api-token>"
export CF_ACCOUNT_ID="<your-cloudflare-account-id>"
export CF_PROJECT_NAME="<site-name>"

# List last 10 deployments
curl -s
"https://api.cloudflare.com/client/v4/accounts/${CF_ACCOUNT_ID}/pages/projects/${CF_PROJECT_NA
ME}/deployments" \
  -H "Authorization: Bearer ${CF_API_TOKEN}" | \
  python3 -c "import sys,json; [print(d['id'], d['created_on'][:19],
d.get('deployment_trigger',{}).get('metadata',{}).get('commit_message','')[:60]) for d in
json.load(sys.stdin)['result'][:10]]"

# Re-deploy target deployment ID
curl -s -X POST \

"https://api.cloudflare.com/client/v4/accounts/${CF_ACCOUNT_ID}/pages/projects/${CF_PROJECT_NA
ME}/deployments/<deployment-id>/retry" \
```

```
  -H "Authorization: Bearer ${CF_API_TOKEN}" \
  -H "Content-Type: application/json" | python3 -c "import sys,json; r=json.load(sys.stdin);
print('OK —', r['result']['id']) if r['success'] else print('ERROR:', r['errors'])"

# Option B: Redeploy from git (if CF deployment history cleared)
cd /path/to/site-repo && npm run build && \
npx wrangler pages deploy ./out --project-name=<site-name> --branch=main
# Time: 30-90 seconds depending on build

# Option C: Emergency static serve from Azure VM (last resort)
scp -r ./out alai-admin@4.223.110.181:/var/www/<site-name>
ssh -i ~/.ssh/azure_alai alai-admin@4.223.110.181 \
  "sudo caddy reverse-proxy --from <domain> --to localhost:8080"
# Time: ~120 seconds
```

Option A is the standard DR path. Target: < 60 seconds. Tested monthly as part of Proveo validation.

---

4. Alarms and Escalation

SENTINEL daemons live in `/Users/makinja/system/tools/`. Alerting routes to Slack `#infra-alerts` channel.

Alert Table

Metric	Threshold	Channel	L1 Action	L2 Action	L3 Action
Uptime (HTTP 200)	< 100% for 5 min	#infra-alerts (Slack)	Auto-retry; post alert	Kelsey investigates: CF status page, DNS check	Escalate to CEO; activate DR (Option C)
Build failure	Any failed build on main	#infra-alerts	Alert with build URL + error log	Kelsey reviews workflow, checks CF Pages build log	Revert last commit: `git revert HEAD && git push`
SSL cert expiry	< 30 days to expiry	#infra-alerts	Alert; verify CF auto-renewal is active	Manual CF cert renewal trigger	Contact Cloudflare support

Metric	Threshold	Channel	L1 Action	L2 Action	L3 Action
5xx rate	> 1% of requests over 10 min	#infra-alerts	Alert with request sample	Kelsey checks CF Pages function logs	Rollback via CF API re-deploy (Option A, DR section)
Traffic anomaly	> 10x baseline in 5 min	#infra-alerts	Alert; verify CF rate limiting active	Check CF analytics for origin; enable under-attack mode	Contact Cloudflare support
Bandwidth overage	> 80% of plan limit	#infra-alerts	Alert; review top assets	Optimize images, add cache headers	Upgrade CF plan or move heavy assets to R2

SENTINEL Integration

Add to `/Users/makinja/system/tools/sentinel-uptime.sh`:

```
#!/usr/bin/env bash
# Uptime check for all ALAI sites — run every 5 minutes via cron
SITES=(
  "https://alai.no"
  "https://snowit.ba"
  "https://getdrop.no"
  "https://app.getdrop.no"
  "https://basicconsulting.no"
  "https://basicfakta.no"
  "https://bilko-demo.alai.no"
  "https://kenyhot.pro"
  "https://merdzanovic.ba"
  "https://docs.alai.no"
  "https://sign.basicconsulting.no"
  "https://boards.basicconsulting.no"
  "https://vault.basicconsulting.no"
)

for SITE in "${SITES[@]}"; do
  STATUS=$(curl -s -o /dev/null -w "%{http_code}" --max-time 10 "$SITE")
  if [ "$STATUS" != "200" ] && [ "$STATUS" != "301" ] && [ "$STATUS" != "302" ]; then
    node /Users/makinja/system/tools/slack.js send "#infra-alerts" \
      "ALERT: $SITE returned HTTP $STATUS at $(date -u +%Y-%m-%dT%H:%M:%SZ)"
```

```
  fi
done
```

Crontab entry: `*/5 * * * * bash /Users/makinja/system/tools/sentinel-uptime.sh`

---

5. Cost

Per-Site Monthly Cost (Target State: Cloudflare Pages)

Site	Current Platform	Current Cost	CF Pages Cost	Notes
alai.no	Cloudflare Pages	€0	€0	Already there
snowit.ba	GitHub Pages	€0	€0	Migrate from GitHub Pages
getdrop.no	Azure VM (Caddy)	Shared with VM	€0	Static landing only
app.getdrop.no	Azure VM (Caddy)	Shared with VM	Not applicable	Next.js app, stays on VM
basicconsulting.no	Vercel	€0 (Free)	€0	Migrate from Vercel
basicfakta.no	Vercel	€0 (Free)	€0	Migrate from Vercel
bilko-demo.alai.no	GCP Cloud Run	€5-10	€0	Static export possible; see note
kenyhot.pro	Vercel	€0 (Free)	€0	Client site, coordinate
merdzanovic.ba	Vercel	€0 (Free)	€0	Client site, coordinate
docs.alai.no	Azure VM	Shared with VM	Not applicable	BookStack = dynamic, stays on VM
sign.basicconsulting.no	Azure VM	Shared with VM	Not applicable	Documenso = dynamic, stays on VM
boards.basicconsulting.no	Azure VM	Shared with VM	Not applicable	Planka = dynamic, stays on VM
vault.basicconsulting.no	Azure VM	Shared with VM	Not applicable	Vaultwarden = dynamic, stays on VM
bilko-api, bilko-intesa-demo	GCP Cloud Run	€5-10	Not applicable	Dynamic services, stay on GCP

Note on bilko-demo.alai.no: If Bilko web can be exported as static (Next.js `output: 'export'`), it moves to CF Pages for €0. If it requires server-side rendering (API routes, auth), it stays on GCP Cloud Run. This is a code-level decision for CodeCraft. Placeholder cost assumes migration

succeeds.

Annual Total (Target State)

Provider	Services After Migration	Monthly	Annual
Cloudflare Pages	9 static sites	€0	€0
GCP Cloud Run	Bilko API + demo services (if SSR)	€5-10	€60-120
Azure VM	BookStack, Documenso, Planka, Vaultwarden, Drop app	€50	€600
GitHub Pages	snowit.ba (until CF migration)	€0	€0
one.com domains	alai.no, basicconsulting.no, getdrop.no, bilko.io	€17	€200
TOTAL		**€72-77/month**	**€860-920/year**

Current vs Target Delta

- **Current:** €72-127/month
- **Target:** €72-77/month (static sites are free; dynamic services stay)
- **Delta:** -€0 to -€50/month (savings only materialize if Vercel Pro tier is confirmed and removed)
- **Key finding:** Most current cost is the Azure VM (€50) and one.com domains (€17). These are not reducible by a hosting platform switch — they serve dynamic apps and DNS. The hosting consolidation eliminates Vercel as a dependency and reduces operational complexity.

Scale: 30 Sites by 2027

At 30 sites, Cloudflare Pages remains €0 (no per-site pricing). The only cost growth vectors are:

- Azure VM upgrade if Drop/BookStack need more resources: +€20-40/month for next tier
- Additional one.com domain registrations: ~€20/year each
- GCP Cloud Run if Bilko scales: usage-based, estimate €10-30/month at moderate traffic

Projected 2027 total: €100-130/month at 30 sites. Cloudflare Pages does not contribute to this increase.

---

6. Migration Plan

Priority 1 = immediate (no dep, low risk). Priority 2 = planned (some coordination). Priority 3 = blocked/external.

Domain	Current Platform	Target Platform	Priority	Downtime Window	Dependency	MC Task
alai.no	Cloudflare Pages	Cloudflare Pages	-	None	None — already done	Done
basicconsulting.no	Vercel	Cloudflare Pages	1	0 (DNS already on CF)	Find repo	#8482
basicfakta.no	Vercel	Cloudflare Pages	1	< 5 min (NS change)	Find repo, change registrar NS	#8483
snowit.ba	GitHub Pages	Cloudflare Pages	2	< 5 min	Move DNS from AWS Route53 to CF	#8484
getdrop.no	Azure VM (Caddy)	Cloudflare Pages (static)	1	0 (DNS on Vercel, move to CF)	Static export of Next.js landing	#8485
app.getdrop.no	Azure VM (Caddy)	Azure VM (stay)	-	None	Dynamic Next.js app	No action
bilko-demo.alai.no	GCP Cloud Run	Cloudflare Pages (if static export works)	2	0 (DNS already on CF)	CodeCraft confirms static export	#8486
kenyhot.pro	Vercel	Cloudflare Pages	3	< 5 min	Coordinate with client, DNS on Vercel	#8487
merdzanovic.ba	Vercel	Cloudflare Pages	3	< 5 min	Coordinate with client, third-party DNS	#8488
bilko.io	None (down)	Cloudflare Pages	2	N/A (currently down)	Fix one.com DNS, point to CF	#8489
docs/sign/boards/vault.basicconsulting.no	Azure VM	Azure VM (stay)	-	None	Dynamic apps	No action
bilko-api, bilko-intesa-demo	GCP Cloud Run	GCP Cloud Run (stay)	-	None	Dynamic API services	No action

Total sites to migrate: 8 static sites. 4 stay on current platform (dynamic apps/services). 2 done (alai.no, basicconsulting.no).

Migration Log

Date	Domain	From	To	Downtime	TTFB Before	TTFB After	Notes
2026-04-20	basicconsulting.no	Vercel (76.76.21.21)	CF Pages	~60s	114ms	51ms (warm avg)	MC #8482. DNS: A->CNAME. Validation required domain re-add. TTFB improved 55%. Proveo pilot validated #8490.
2026-04-20	bilko.io	one.com (down)	CF Pages	N/A (site was down)	N/A	68ms (warm avg)	MC #8489. Apex CNAME not possible on one.com free tier (paid feature). Switched to Cloudflare NS (ana.ns.cloudflare.com, bob.ns.cloudflare.com). CF Pages zone ID: 62d89b79f0648d3fa1d045335a989ea7. DNS: CNAME flattening bilko.io → bilko-io.pages.dev (proxied), www → bilko-io.pages.dev.

Paused migrations:

- MC #8483 (basicfakta.no) — Inventory error: site has serverless functions (Vercel Edge), not pure static. Requires CodeCraft assessment.

- MC #8484 (snowit.no) — Inventory error: site has API routes (Next.js), not pure static. Requires CodeCraft assessment.

Audit verdict for #8486 (bilko-demo.alai.no): Full-stack Next.js app with dynamic API routes. Stays on GCP Cloud Run. Not eligible for CF Pages migration.

---

7. Lessons Learned

2026-04-20 — CF Browser Integrity Check blocks headless clients

Incident: LightRAG 46h outage (MC #8487 followup)

Problem: Automation HTTP clients (Python urllib, Node fetch, etc.) get HTTP 403 (error code 1010) from CF-proxied hostnames with Browser Integrity Check (BIC) enabled, even when IP bypass or CF Access service tokens are configured.

Root cause: BIC layer evaluates BEFORE Access policies and blocks requests based on User-Agent string. Python/Node default UAs trigger block, but curl/wget/browser tests pass — creating a false sense of security.

Fix: Create Cloudflare Configuration Rule disabling BIC per hostname. See rule INFRA-CF-001 (`~/system/rules/cf-proxied-api-bic-whitelist.md`) and BookStack page ID 2692.

Evidence: `~/system/evidence/lightrag-ingestion-investigation-20260420-215700.md`

Hostnames affected: ollama.basicconsulting.no (fixed), lightrag.basicconsulting.no (verify needed)

---

8. DoD Checklist

☐ File exists at `/Users/makinja/system/specs/ALAI-STATIC-HOSTING-BLUEPRINT.md`

☐ BookStack sync task created — MC #8491 (Skillforge owner) — sync this file to docs.alai.no under "Infrastructure > Hosting"

☐ Proveo validation task created — MC #8490 (Angie Jones owner) — deploy blueprint to 1 test site (basicconsulting.no), verify < 60s rollback works end-to-end

☐ 8 migration MC tasks created: #8482 #8483 #8484 #8485 #8486 #8487 #8488 #8489

☐ SENTINEL uptime script deployed and crontab entry added

- [ ] Renovate enabled on all repos
- [ ] getdrop.no DNS moved from Vercel to Cloudflare
- [ ] 8 stale Vercel projects deleted (see inventory)

Revision #10
Created 2026-04-20 16:31:05 UTC by John
Updated 2026-05-25 07:34:01 UTC by John